MEKÂNSAL ZAMANSAL VERİ MADENCİLİĞİNDE SINIFLANDIRMA İLE SUÇ VERİLERİNİN TAHMİNİ

Muhammed Çağrı AKSU

Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Bilim Dalı
Yrd. Doç. Dr. Tolga AYDIN
2016

ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

MEKÂNSAL ZAMANSAL VERİ MADENCİLİĞİNDE SINIFLANDIRMA İLE SUÇ VERİLERİNİN TAHMİNİ

M. Çağrı AKSU

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
Bilgisayar Mühendisliği Bilim Dalı

ERZURUM
2016

T.C.
ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TEZ ONAY FORMU

MEKÂNSAL ZAMANSAL VERİ MADENCİLİĞİNDE SINIFLANDIRMA İLE SUÇ VERİLERİNİN TAHMİNİ

Yrd. Doç. Dr. Tolga AYDIN danışmanlığında, **Muhammed Çağrı AKSU** tarafından hazırlanan bu çalışma **06/01/2016** tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı – Bilgisayar Mühendisliği Bilim Dalı'nda Yüksek Lisans Tezi olarak **oybirliği/~~oy çokluğu~~ (3/0)** ile kabul edilmiştir.

Başkan	: Yrd. Doç. Dr. Tolga AYDIN	*İmza*	:
Üye	: Yrd. Doç. Dr. Emin Argun ORAL	*İmza*	:
Üye	: Yrd. Doç. Dr. Çağlar DUMAN	*İmza*	:

Yukarıdaki sonuç;

Enstitü Yönetim Kurulunun 14...../ 01...../ 2016..... tarih ve .03./.26..... nolu kararı ile onaylanmıştır.

Prof. Dr. Ertan YILDIRIM
Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ÖZET

Yüksek Lisans Tezi

MEKÂNSAL ZAMANSAL VERİ MADENCİLİĞİNDE SINIFLANDIRMA İLE SUÇ VERİLERİNİN TAHMİNİ

M. Çağrı AKSU

Atatürk Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Bilim Dalı

Danışman: Yrd. Doç. Dr. Tolga AYDIN

Halkın güvenliğini sağlamak kolluk kuvvetlerinin en önemli vazifelerinden birisidir. Teknolojinin gelişmesiyle birlikte teknoloji birçok alanda kullanılabilir olduğundan, suç tahmini ve suç analizi gibi halkın güvenliğini sağlamakta kolluk kuvvetlerine yardım etmeyi amaçlayan alanlarda da teknolojiden yararlanılmaktadır.

Veri saklama ve işleme araçlarının gelişmesi büyük miktarda veriyi işleme yeteneğine sahip olan teknolojilerin üretilmesini sağladı. Bu teknolojilerden birisi de veri madenciliğidir. Bu çalışmada mekânsal-zamansal veri madenciliği sınıflandırma algoritmalarından CR-Ω+ algoritmasıyla suç tahmini ve analizi gerçekleştirilmektedir. Elde edilen sonuçların kolluk kuvvetlerinin taktik ve planlama operasyonlarına katkı sağlaması amaçlanmıştır.

2016, 102 Sayfa

Anahtar Kelimeler: Zamansal Mekânsal Veri Madenciliği, Sınıflandırma, CR-Ω+ algoritması

ABSTRACT

Master Thesis

FORECAST OF CRIME DATA BY CLASSIFICATION IN SPATIO-TEMPORAL DATA MINING

M. Çağrı AKSU

Ataturk University
Institute of Science and Technology
Department of Computer Engineering
Computer Engineering Department

Supervisor: Asst. Prof. Dr. Tolga AYDIN

Ensuring the safety of the public is one of the most important duties of the police force. As technology can be used in various fields with the development of technology, technology is used in the fields which aim to help the police force in ensuring the safety of the public via activities such as forecast of crime and crime analysis.

The development of data processing and storage tools enables the development of technologies which can process large amounts of data. One of these technologies is data mining. In this study, crime forecasting and analysis is conducted by using the CRC-Ω+ algorithm which is one of the spatio-temporal data mining classification algorithms. The obtained results have been aimed to contribute towards the tactical and planning operations of the police force.

2016, 102 Pages

Keywords: Spatio-Temporal Data Mining, Classification, CR-Ω+ Algorithm

TEŞEKKÜR

Bu çalışma boyunca bilgi, tecrübe ve güler yüzü ile çalışmamda yol gösteren danışman hocam Sayın Yrd. Doç. Dr. Tolga AYDIN'a, tez çalışmam boyunca desteklerini esirgemeyen sevgili eşim Elif AKSU' ya, lisans ve yüksek lisans eğitimim boyunca bir çok yardımda bulunan Prof. Dr. Aslan GÜLCÜ ve Suna GÜLCÜ'ye, çalışmada kullanılan verilerin alınmasında yardımcı olan Artvin Çoruh Üniversitesi Bilgi İşlem Daire Başkanı Mehmet ÖZDEMİR'e, İngilizce makalelerin Türkçeye çevrilmesinde yardımcı olan mesai arkadaşım Rahman DURAN'a teşekkür ederim.

Bu çalışmayı, şimdiye kadar benden, maddi, manevi hiçbir desteği esirgemeyen anne ve babama ithaf ederim.

Muhammed Çağrı AKSU
Ocak, 2016

İÇİNDEKİLER

KISALTMALAR DİZİNİ

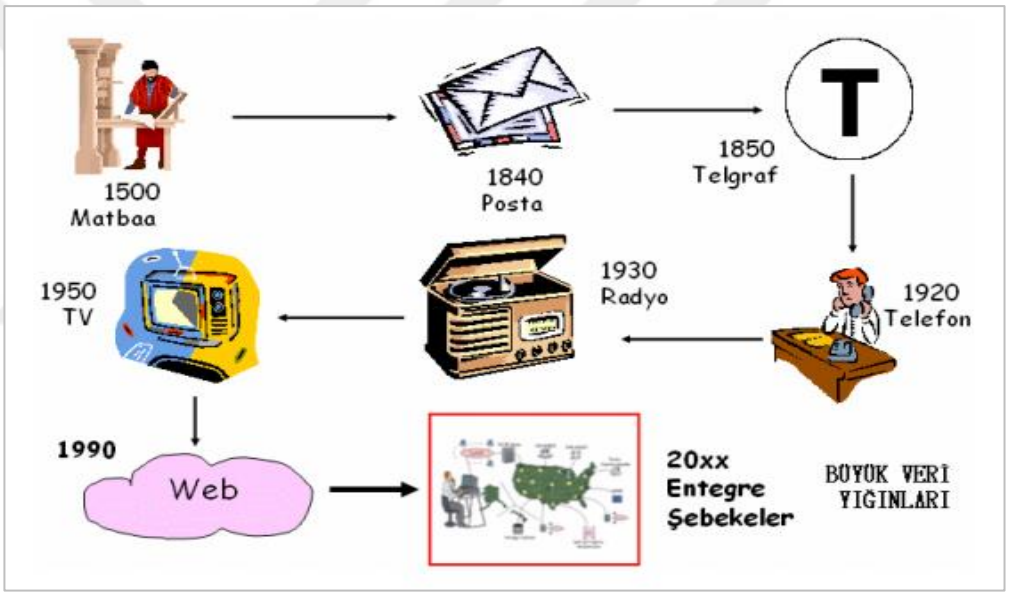
CBS	Coğrafi Bilgi Sistemleri
HMM	Hidden Markov Model
MBR	Minimum Bounding Rectangle
MM	Markov Model
SJI	Spatial Join Index
YSA	Yapay Sinir Ağı

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

1. GİRİŞ

Elektronik veri yönetimi kavramı 1960'lı yılların başlarında duyulmaya başlamıştır. Ancak o dönemin standartlarını günümüz standartları ile kıyasladığımızda oldukça ilkel kalmaktadır. O dönemki standartlar yazılım ve donanım maliyetleri açısından oldukça pahalıdır. Şekil 1.1'de görüleceği üzere, 1960'lardan sonraki yıllarda, toplanan veri miktarında artış gözlenmiştir. Bu nedenle, daha gelişmiş elektronik veri yönetim tekniklerine gereksinim duyulmuştur (Schumann 2005).



Şekil 1.1. Gelişen bilişim teknolojisi ve veri oluşumu (Baykasoğlu 2005)

Elektronik veri saklama ve elektronik veri analizi işlemlerinin gelişmesi, veri yığınlarını işleyebilen teknolojilerin oluşmasını sağlamıştır. Veri madenciliği de bu teknolojilerden birisidir (Schumann 2005). Veri madenciliği 1980'lerin sonunda ortaya çıkmış, 1990'larda büyük bir gelişim göstermiş ve gün geçtikçe gelişimi devam etmektedir. Veri madenciliği, karar alma konusunda önemli katkıları olan bir teknolojidir (Han and Kamber 2000; Beitel 2005; Özçınar 2006).

Bu çalışmada yukarıda bahsedilen veri madenciliği teknolojisinden yararlanılmıştır. Mekânsal-zamansal veri madenciliği sınıflandırma algoritmalarından CR-Ω+ algoritmasıyla (Calderón *et al.* 2010) suç tahmini ve analizi gerçekleştirilmiştir. Elde edilen sonuçların kolluk kuvvetlerinin taktik ve planlama çalışmalarına katkı sağlaması amaçlanmıştır.

2. KURAMSAL TEMELLER

Veri madenciliği, büyük veri yığınlarından kullanışlı bilgilerin algoritmalar vasıtasıyla otomatik olarak keşfedilme sürecidir. Veri madenciliği ile hava tahmini veya yeni gelen müşterinin ne kadar para harcayacağı gibi gelecek ile ilgili tahminler yapılabilir (Tan *et al.* 2005).

Veri madenciliği konusunda birçok yöntem ve algoritma bulunmaktadır. Veri madenciliği yöntemleri sınıflandırma, kümeleme ve birliktelik kuralları olarak üç ana başlığa ayrılır. Sınıflandırma, sınıfı tanımlanmış verileri kullanarak, sınıfı belli olmayan verilerin sınıfını tahmin etmek için kullanılan bir yöntemdir. Kümeleme ise veriler içerisinde bulunan benzerliklere göre verilerin gruplandırılması işlemidir. Birliktelik kuralları, veriler içerisinde bulunan, birlikteliklerin, ilişkilerin ve bağıntıların kurallar halinde çıkarılması işlemidir (Birant vd 2010).

Bu çalışmada veri madenciliği yöntemlerinden sınıflandırma kullanılarak, zamansal-mekânsal veri madenciliği algoritmalarından CR-Ω+ algoritmasıyla (Calderón *et al.* 2010) suç tahmini ve analizi gerçekleştirilmiştir.

3. MATERYAL ve YÖNTEM

3.1. Materyal

Bu çalışmada suç verisi olarak Artvin Emniyet Müdürlüğü'nden alınan suç verileri kullanılmıştır. Toplamda 2012 ve 2013 yıllarını kapsayan 1998 adet suç verisi alınmıştır. Alınan veriler, veri madenciliği için uygun olmadığından, veri madenciliği yapmadan önce ön işleme tabi tutulmuştur. Eksik alana sahip veriler veri kümesinden çıkartılmıştır. Ayrıca saat alanı Çizelge 3.1'deki görüldüğü gibi 8 dilime ayrılmıştır. Böylelikle saat alanı yumuşatılmıştır. Kalan verilerden 423 tanesi Artvin il merkezinde gerçekleşen suçlardır. 423 adet suç verisinden rastgele seçilen 30 veri, tahminlerin doğruluğunu saptayabilmek için ayrılmıştır. Sonuç olarak veri madenciliği 393 adet suç verisiyle gerçekleştirilmiştir. Bu suç verileri içerisinde ay, mahalle, suçun türü ve saat dilimi bulunmaktadır.

Çizelge 3.1. Saat dilimlerine karşılık gelen saat aralıkları

Saat Dilimi	Aralık
1	00:00 - 02:59
2	03:00 - 05:59
3	06:00 - 08:59
4	09:00 - 11:59
5	12:00 - 14:59
6	15:00 - 17:59
7	18:00 - 20:59
8	21:00 - 23:59

3.2. Yöntem

Suç tahmini ve analizini gerçekleştirmek için PHP programlama dili vasıtasıyla bir web tabanlı uygulama hazırlanmıştır. Uygulamaya veri madenciliği yapabilmek için sınırsız veri kümesi eklenebilmektedir. Eklenilen herhangi bir veri kümesine veri girişi tek olarak yapılabildiği gibi Excel dosyası aracılığıyla toplu olarak da yapılabilmektedir.

Veri kümesindeki tüm alanlar güncellenebilir ve silinebilirdir. Uygulama ile veri kümesindeki alanlar için istatistiksel grafikler hazırlanabilir ve CR-Ω+ algoritmasıyla suç tahmininde bulunulabilir. Tahmin işleminde, tahmin edilmesi istenilen veriler programa, tek tek girilebildiği gibi Excel dosyası vasıtasıyla toplu olarak da girilebilir. Tahmin işlemi sonucunda son kullanıcıya, girilen verilerin sınıflandırılma oranı ve tahmin oranı grafiksel olarak sunulmaktadır. Uygulama, veri madenciliği için hatalı ve eksik verilerin temizlenmesi işlemini gerçekleştirmemektedir. Veri madenciliği yapılacak verilerin, veri madenciliğine hazır yani hatalı yada eksik verilerin temizlenmiş olması gerekmektedir.

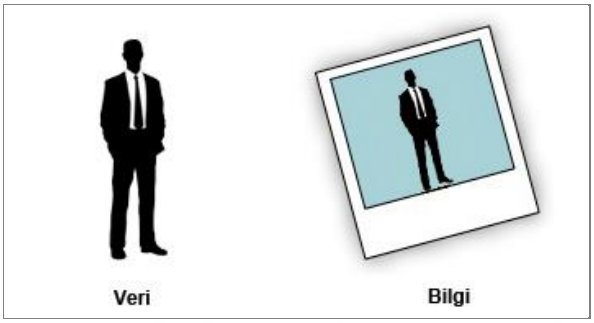
Yöntem olarak zamansal-mekânsal veri madenciliğinde sınıflandırma yöntemi kullanılmıştır. CR-Ω+ algoritmasıyla suç tahmini ve analizi gerçekleştirilmiş ve elde edilen sonuçların kolluk kuvvetlerinin taktik ve planlama operasyonlarına katkı sağlaması amaçlanmıştır.

Bu kısımdan 4. bölüme kadar çalışmanın yöntemi olan veri madenciliği, zamansal veri madenciliği, mekânsal veri madenciliği ve zamansal-mekânsal veri madenciliği ayrı ayrı incelenmiş, ilgili konulardaki algoritmalar örneklerle açıklanmıştır.

3.3. Veri Madenciliği

Veri dünyadaki gerçekler olarak tanımlanabilir. Sayılar, metinler, sesler, görüntüler hepsi birer veridir. Veriler karar vermeye yarayan soyut simge dizileridir. Örneğin boyunuzun 1.80 cm olması, saçınızın kahverengi olması gibi. Bizler veriyi duyu organlarımız ile algılarız ve beynimiz bu veriyi işler. Veri ve bilgi kavramları, veri tabanlarının ve veri ambarlarının temel unsurlarıdır. Verinin temel özelliği henüz işlenmemiş olmasıdır. Bilgi ise öğrenerek, deneyerek, araştırarak elde edilen her türlü sonuç, yani bir anlamda verinin işlenmiş hali veya verinin bir ürünüdür. Şekil 3.1'de görüldüğü gibi bir kişiyi fotoğraf çektiğimizde, çekilen fotoğraf bilgidir, o kişinin dış görünüşü ise veridir. Veri her zaman doğrudur. Örneğin yirmi altı yaşındaki biri aynı zamanda kırk beş yaşında olamaz. Ancak bilgi yanlış olabilir, bir kişiyle alakalı iki

farklı bilgi olabilir, bu bilgilerden birisi bu kişinin 1987 doğumlu olduğunu belirtirken, diğeri 1968 doğumlu olduğunu belirtebilir (Ingebrigtsen 2013).

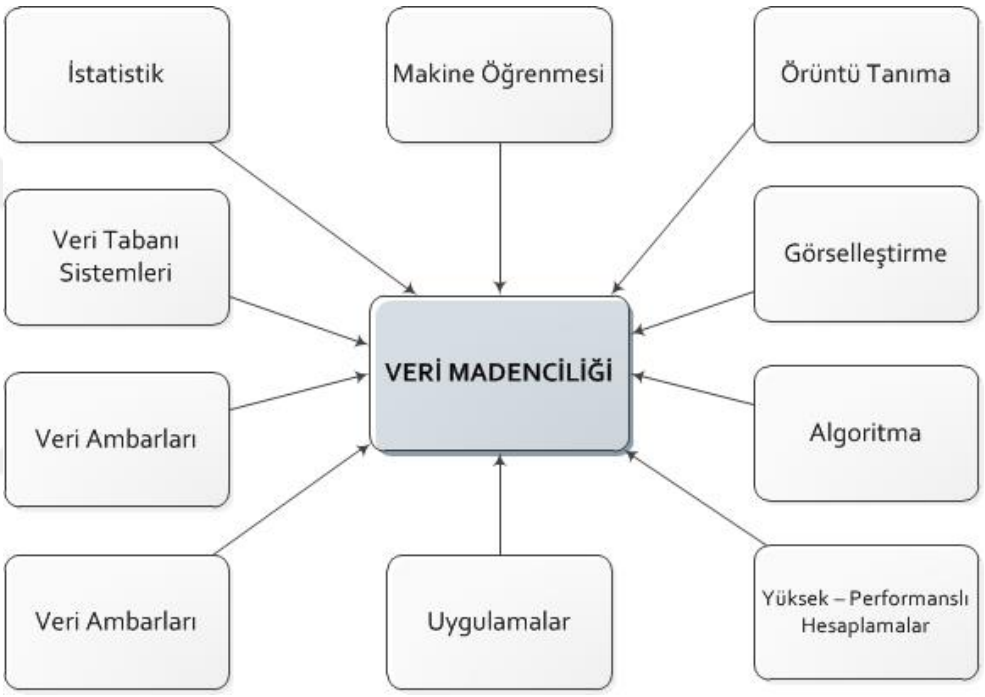


Şekil 3.1. Veri ve bilgi arasındaki fark

Veri, günlük yasam döngüsü içerisinde doğal olarak oluşmakta ya da çeşitli cihazların olağan aktiviteleri sonucu meydana gelmektedir. Teknolojinin gelişmesine paralel olarak bilişim sistemleri, teknolojik cihazlar hayatımızın her alanına dâhil olmaktadır. Günümüzde sağlık, eğitim, ticaret, bankacılık, turizm, güvenlik gibi birçok alanda bilişim sistemleri kullanılmaktadır. Kullanılan bilgi sistemleri neticesinde milyonlarca veri, veri tabanlarında ya da depolama birimlerinde birikmektedir. Biriken bu veriler kaydedildikleri yerde öylece beklerken bir anlam ifade etmezler. Kaydedilen verilerin bir anlam ifade edebilmesi için bilgiye dönüştürülmeleri gerekir (Özçınar 2006). Depolanan bu verilerin, bilgiye dönüştürülesi veri madenciliği vasıtasıyla gerçekleştirilir. Aşağıda veri madenciliği için farklı tanımlar verilmiştir.

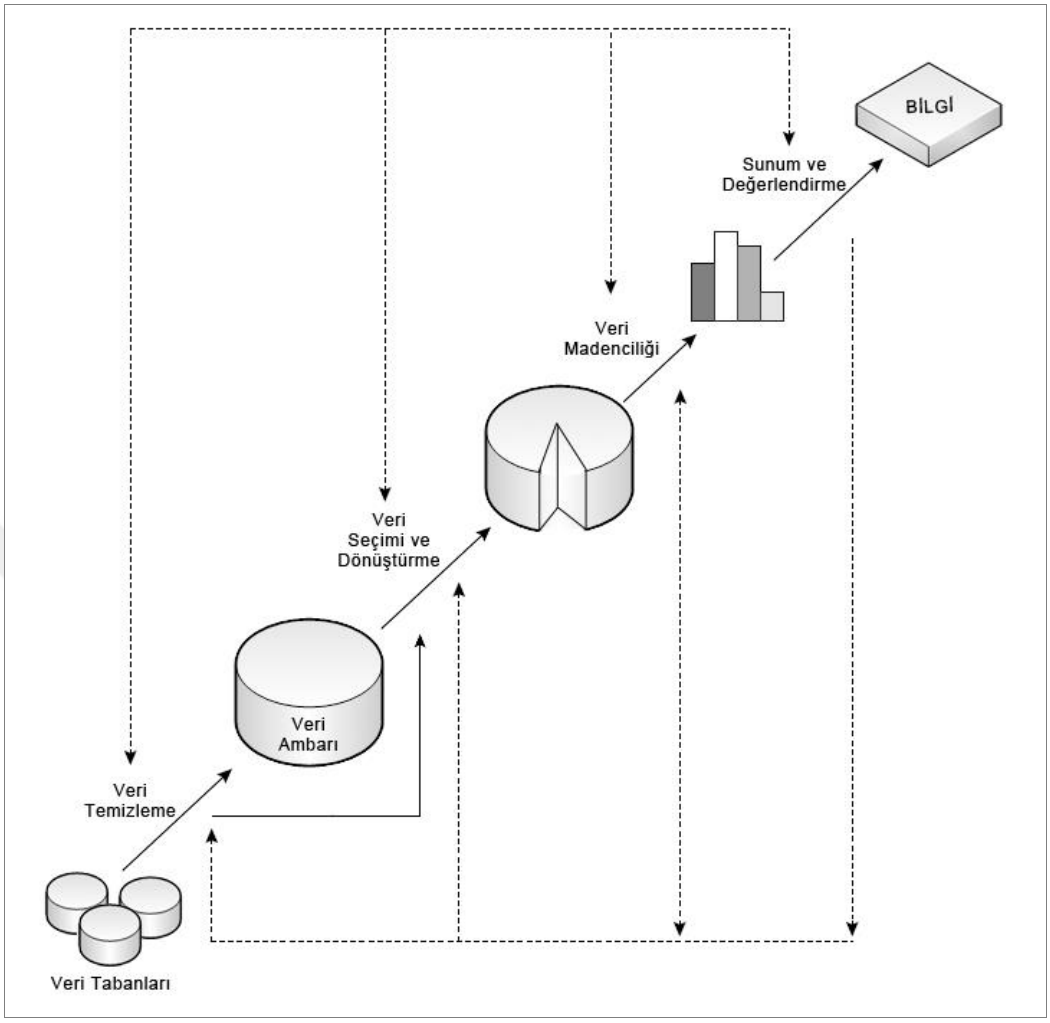
- Veri madenciliği, genellikle büyük ölçekli veriler arasından ilişkiler bularak, bunları veri sahibi için hem anlaşılır hem de kullanılabilir bir biçimde, yeni bir yöntemle özetleyerek analiz etme işlemidir (Han *et al.* 2001).
- Veri madenciliği, büyük veri tabanlarından bilgi çıkartabilmek amacıyla makine öğrenmesi, örüntü tanıma, istatistik, görselleştirme, veri tabanı sistemleri, uygulamalar, algoritmalar gibi alanların tekniklerini bir araya getiren disiplinler arası bir alandır (Cabena *et al.* 1998). Şekil 3.2'ye bakıldığında veri madenciliğinin disiplinler arası bir alan olduğunu görülmektedir.

- Veri madenciliği büyük miktarda veri içinden gelecekle ilgili tahmin yapmamızı sağlayacak bağıntı ve kuralların bilgisayar programları kullanarak aranmasıdır (Alpaydın 2000).
- Veri madenciliği daha önceden bilinmeyen, geçerli ve uygulanabilir bilgilerin geniş veri tabanlarından elde edilmesi ve bu bilgilerin işletme kararları verirken kullanılmasıdır (Silahtaroğlu 2008).



Şekil 3.2. Veri madenciliği disiplinler arası bir alandır (Cabena *et al.* 1998)

Veri madenciliği ismi, veri madenciliğini tarif ederken yanlış anlaşılabilir. Altın madencisi altın arayan kişiye denilirken, veri madencisi veri aramaz, veriden çıkartılacak bilgiyi arar (Han and Kamber 2001). Veri madenciliği tabiri yanlış anlaşılabileceği için bunun yerine, veriden bilgi keşfi (Knowledge Discovery from Data veya kısaca KDD) veya bilgi madenciliği tabiri kullanılmalıdır. Ancak Şekil 3.3'de gösterildiği gibi veri madenciliği, veriden bilgi keşfi sürecinin bir adımı olmasına rağmen, bir çok kaynakta veriden bilgi keşfi süreci için veri madenciliği tabiri kullanıldığından, bu çalışmada da veriden bilgi keşfi tabiri yerine, veri madenciliği tabiri kullanılacaktır.



Şekil 3.3. Veri madenciliği, veriden bilgi keşfinin bir adımıdır

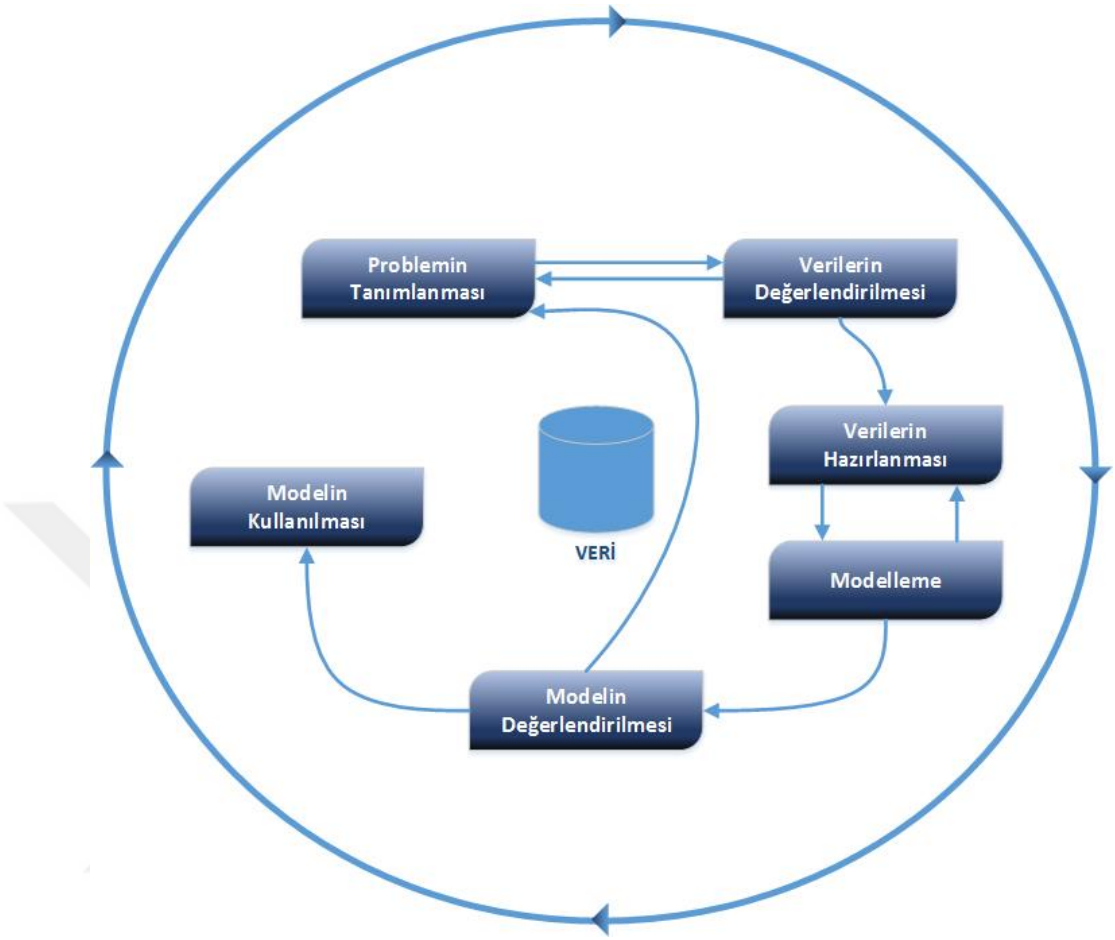
Veri madenciliğinde önemli olan nokta, keşfedilen bilginin önceden bilinmeyen bir bilgi olmasıdır. Yani elde edilen bilgi veri madenciliği kullanmadan tahmin edilebiliyorsa veri madenciliğini kullanmak ekonomik değildir. Örneğin, bir markette domates alan birçok kişi salatalık da alır. Bu bilinen bir bilgidir ve bu nedenle market çalışanları domatesleri koydukları yerin hemen yanına salatalıkları da koyarlar. Bu bilgi için veri madenciliği yapmak gereksizdir. Ancak meşhur bira-çocuk bezi örneğinde bilinmeyen yani tahmin edilemeyen bilgi ortaya çıkmaktadır. Özellikle Cuma günleri çocukları için bez satın alan babalar kendileri için de bira satın almaktadırlar. Bu bilgi tahmin edilemeyen bir bilgidir. Bunun gibi bilinmeyen bilgilerin çıkarılması için veri madenciliğine gerek duyulur ve veri madenciliği bu tip bilgilere ulaşmak için kullanılmalıdır.

3.3.1. Veri madenciliği süreci

Veriden bilgi keşfetme yani veri madenciliği bir süreçten oluşmaktadır. Veri madenciliği süreçlerinden en yaygın olarak kullanılanı, Şekil 3.4'de diyagramı gösterilen, veri madenciliği araçlarını satan bazı firmaların oluşturduğu bir şirketler birliği tarafından geliştirilen, Çapraz Endüstri Veri Madenciliği Standart Süreci (The Cross-Industry Standart Process for Datamining - CRISP-DM) dir. 1996 yılında oluşturulan CRISP-DM şirketler birliği, Daimler-Chrysler, NCR (National Cash Register), SPSS firmalarından oluşmaktadır. CRISP-DM süreci aşağıda listesi verilen 6 aşamadan oluşmaktadır.

- Problemin tanımlanması
- Verinin değerlendirilmesi
- Verinin hazırlanması
- Modelleme
- Modelin değerlendirilmesi
- Modelin kullanılması

Veri madenciliği sürecin her bir aşamasının dikkatle izlenmesi gerekmektedir. Bir aşamanın sonucu, diğer bir aşamanın girdisidir. Bu sebeple her aşama bir önceki aşamanın sonuçlarına bağlıdır (Gürsoy 2009).



Şekil 3.4. CRISP-DM süreci

3.3.1.a. Problemin tanımlanması

Veri madenciliği sürecinin en önemli aşamasıdır. Veri madenciliği çalışmalarında başarılı olmanın ilk şartı, uygulamanın hangi işletme amacı için yapılacağının açık bir şekilde tanımlanmasıdır. İlgili işletme amacı, işletme problemi üzerine odaklanmış ve açık bir dille ifade edilmiş olmalı, elde edilecek sonuçların başarı düzeylerinin nasıl ölçüleceği bu aşamada tanımlanmalıdır (Gürsoy 2009).

Bu aşamada yapılacak veri madenciliği işleminin hangi amaç için yapıldığı, hangi sektör için yapıldığı, neyin hedeflendiği, nelere ihtiyaç duyulduğu, proje sonucunda elde edilen bilginin nasıl değerlendirileceği, açık bir şekilde anlatılmalıdır.

Aşağıdaki listede veri madenciliğine sürecinde tanımlanabilecek problemler listelenmiştir.

- Hangi müşteri aldığı krediyi geri ödemeyebilir?
- Bir müşteriye ne kadar kredi verilebilir?
- Hangi yatırım araçlarına yatırım yapmalıyım?
- Yarın hava nasıl olacak?
- Yarın herhangi bir suç işlenebilir mi?
- Yarın orman yangını olabilir mi?

3.3.1.b. Verinin değerlendirilmesi

Bu aşama veri madenciliği için kullanılacak verilerin toplanmasıyla başlar. Veri kaynaklarının neler olduğu bulunur. Verinin nasıl bir veri olduğunun anlaşılması için veri analizi yapılır ve verinin kalitesi araştırılır. Veri madenciliği için kullanılacak verilerde eksik değerler, yanlış ya da aşırı uçta bulunan değerler var mı diye bakılır. Sonuç olarak bu aşamada veri madenciliği için kullanılacak olan verinin tam anlamıyla kavranması gerekmektedir.

3.3.1.c. Verinin hazırlanması

Bu aşamada veri madenciliği için toplanılan veriler üzerinde bir takım işlemler yapılmaktadır. Veri madenciliği için toplanılan veriler bazen eksik ya da hatalı olabilir. Örneğin veri tabanında kayıtlı olan bir alanın değeri girilmemiş olabilir, bunlar kayıp veri olarak adlandırılırlar. Kayıp verilerin yaratacağı sorunları ortadan kaldırmak için bazı teknikler aşağıdaki listede verilmiştir.

- Kayıp verinin bulunduğu kaydı veri kümesinden çıkartmak
- Kayıp verileri elle teker teker doldurmak
- Tüm kayıp verilere aynı bilgiyi girmek

- Kayıp verilere tüm verilerin ortalama değerini vermek
- Regresyon yöntemi kullanılarak diğer değişkenlerin yardımı ile kayıp olan verilerin tahmin edilmesi

Bazen de girilen veriler aşırı uçta bulunabilir bu tip verilere de gürültülü veri adı verilir. Bir kişinin boyunun 4 metre olması ya da yaşamakta olan bir insanın doğum tarihinin 1453 olması gürültülü verilere örnek olabilir. Bunların dışında veri tabanında fazladan yapılmış bir alan bulunabilir. Örneğin veri tabanında hem yaşın hem doğum tarihinin tutulması. Doğum tarihinden yaş kolaylıkla öğrenileceğinden yaş alanının doğum tarihiyle birlikte saklanması gereksizdir. Son olarak da birden fazla kaynaktan gelen, aynı alan adına sahip bilgilerin verilerin saklanış biçimleri farklı olabilir. Bu farklılığa örnek olarak veri tabanında idari ve akademik personellerin tutuluş biçimleri Çizelge 3.2'de gösterilmiştir.

Çizelge 3.2. Veri tabanlarında aynı alanların farklı şekilde saklanışları

İdari	**Akademik**
0	1
1	0
1	2
İ	A
İdari	Akademik

3.3.1.d. Modelleme

Bu aşamada çeşitli modelleme teknikleri seçilir ve uygulanır. Bunların parametreleri uygun değerlere ayarlanır. Genellikle, aynı veri madenciliği problem türü için çeşitli teknikler vardır. Bazı teknikler veri üzerinde birtakım değişikliklere ihtiyaç duyabilir. Bu nedenle bu aşamadan verilerin hazırlanması aşamasına geri dönülebilir (Chapman *et al.* 2000)

3.3.1.e. Modelin değerlendirilmesi

Bu aşamada, daha önce oluşturulmuş olan model, uygulamaya koyulmadan önce son kez tüm yönleriyle değerlendirilir, modeli oluşturan adımlar gözden geçirilir, modelin kalitesi ve etkinliği ölçülür. Modelin hedeflerine ulaştığından emin olmak için modeli iyice değerlendirmek ve modeli inşa ederken kullanılan adımları gözden geçirmek önemlidir. Bu aşamanın sonunda veri madenciliğinde hakkında bir karara varılmış olması gerekmektedir (Chapman *et al.* 2000).

3.3.1.f. Modelin kullanılması

Veri madenciliği sürecinin son aşamasıdır. Değerlendirme aşamasında gözden geçirilen ve tüm yönleriyle incelenen model bu aşamada kullanılır. Kullanılan model ile birtakım bilgiler elde edilir. Elde edilen bilgiler son kullanıcının anlayabileceği ve karar alma sürecinde kullanabileceği türden veriler olması gerekmektedir.

3.3.2. Veri madenciliği yöntemleri

Veri madenciliği konusunda birçok yöntem ve algoritma bulunmaktadır. Bu yöntemler sınıflandırılacak olursa 3 ana başlık karşımıza çıkar. Veri madenciliği yöntemleri sınıflandırma, kümeleme ve birliktelik kuralları olarak 3 ana başlığa ayrılır.

3.3.2.a. Sınıflandırma

Sınıflandırma, veri madenciliğinde sıkça kullanılan bir yöntemdir. Sınıflandırma, sınıfı tanımlanmış verileri kullanarak, sınıfı belli olmayan verilerin sınıfını tahmin etmek için kullanılan bir yöntemdir. Sınıflandırma iki adımdan oluşan bir işlemdir. Birinci adımda kullanılacak model oluşturulmaktadır. Veriler sınıflandırma algoritmasına tabi tutularak sınıflandırma kuralları oluşturulur. İkinci adımda ise oluşturulan sınıflandırma kuralları,

sınıfı belli olmayan veriler üzerine uygulanarak sınıfı tahmin edilir (Han and Kamber 2001).

Başlıca sınıflandırma yöntemleri karar ağaçları, yapay sinir ağları, genetik algoritmalar, naive-bayes, destek vektörleri ve k-en yakın komşu algoritmasıdır. Sınıflandırma yöntemi genellikle bir tahmin yapma problemi söz konusuysa kullanılır.

3.3.2.b. Kümeleme

Kümeleme, veriler içerisinde bulunan benzerliklere göre verilerin gruplandırılması işlemidir. Bu işlem birbirine benzer olan verilerin aynı kümede, birbirinden farklı olan verilerin, nesnelerin ise ayrı kümelerde yer almasıyla gerçekleştirilir. Kümeleme yöntemi, verilerin gruplandırılmasını kapsadığı için birçok alanda kullanılmaktadır. Pazar araştırmaları, desen tanıma, resim işleme ve uzaysal harita verilerinin analizinde kullanılmaktadır (Özkan 2008).

3.3.2.c. Birliktelik kuralları

Birliktelik kuralları, veriler içerisinde bulunan birliktelik kurallarının çıkarılması işlemidir. Veriler içerisindeki bütün kombinasyonlara bakılarak, verilerin birbirleriyle olan ilişkileri incelenerek birtakım örüntüler, birliktelikler keşfedilmeye çalışılır. Birliktelik kuralları genellikle “eğer şu olursa daha sonra bu olur” şeklinde ifade edilebilir (Akpınar 2000).

Birliktelik kuralları genellikle pazarlama alanında kullanılmaktadır. Pazar sepeti analizlerinde genellikle bu veri madenciliği yöntemi kullanılmaktadır ve bu yöntemle kullanıcıların alışveriş alışkanlıkları saptanmaktadır (Özkan 2008).

3.3.3. Veri madenciliğinin uygulama alanları

Veri madenciliğinde birçok yöntem ve algoritmalar bulunmaktadır. Bu yöntem ve algoritmaların kullanıldığı sektör ve uygulama alanları Güvenç (2001) tarafından aşağıdaki gibi sıralanmıştır.

Pazarlama

- Müşteri gruplandırmasında.
- Müşterilerin demografik özellikleri arasındaki bağlantıların kurulmasında.
- Çeşitli pazarlama kampanyalarında.
- Mevcut müşterilerin elde tutulması için geliştirilecek pazarlama stratejilerinin oluşturulmasında.
- Pazar sepeti analizinde.
- Çapraz satış analizleri.
- Müşteri değerleme.
- Müşteri ilişkileri yönetiminde.
- Çeşitli müşteri analizlerinde.
- Satış tahminlerinde.

Bankacılık

- Farklı finansal göstergeler arasındaki gizli korelasyonların bulunmasında.
- Kredi kartı dolandırıcılıklarının tespitinde.
- Müşteri segmentasyonunda.
- Kredi taleplerinin değerlendirilmesinde.
- Usulsüzlük tespitinde.
- Risk analizlerinde.
- Risk yönetiminde.

Sigortacılık

- Yeni poliçe talep edecek müşterilerin tahmin edilmesinde.
- Sigorta dolandırıcılıklarının tespitinde.
- Riskli müşteri tipinin belirlenmesinde.

Perakendecilik

- Satış noktası veri analizleri.
- Alış-veriş sepeti analizleri.
- Tedarik ve mağaza yerleşim optimizasyonu.

Borsa

- Hisse senedi fiyat tahmini.
- Genel piyasa analizleri.
- Hisse tespitlerinde.
- Alım-satım stratejilerinin optimizasyonu.

Telekomünikasyon

- Kalite ve iyileştirme analizlerinde.
- Hatların yoğunluk tahminlerinde.

Sağlık ve İlaç

- Test sonuçlarının tahmini.
- Ürün geliştirme.
- Tıbbi teşhis.
- Tedavi sürecinin belirlenmesinde.

Endüstri

- Kalite kontrol analizlerinde.
- Lojistik.
- Üretim süreçlerinin optimizasyonunda.

Eğitim

- Öğrenci davranışlarının öngörülmesi.
- Öğrencilerin ders seçme eğilimlerinin belirlenmesi.

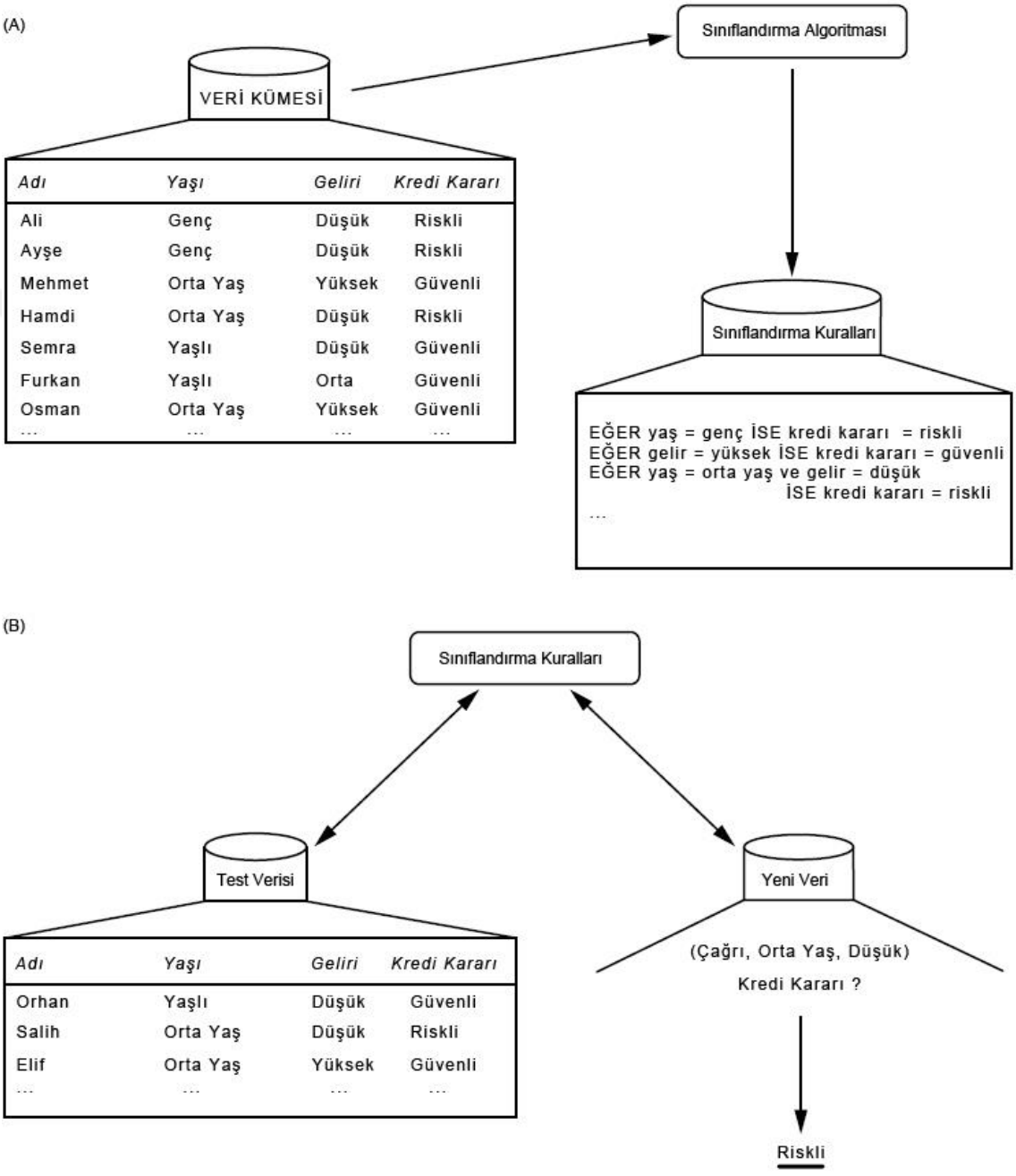
3.4. Sınıflandırma

Sınıflandırma, veri madenciliğinde sıkça kullanılan bir yöntemdir. Sınıflandırma, sınıfı tanımlanmış verileri kullanarak, sınıfı belli olmayan verilerin sınıfını tahmin etmek için kullanılan bir yöntemdir. Sınıflandırma iki adımdan oluşan bir işlemdir. Birinci adımda kullanılacak model oluşturulmaktadır. Veriler sınıflandırma algoritmasına tabi tutularak sınıflandırma kuralları oluşturulur. İkinci adımda ise oluşturulan sınıflandırma kuralları, sınıfı belli olmayan veriler üzerine uygulanarak sınıfı tahmin edilir (Han and Kamber 2001).

Şekil 3.5'de görüldüğü gibi sınıflandırma işlemi iki aşamadan oluşmaktadır. Birinci aşama (A) öğrenme aşamasıdır. Bu aşamada veriler sınıflandırma algoritmasına tabi tutularak sınıflandırma kuralları oluşturulur. İkinci aşama (B) sınıflandırma aşamasıdır. Bu aşamada ise birinci aşamada oluşturulan sınıflandırma kurallarının doğruluğu test verileri ile tespit edilir. Eğer sınıflandırma kurallarının doğruluğu kabul edilebilirse, yeni verinin kredi kararı, sınıflandırma kurallarına bakılarak bulunur.

Şekil 3.5'deki örnekte yeni veri Çağrı, Orta Yaş, Düşük şeklindedir. Bu verinin kredi kararını bulmak için sınıflandırma kurallarına bakarsak, yeni verimizin yaşı = orta yaş

ve geliri = düşük olduğundan 3.sınıflandırma kuralının kredi kararı yeni verimizin kredi kararı olur. Sonuç olarak yeni verimizin kredi kararı riskli olarak tahmin edilir.



Şekil 3.5. Sınıflandırma işlemi
***(A) Öğrenme aşaması, (B) Sınıflandırma Aşaması (Han and Kamber 2001)**

Sınıflandırma yapmak için birçok yöntem ve algoritma geliştirilmiştir. Bu yöntem ve algoritmaların en yaygın olanları karar ağaçları, istatistiğe dayalı algoritmalar, mesafeye dayalı algoritmalar, yapay sinir ağları ve genetik algoritmalardır (Silahtaroğlu 2008).

3.4.1. Karar ağaçları

Karar ağaçları veri sınıflandırma yöntemlerinden birisi olup sınıflandırma problemlerinde çokça kullanılmaktadır. Karar ağaçları yönetiminde ilk önce bir ağaç oluşturulur, daha sonra veri tabanındaki kayıtlar bu ağaca uygulanır ve çıkan sonuca göre de bu kayıtlar sınıflandırılır (Silahtaroğlu 2008).

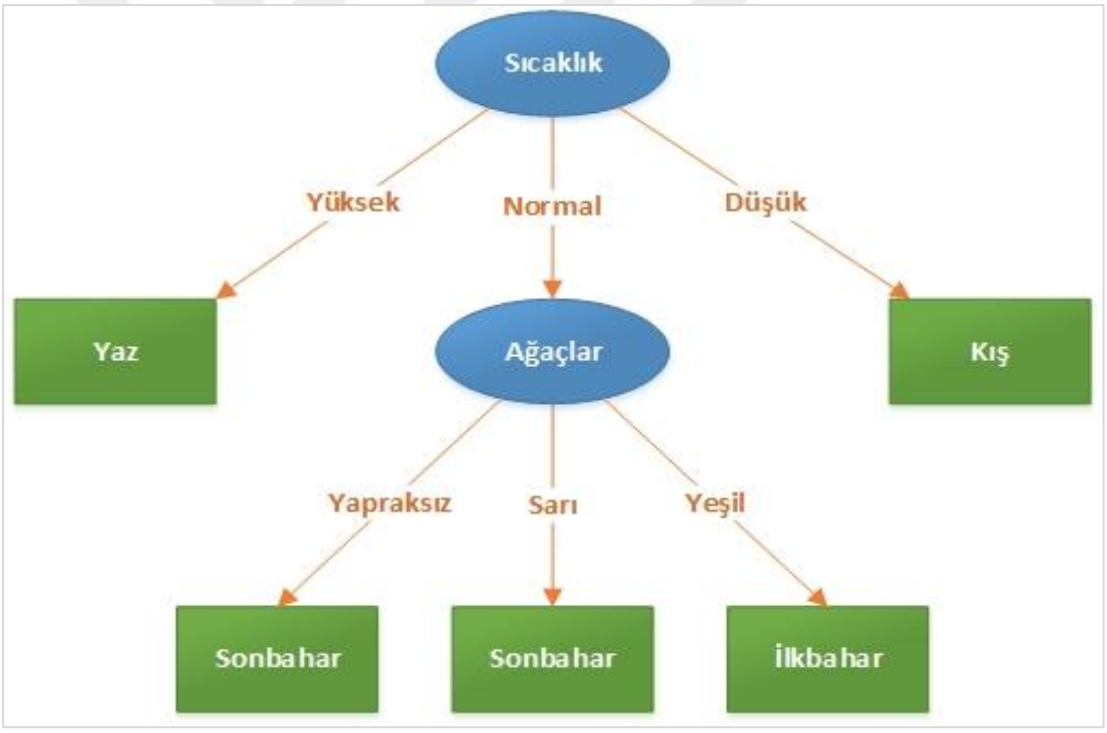
Karar ağaçlarının, kuruluşlarının ucuz olması, yorumlanmalarının kolay olması, veri tabanı sistemleri ile kolayca entegre edilebilmeleri, güvenilirliklerinin yüksek olması nedenleriyle sınıflandırma yöntemi içerisinde en yaygın kullanıma sahip tekniktir (Gürsoy 2009).

Karar ağaçları veri kümesindeki herhangi bir alana karşılık gelen karar düğümlerinden, o alandaki değerleri belirten dallardan ve sınıfı belirten yapraklardan oluşan, akış diyagramı şeklindeki ağaç yapılarıdır (Han and Kamber 2001).

Karar ağaçları, ağaç yapısında bir akış şemasına benzer. Ağaç üç tip düğümden oluşur (Tan *et al.* 2005).

- **Kök düğümü:** Karar ağacının tepesinde bulunur, kendisine gelen bir bağlantı (dal) bulunmazken kendisinden dışarıya bağlantı olabilir.
- **İç düğümler:** Kendisine gelen bir bağlantıya sahipken iki ya da daha fazla kendisinden giden bağlantıya sahip olan düğümlerdir.
- **Yaprak ya da uç düğümler:** Yaprak ya da uç düğümlerin kendisine gelen bir bağlantısı vardır, ancak kendisinden giden bağlantı bulunmaz.

Şekil 3.6'daki karar ağacı örneğinde görüldüğü gibi, karar ağacı, ağaç yapısında bir akış şemasıdır. Örnekte sıcaklık alanına sahip olan ve karar ağacının tepesinde bulunan düğüm karar ağacının kök düğümüdür. Düğümlerden ayrılan ve o düğümün özelliklerini belirten bağlantılara ise dal denilmektedir. Dikdörtgen şeklinde olan ve içerisinde sınıf alanının değerlerini bulunduran uç düğümler, yaprak olarak tanımlanmaktadır. Görüldüğü gibi yapraklara bir bağlantı gelmekte ancak kendisinden diğer bir düğüme bağlantı gitmemektedir. Bu örnekteki iç düğüm ise ağaçlar alanına sahip düğümdür. Görüldüğü gibi ağaçlar alanına bir bağlantı gelmekte, ağaçlar düğümünden ise bağlantılar diğer düğümlere gitmektedir. Bu özelliğinden dolayı ağaçlar düğümü bir iç düğümdür.



Şekil 3.6. Karar ağacı örneği

3.4.2. Karar ağaçlarında kullanılan algoritmalar

Karar ağaçlarında kullanılan algoritmalar aşağıdaki gibi gruplara ayrılabilir.

- Entropiye dayalı algoritmalar
- ID3
- C 4.5, C 5
- Sınıflandırma ve regresyon ağaçları
- Twoing
- Gini
- SLIQ (Supervised Learning In Quest – Araştırmada Denetimli Öğrenme)
- SPRINT (Scalable Parallelizable Induction of Decision Trees)

3.4.2.a. ID3 algoritması

ID3 algoritması Ross Quinlan tarafından geliştirilmiş bir karar ağacı algoritmasıdır (Quinlan 1986). ID3 algoritması C4.5 algoritmasının öncüsüdür ya da C4.5 algoritması ID3 algoritmasının gelişmiş versiyonudur denilebilir (Anonymous 2013). ID3 algoritması entropi tabanlı bir algoritmadır. Entropi veri kümesindeki belirsizliğin bir ölçüsüdür (Shannon 1948).

Şekil 3.7 gösterilen veri kümesinde adı, yaşı, geliri, kredi kararı alanları bulunmaktadır. Bu alanlardan kredi kararı alanı, veri kümesinin sınıfıdır. Eğer veri kümesindeki bütün verilerin kredi kararları riskli olsaydı entropi değeri yani belirsizlik sıfır (0) olurdu. Bunun tam tersi yani sınıf alanındaki farklı veriler eşit sayıda olsaydı (buradaki örnekte 3 tane riskli 3 tane güvenli sınıfı bulunmaktadır) entropi maksimum değer olan 1 olurdu. Bu durumda entropinin 0 ile 1 arasında değer aldığı anlaşılmaktadır. Entropinin 0 olması belirsizlik yok, entropinin 1 olması ise tam bir belirsizlik var anlamına gelmektedir (Silahtaroğlu 2008).

Şekil 3.7. Veri kümesi örneği

ID3 algoritması ile karar ağacı oluşturma

Çizelge 3.3'de gösterilen veri kümesine, ID3 algoritmasını uygulayarak veri kümesinin karar ağacı aşağıdaki gibi oluşturulabilir.

Çizelge 3.3. ID3 algoritması için örnek veri kümesi

NO	GELİR	YAŞ	BORÇ	PERFORMANS (Ödeme Performansı)	KARAR (Kredi kararı)
1	Yüksek	Genç	Var	Kötü	Riskli
2	Yüksek	Genç	Var	İyi	Güvenli
3	Yüksek	Orta	Var	Kötü	Güvenli
4	Orta	Yaşlı	Var	Kötü	Riskli
5	Düşük	Yaşlı	Yok	Kötü	Güvenli
6	Düşük	Yaşlı	Yok	İyi	Güvenli
7	Düşük	Orta	Yok	İyi	Güvenli
8	Orta	Genç	Var	Kötü	Riskli
9	Düşük	Genç	Yok	Kötü	Riskli
10	Orta	Yaşlı	Yok	Kötü	Güvenli
11	Orta	Genç	Yok	İyi	Güvenli
12	Orta	Orta	Var	İyi	Güvenli
13	Yüksek	Orta	Yok	Kötü	Güvenli
14	Orta	Yaşlı	Var	İyi	Güvenli

Önce veri kümesinin sınıfı belirlenir. Bu veri kümesinde sınıf karar alanıdır. Çünkü bu veri kümesiyle bir bankaya gelen müşteriye kredi verme riski ölçülmektedir. Sonrasında yapılacak ilk iş genel entropiyi hesaplamaktır.

Entropi $H(S) = -\sum_{i=1}^{n}(p_i \log_2(p_i))$ formülüyle hesaplanır (Shannon , 1948). Sınıf sayılarına bakılırsa 4 adet riskli sınıfından 10 adet ise güvenli sınıfından bulunduğu görülür. Bunların olasılıkları yani p_i değerleri $p_1 = \frac{4}{14}$, $p_2 = \frac{10}{14}$ şeklindedir. Bu durumda genel entropi aşağıdaki şekilde hesaplanır.

$$H(KARAR) = -\left(\frac{4}{14}\log_2\frac{4}{14} + \frac{10}{14}\log_2\frac{10}{14}\right) = 0.863$$

1. Adım: Karar Ağacının Kök Düğümünün Bulunması

Kök düğümünü bulabilmemiz için sınıf alanı hariç bütün alanların kazanç değerlerinin hesaplanması gerekmektedir. Kazancı en yüksek olan alan, karar ağacının kök düğümünü oluşturur. Kazanç, sınıf alanının entropisinden (genel entropi) hangi alan için kazanç hesaplanıyorsa o alanın entropisi çıkartılarak bulunur.

Kazancın Formülü

$$\text{Kazanç}(X,T) = H(T) - H(X,T)$$

$$H(X,T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} H(T_i)$$

a- Gelir alanı için kazanç hesabı

Yukarıda bulunan değerler kazanç formülünde yerine konularak gelir alanı için kazanç hesaplanır.

1. Bu aşamada ilk önce gelir alanındaki her bir verinin tekrar miktarı hesaplanır.

$\text{GELİR}_{\text{yüksek}} = 4\,,\ \text{GELİR}_{\text{orta}} = 6\,, \text{GELİR}_{\text{düşük}} = 4$

2. Gelir alanındaki her bir verinin entropisi hesaplanır.

$$H\left(\text{GELİR}_{\text{yüksek}}\right) = -\left(\frac{1}{4}\log_2\frac{1}{4} + \frac{2}{4}\log_2\frac{2}{4}\right) = 0.811$$

$$H\left(\text{GELİR}_{\text{orta}}\right) = -\left(\frac{2}{6}\log_2\frac{2}{6} + \frac{4}{6}\log_2\frac{4}{6}\right) = 0.918$$

$$H\left(\text{GELİR}_{\text{düşük}}\right) = -\left(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}\right) = 0.811$$

Geliri düşük olan verilerin entropisi hesaplanırken, veri kümesine bakıldığında geliri düşük olan 4 adet veri olduğu görülür. Bu 4 adet veriden de sadece 1 tanesi riskli sınıfına ait, 3 tanesi güvenli sınıfına aittir.

3. Gelir alanının entropisi hesaplanır.

$$\text{H}(\text{GELİR}, \text{KARAR}) = \left(\frac{4}{14} * 0.811 + \frac{6}{14} * 0.918 + \frac{4}{14} * 0.811\right) = 0.856$$

4. Bulunan değerler kazanç formülünde yerine konularak gelir alanı için kazanç hesaplanır.

$$\text{Kazanç}(\text{GELİR}, \text{KARAR}) = 0.863 - 0.856 = 0.007$$

b- Yaş alanı için kazanç hesabı

$\text{YAŞ}_{\text{genç}} = 5\ ,\ \text{YAŞ}_{\text{orta}} = 4\ ,\text{YAŞ}_{\text{yaşlı}} = 5$

$$\text{H}(\text{YAŞ}_{\text{genç}}) \quad = -\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right) = 0.971$$

$$\text{H}(\text{YAŞ}_{\text{orta}}) \quad = -\left(\frac{4}{4}\log_2\frac{4}{4}\right) = 0$$

$$\text{H}(\text{YAŞ}_{\text{yaşlı}}) \quad = -\left(\frac{1}{5}\log_2\frac{1}{5} + \frac{4}{5}\log_2\frac{4}{5}\right) = 0.722$$

$$\text{H}(\text{YAŞ}, \text{KARAR}) = \left(\frac{5}{14} * 0.971 + \frac{4}{14} * 0 + \frac{5}{14} * 0.722\right) = 0.605$$

$$\text{Kazanç}(\text{YAŞ}, \text{KARAR}) = 0.863 - 0.605 = 0.258$$

c- Borç alanı için kazanç hesabı

$\text{BORÇ}_{\text{var}} = 7\ ,\text{BORÇ}_{\text{yok}} = 7$

$$\text{H}(\text{BORÇ}_{\text{var}}) \quad = -\left(\frac{3}{7}\log_2\frac{3}{7} + \frac{4}{7}\log_2\frac{4}{7}\right) = 0.985$$

$$\text{H}(\text{BORÇ}_{\text{yok}}) \quad = -\left(\frac{1}{7}\log_2\frac{1}{7} + \frac{6}{7}\log_2\frac{6}{7}\right) = 0.592$$

$$\text{H}(\text{BORÇ}, \text{KARAR}) = \left(\frac{7}{14} * 0.985 \ + \ \frac{7}{14} * 0.592\right) = 0.788$$

$$\text{Kazanç}(\text{BORÇ}, \text{KARAR}) = 0.863 - 0.788 = 0.075$$

d- Performans alanı için kazanç hesabı

$\text{PERFORMANS}_{\text{iyi}} = 6\ ,\text{PERFORMANS}_{\text{kötü}} = 8$

$$H(\text{PERFORMANS}_{\text{iyi}}) = -\left(\frac{6}{6}\log_2\frac{6}{6}\right) = 0$$

$$H(\text{PERFORMANS}_{\text{kötü}}) = -\left(\frac{4}{8}\log_2\frac{4}{8} + \frac{4}{8}\log_2\frac{4}{8}\right) = 1$$

$$H(\text{PERFORMANS}, \text{KARAR}) = \left(\frac{6}{14} * 0 + \frac{8}{14} * 1\right) = 0.571$$

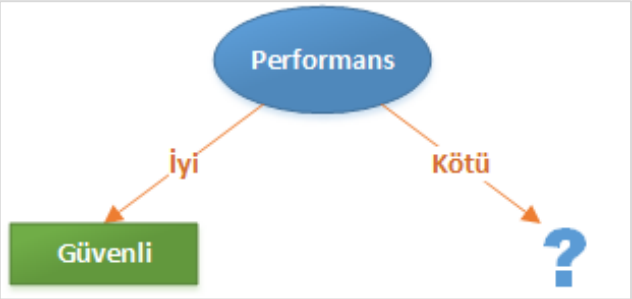
$$\text{Kazanç}(\text{PERFORMANS}, \text{KARAR}) = 0.863 - 0.571 = 0.292$$

Çizelge 3.4'e bakıldığında kazancı en fazla olan alanın performans alanı olduğu görülmektedir. Bu durumda performans alanı karar ağacımızın kök düğümünü oluşturur.

Çizelge 3.4. Birinci adımda oluşan kazanç değerleri

ALAN	KAZANÇ
Gelir	0.007
Yaş	0.258
Borç	0.075
Performans	**0.292**

Performans alanının iyi değerlerine bakıldığında, bütün iyi verilerinin sınıfının güvenli olduğu görülür. Bu durumda, performans alanı iyi değerinde dallanma yapmaz ve ağacın ilk yaprağını oluşturur. Karar ağacının ilk aşaması Şekil 3.8'deki gibi çizilir.



Şekil 3.8. Birinci adım sonucunda oluşan karar ağacı

2. Adım: Performans Alanının Kötü Değeri İçin Dallanma

Bu adımda birinci adımda yapılan işlemler tekrar edilir. Ancak veri kümesi Çizelge 3.5'de görüldüğü gibi sadece performansı kötü olan verileri içerir.

Çizelge 3.5. İkinci adımda oluşan veri kümesi

NO	GELİR	YAŞ	BORÇ	PERFORMANS	KARAR
1	Yüksek	Genç	Var	Kötü	Riskli
3	Yüksek	Orta	Var	Kötü	Güvenli
4	Orta	Yaşlı	Var	Kötü	Riskli
5	Düşük	Yaşlı	Yok	Kötü	Güvenli
8	Orta	Genç	Var	Kötü	Riskli
9	Düşük	Genç	Yok	Kötü	Riskli
10	Orta	Yaşlı	Yok	Kötü	Güvenli
13	Yüksek	Orta	Yok	Kötü	Güvenli

Karar sınıfına bakıldığında, 4 adet riskli sınıfına ait veri, 4 adet güvenli sınıfına ait veri olduğu görülmektedir. Bu durumda ikinci adımdaki genel entropi aşağıdaki şekilde hesaplanır.

$$\mathrm{H(KARAR)} = -\left(\frac{4}{8}\log_2\frac{4}{8} + \frac{4}{8}\log_2\frac{4}{8}\right) = 1$$

Bu adımda performans alanı kök düğümü olduğundan, performans alanı ve sınıf alanı hariç gelir, yaş ve borç alanları için kazanç değerleri hesaplanır.

a- Gelir alanı için kazanç hesabı

$$\mathrm{GELİR_{yüksek}} = 3\,,\ \mathrm{GELİR_{orta}} = 3\,, \mathrm{GELİR_{düşük}} = 2$$

$$\mathrm{H(GELİR_{yüksek})} = -\left(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right) = 0.918$$

$$\mathrm{H(GELİR_{orta})} = -\left(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right) = 0.918$$

$$\mathrm{H(GELİR_{düşük})} = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1$$

$$\mathrm{H(GELİR, KARAR)} = \left(\frac{3}{8} * 0.918 + \frac{3}{8} * 0.918 + \frac{2}{8} * 1\right) = 0.938$$

$$\mathrm{Kazanç(GELİR, KARAR)} = 1 - 0.938 = 0.062$$

b- Yaş alanı için kazanç hesabı

$$\mathrm{YAŞ_{genç}} = 3\ ,\ \mathrm{YAŞ_{orta}} = 2\ ,\ \mathrm{YAŞ_{yaşlı}} = 3$$

$$\mathrm{H(YAŞ_{genç})} = -\left(\frac{3}{3}\log_2\frac{3}{3}\right) = 0$$

$$\mathrm{H(YAŞ_{orta})} = -\left(\frac{2}{2}\log_2\frac{2}{2}\right) = 0$$

$$\mathrm{H(YAŞ_{yaşlı})} = -\left(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right) = 0.918$$

$$\mathrm{H(YAŞ, KARAR)} = \left(\frac{3}{8} * 0 + \frac{2}{8} * 0 + \frac{3}{8} * 0.918\right) = 0.344$$

$$\mathrm{Kazanç(YAŞ, KARAR)} = 0.863 - 0.344 = 0.656$$

c- Borç alanı için kazanç hesabı

$$\mathrm{BORÇ_{var}} = 4\ ,\ \mathrm{BORÇ_{yok}} = 4$$

$$\mathrm{H(BORÇ_{var})} = -\left(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}\right) = 0.811$$

$$\mathrm{H(BORÇ_{yok})} = -\left(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}\right) = 0.811$$

$$\mathrm{H(BORÇ, KARAR)} = \left(\frac{4}{8} * 0.811 + \frac{4}{8} * 0.811\right) = 0.811$$

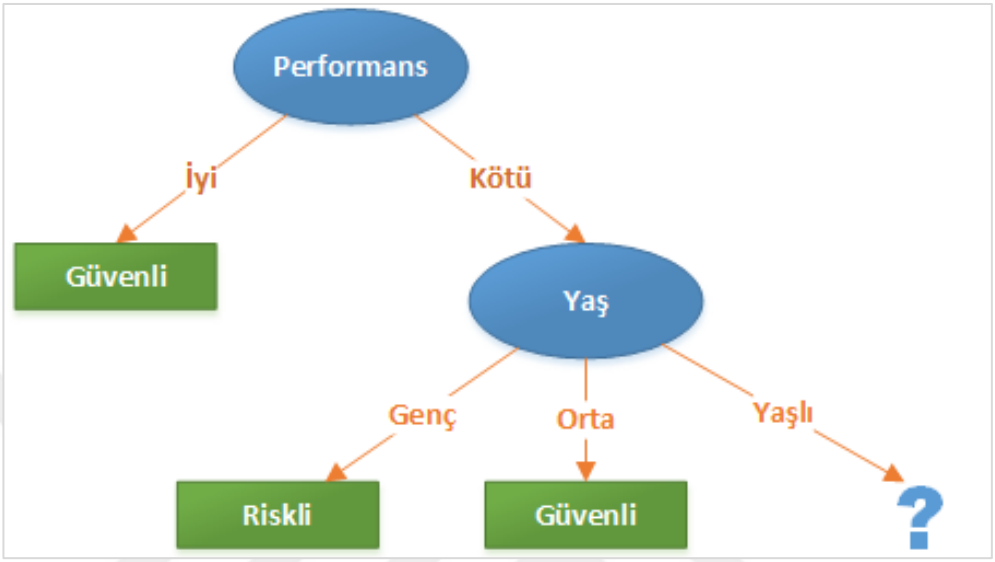
$$\mathrm{Kazanç(BORÇ, KARAR)} = 1 - 0.811 = 0.189$$

Çizelge 3.6'e bakıldığında kazancı en fazla olan alanın yaş alanı olduğu görülür.

Çizelge 3.6. İkinci adımda oluşan kazanç değerleri

ALAN	**KAZANÇ**
Gelir	0.062
Yaş	**0.656**
Borç	0.189

Bu durumda yaş alanı, karar ağacının ikinci düzeydeki düğümü olarak seçilir ve karar ağacı, ikinci aşama sonucunda Şekil 3.9'daki gibi çizilir.



Şekil 3.9. İkinci adım sonucunda oluşan karar ağacı

3. Adım: Yaş Alanının Yaşlı Değeri İçin Dallanma

Bu adımda birinci adımda yapılan işlemler tekrar edilir. Ancak veri kümesi Çizelge 3.7'de görüldüğü gibi sadece performans = kötü ve yaş = yaşlı olan verileri içerir.

Çizelge 3.7. Üçüncü adımda oluşan veri kümesi

NO	GELİR	YAŞ	BORÇ	PERFORMANS	KARAR
4	Orta	Yaşlı	Var	Kötü	Riskli
5	Düşük	Yaşlı	Yok	Kötü	Güvenli
10	Orta	Yaşlı	Yok	Kötü	Güvenli

Karar sınıfına bakıldığında 1 adet riskli sınıfına ait veri, 2 adet güvenli sınıfına ait veri olduğu görülmektedir. Bu durumda üçüncü adımdaki genel entropi aşağıdaki şekilde hesaplanır.

$$H(KARAR) = -\left(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right) = 0.918$$

Bu adımda, performans alanı kök düğümü ve yaş alanı ikinci düzeyde oluşan düğüm olduğundan, sadece gelir ve borç alanları için kazanç hesabı yapılır.

a- Gelir alanı için kazanç hesabı

$$\text{GELİR}_{\text{orta}} = 2\,,\ \text{GELİR}_{\text{düşük}} = 1$$

$$\text{H}(\text{GELİR}_{\text{orta}}) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1$$

$$\text{H}\left(\text{GELİR}_{\text{düşük}}\right) = -\left(\frac{1}{1}\log_2\frac{1}{1}\right) = 0$$

$$\text{H}(\text{GELİR}, \text{KARAR}) = \left(\frac{2}{3} * 1 + \frac{1}{3}\right) = 0.667$$

$$\text{Kazanç}(\text{GELİR}, \text{KARAR}) = 0.918 - 0.667 = 0.251$$

b- Borç alanı için kazanç hesabı

$$\text{BORÇ}_{\text{var}} = 1\,,\ \text{BORÇ}_{\text{yok}} = 2$$

$$\text{H}(\text{BORÇ}_{\text{var}}) = -\left(\frac{1}{1}\log_2\frac{1}{1}\right) = 0$$

$$\text{H}\left(\text{BORÇ}_{\text{yok}}\right) = -\left(\frac{2}{2}\log_2\frac{2}{2}\right) = 0$$

$$\text{H}(\text{BORÇ}, \text{KARAR}) = \left(\frac{1}{3} * 0 + \frac{2}{3} * 0\right) = 0$$

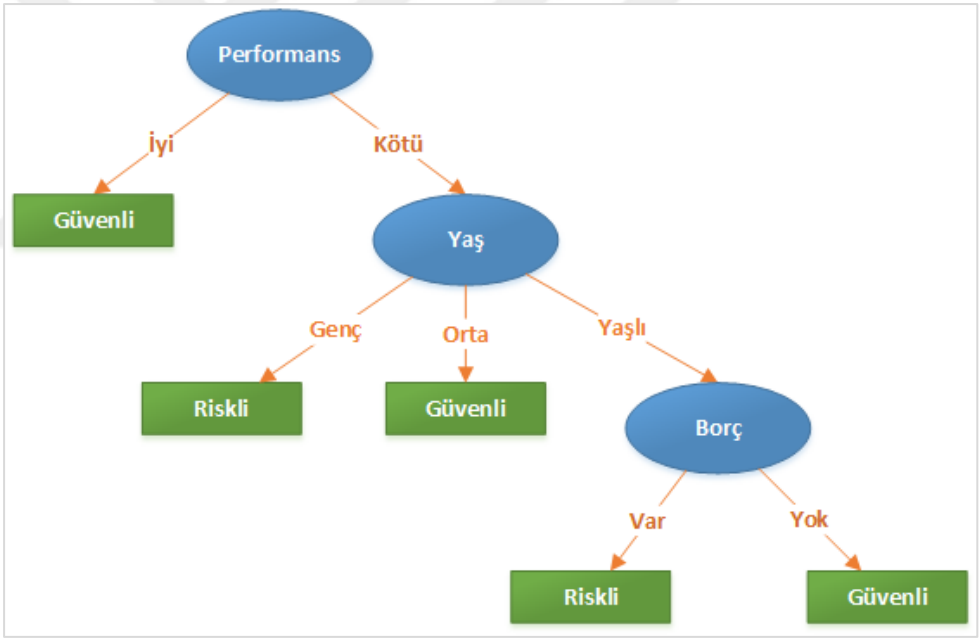
$$\text{Kazanç}(\text{BORÇ}, \text{KARAR}) = 0.918 - 0 = 0.918$$

Çizelge 3.8'e bakıldığında kazancı en fazla olan alanın borç alanı olduğu görülür.

Çizelge 3.8. Üçüncü adımda oluşan kazanç değerleri

ALAN	KAZANÇ
Gelir	0.251
Borç	**0.918**

Bu durumda borç alanı, karar ağacının, üçüncü düzeydeki düğümü olarak seçilir. Borç alanının değerlerine Çizelge 3.7'ye bakıldığında, "var" değeri için karar riskli, "yok" değerleri için ise karar güvenli olduğu görülür. Bu durumda ağaç başka dallanma yapmaz ve algoritma burada sonlandırılır. Algoritma sonucunda karar ağacı, Şekil 3.10'daki gibi oluşur.



Şekil 3.10. ID3 algoritması sonucunda oluşan karar ağacı

Bu karar ağacını kullanılarak, sınıfı belli olmayan bir verinin sınıfı tespit edilebilir. Gelir = Düşük, Yaş = Orta, Borç = Var, Performans = Kötü şeklinde Çizelge 3.3'deki veri kümesinde olmayan bir verinin sınıfını tahmin edilirken Şekil 3.10'daki karar ağacına bakılarak sınıf tahmin edilebilir. Yeni verinin performans alanının değeri "kötü" ve yaş alanının değeri "orta" olduğundan diğer alanlara bakılmadan yeni verinin sınıfı "güvenlidir" denilebilir.

3.4.2.b. C4.5 algoritması

C4.5 algoritması ID3 algoritması gibi Ross Quinlan tarafından geliştirilmiş bir algoritmadır. C4.5, ID3 algoritmasından daha üstün bir algoritma olup ID3 algoritmasının eksikliklerini gidermiştir (Quinlan 1996).

C4.5 algoritması ile birlikte ID3 algoritmasında olmayan bir takım özellikler gelmiştir. Bu özellikler;

- Karar ağacı oluşturulurken sayısal değerlere sahip alanlar kullanılabilir
- Kayıp veriler ile işlem yapılabilir
- Karar ağacı oluşturulduktan sonra budanabilir.

Sayısal Değerlere Sahip Niteliklerin Dönüştürülmesi

ID3 algoritması sadece kategori şeklinde (güvenli – riskli, iyi - kötü vs.) olan veriler ile çalışabilmektedir. ID3 algoritmasının bir üst versiyonu olan C4.5 algoritmasında bu problem giderilmiştir (Quinlan 1996).

Çizelge 3.3'de ID3 algoritmasını anlatılırken verilen veri kümesinin sayısal değerleri içeren şekli, performans alanı sayısallaştırılarak Çizelge 3.9'de verilmiştir. Performans alanının, kişinin borç ödeme performansını gösteren puanlama sistemi ile oluşturulduğu varsayılmaktadır.

Çizelge 3.9. Sayısal değerlere sahip veri kümesi

NO	GELİR	YAŞ	BORÇ	PERFORMANS	KARAR
1	Yüksek	Genç	Var	20	Riskli
2	Yüksek	Genç	Var	80	Güvenli
3	Yüksek	Orta	Var	30	Güvenli
4	Orta	Yaşlı	Var	30	Riskli
5	Düşük	Yaşlı	Yok	25	Güvenli
6	Düşük	Yaşlı	Yok	75	Güvenli
7	Düşük	Orta	Yok	80	Güvenli
8	Orta	Genç	Var	35	Riskli
9	Düşük	Genç	Yok	40	Riskli
10	Orta	Yaşlı	Yok	35	Güvenli
11	Orta	Genç	Yok	90	Güvenli
12	Orta	Orta	Var	95	Güvenli
13	Yüksek	Orta	Yok	25	Güvenli
14	Orta	Yaşlı	Var	70	Güvenli

C4.5 algoritması ID3 algoritmasından farklı olarak Çizelge 3.9'daki performans alanının sayısal verilerini dönüştürebilir. Dönüştürme işlemi için bir eşik değerinin hesaplanması gerekmektedir. Bu eşik değeri performans alanını iki ayrı kategoriye bölecektir ve bu eşik değeri hesaplanırken en büyük bilgi kazancının sağlanması hedef alınmalıdır. Eşik değeri hesaplanırken önce hesaplanan alanın veri kümesindeki bütün verileri $\{ v_1, v_2, v_3 \ldots . v_m \}$ şeklinde kümelenir. Buradaki örnekte küme { 20, 25, 30, 35, 40, 70, 75, 80, 90, 95 } şeklinde oluşturulur. Eşik değeri hesaplanırken aşağıdaki formül kullanılır.

$$t_i = \frac{v_i + v_{i+1}}{2}$$

Kümenin orta noktaları olan (40,70) değerleri formülde yerine koyulursa eşik değeri aşağıdaki gibi hesaplanır.

$$t = \frac{40 + 70}{2} = 55$$

Bu aşamada veri kümesindeki sayısal alanlar iki ayrı kategoriye ayırılmış olunur. Bu kategoriler Performans ≤55 ve Performans >55 şeklindedir. Bu aşamadan sonra veri kümesi Çizelge 3.10'daki gibi olur.

Çizelge 3.10. Veri kümesinin yeni görünümü

NO	GELİR	BORÇ	PERFORMANS (Ödeme Performansı)	KARAR
1	Yüksek	Genç	Var	Performans ≤55
2	Yüksek	Genç	Var	Performans >55
3	Yüksek	Orta	Var	Performans ≤55
4	Orta	Yaşlı	Var	Performans ≤55
5	Düşük	Yaşlı	Yok	Performans ≤55
6	Düşük	Yaşlı	Yok	Performans >55
7	Düşük	Orta	Yok	Performans >55
8	Orta	Genç	Var	Performans ≤55
9	Düşük	Genç	Yok	Performans ≤55
10	Orta	Yaşlı	Yok	Performans ≤55
11	Orta	Genç	Yok	Performans >55
12	Orta	Orta	Var	Performans >55
13	Yüksek	Orta	Yok	Performans ≤55
14	Orta	Yaşlı	Var	Performans >55

Kayıp Verilerin Giderilmesi

C4.5 algoritmasının, ID3 algoritmasından üstün olan bir diğer özelliği ise veri kümesine eksik girilmiş veriler ile işlem yapabiliyor olmasıdır. C4.5 algoritmasında eğer kayıp veriler varsa kazanç aşağıdaki formülle hesaplanır.

$$\text{Kazanç}(X,T) = F(H(T) - H(X,T))$$

$$F = \frac{\text{Veri kümesinde değeri bilinen alanların sahip olduğu veri sayısı}}{\text{Veri kümesindeki tüm verilerin sayısı}}$$

C4.5 algoritmasında kazanç hesabı aşağıdaki şekilde yapılır.

Çizelge 3.11'a bakılırsa 5 nolu verinin Borç alanında kayıp veri olduğu görülür. Borç alanının kazancı, ID3 algoritmasından farklı olarak kayıp veriler dikkate alınarak hesaplanmalıdır.

Çizelge 3.11. Kayıp veriye sahip veri kümesi

NO	GELİR	YAŞ	BORÇ	PERFORMANS (Ödeme Performansı)	KARAR (Kredi Kararı)
1	Yüksek	Genç	Var	20	Riskli
2	Yüksek	Genç	Var	80	Güvenli
3	Yüksek	Orta	Var	30	Güvenli
4	Orta	Yaşlı	Var	30	Riskli
5	Düşük	Yaşlı	?	25	Güvenli
6	Düşük	Yaşlı	Yok	75	Güvenli
7	Düşük	Orta	Yok	80	Güvenli
8	Orta	Genç	Var	35	Riskli
9	Düşük	Genç	Yok	40	Riskli
10	Orta	Yaşlı	Yok	35	Güvenli
11	Orta	Genç	Yok	90	Güvenli
12	Orta	Orta	Var	95	Güvenli
13	Yüksek	Orta	Yok	25	Güvenli
14	Orta	Yaşlı	Var	70	Güvenli

İlk önce veri kümesinin sınıfı olan karar alanının entropisi hesaplanır.

$$\mathrm{H(KARAR)} = -\left(\frac{4}{14}\log_2\frac{4}{14} + \frac{10}{14}\log_2\frac{10}{14}\right) = 0.863$$

Daha sonra Borç alanının entropisi hesaplanır.

$$\mathrm{H(BORÇ_{var})} = -\left(\frac{3}{7}\log_2\frac{3}{7} + \frac{4}{7}\log_2\frac{4}{7}\right) = 0.985$$

$$\mathrm{H\left(BORÇ_{yok}\right)} = -\left(\frac{1}{6}\log_2\frac{1}{6} + \frac{5}{6}\log_2\frac{5}{6}\right) = 0.650$$

$$\mathrm{H(BORÇ, KARAR)} = \left(\frac{7}{13} * 0.985 + \frac{6}{13} * 0.650\right) = 0.830$$

Son olarak kazanç F faktörüyle hesaplanır.

$$\mathrm{Kazanç(BORÇ, KARAR)} = \frac{13}{14}(0.863 - 0.830) = 0.030$$

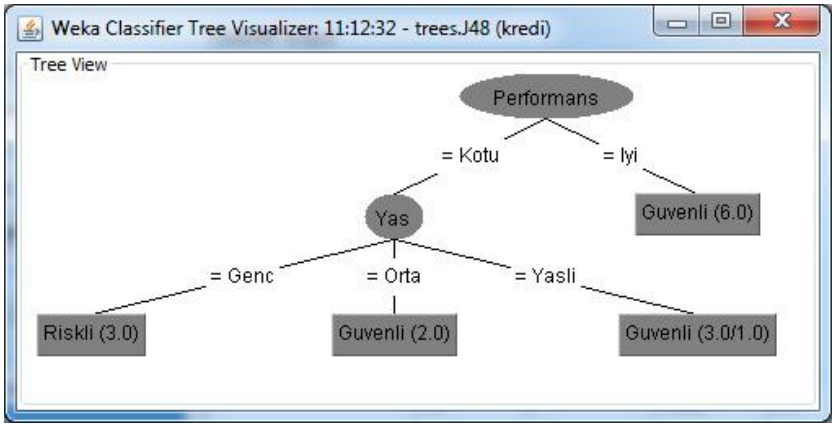
Böylelikle veri kümesinde bulunan kayıp veriler, ID3 algoritmasından farklı olarak C4.5 algoritmasında dikkate alınmıştır.

Karar Ağaçlarının Budanması

Bir yada birkaç alt ağacın yerine yaprakların koyulması işlemine karar ağaçlarında budama denilmektedir. Budama yapılarak algoritmanın tahmin yaparken daha az hata yapacağı umulurken, sınıflandırma modelinin kalitesinin artması beklenir (Kantardzic 2011).

Weka[1] programında Çizelge 3.3'de gösterilen ID3 algoritması anlatılırken kullanılan veri kümesine C4.5 algoritmasını uygulanırsa, Şekil 3.10 da gösterilen ID3 algoritması ile oluşturulmuş karar ağacının aksine Şekil 3.11'deki budanmış karar ağacını elde edilir. Weka programında C4.5 algoritmasının karşılığı J48'dir.

[1] Weka, Yeni Zellanda'daki Waikato Üniversitesindeki makine öğrenmesi grubu tarafından geliştirilmiş, makine öğrenimi algoritmalarının bir arada barındıran, veri hazırlama, sınıflandırma, regresyon, kümeleme, ilişki kuralları ve görselleştirme araçlarını içeren, açık kaynak kodlu bir veri madenciliği programıdır (Anonymous 2011).



Şekil 3.11. C4.5 algoritması sonucunda oluşan karar ağacı

Şekil 3.11'e bakılacak olunursa Şekil 3.10'daki karar ağacından farklı olarak Yaş = Yaşlı için alt ağaç bulunmamaktadır. Yani bu alt ağaç budanmıştır.

Kuralların oluşturulması

Şekil 3.11'deki karar ağacı göz önüne alınarak kurallar aşağıdaki gibi oluşturulur.

Kural 1 : Eğer Performans = İyi ise Karar = Güvenli
Kural 2 : Eğer Performans = Kötü ve Eğer Yaş = Genç ise Karar = Riskli
Kural 3 : Eğer Performans = Kötü ve Eğer Yaş = Orta ise Karar = Güvenli
Kural 4 : Eğer Performans = Kötü ve Eğer Yaş = Yaşlı ise Karar = Güvenli

3.4.2.c. Sınıflandırma ve regresyon ağaçları

Sınıflandırma ve regresyon ağaçları veri madenciliğinde sınıflandırma alanına giren ağaçlardır. Bu yöntem Breiman tarafından 1984 yılında çıkarılmıştır. Sınıflandırma ve regresyon ağaçları karar düğümünden itibaren ağacın iki dala ayrılması ile oluşturulur (Breiman 1998).

Sınıflandırma ve regresyon ağaçlarında bir düğümden sadece iki ayrı dal çıkar. Bu dallanma işlemi belirli kurallara göre yapılır. Bu kurallar Twoing ve Gini bölme kurallarıdır.

1. Twoing Bölme Kuralı

Twoing bölme kuralı Larose (2005) tarafından aşağıdaki gibi açıklanmıştır.

Eğitim kümesi aday bölünmeler denilen ikili bölümlere ayrılır. Bir t düğümünde eğitim kümesi iki ayrı dala ayrılır. Bu dallara aday bölünme denilir. Bu t düğümünde $t_{sağ}$ ve t_{sol} olmak üzere iki ayrı dal bulunur. Her bir aday bölünmenin uygunluk ölçüsü hesaplanır. Uygunluk ölçüsü aşağıdaki formül ile bulunur.

$$\Phi(s|t) = 2P_{sol}P_{sağ}\sum_{j=1}^{n}|P(j|t_{sol}) - P(j|t_{sağ})|$$

$$P_{sol} = \frac{t_{sol}\text{ daki herbir nitelik değerinin ilgili nitelik sütunundaki tekrar sayısı}}{\text{Eğitim kümesindeki kayıtların sayısı}}$$

$$P_{sağ} = \frac{t_{sağ}\text{ daki herbir nitelik değerinin ilgili nitelik sütunundaki tekrar sayısı}}{\text{Eğitim kümesindeki kayıtların sayısı}}$$

$$P(j|t_{sol}) = \frac{t_{sol}\text{ daki kayıtların j sınıfları sayısı}}{t_{sol}\text{ daki herbir nitelik değerinin ilgili nitelik sütunundaki tekrar sayısı}}$$

$$P(j|t_{sağ}) = \frac{t_{sağ}\text{ daki kayıtların j sınıfları sayısı}}{t_{sağ}\text{ daki herbir nitelik değerinin ilgili nitelik sütunundaki tekrar sayısı}}$$

Twoing Bölme Kuralı Kullanarak Karar Ağacı Oluşturma

Çizelge 3.12'deki veri kümesini kullanarak twoing bölünme kuralı aşağıdaki gibi açıklanabilir.

Çizelge 3.12. Twoing bölünme kuralı için eğitim kümesi

NO	GELİR	YAŞ	BORÇ	KARAR (Kredi kararı)
1	Normal	Orta	Yok	Güvenli
2	Yüksek	Genç	Yok	Güvenli
3	Düşük	Genç	Var	Riskli
4	Düşük	Yaşlı	Var	Riskli
5	Düşük	Yaşlı	Yok	Güvenli
6	Normal	Yaşlı	Yok	Güvenli
7	Düşük	Orta	Yok	Güvenli
8	Yüksek	Genç	Var	Riskli
9	Düşük	Genç	Yok	Riskli
10	Yüksek	Yaşlı	Yok	Güvenli

1.Adım

Eğitim kümesi Çizelge 3.13'de gösterildiği gibi ikili aday bölünmelere ayrılır.

Çizelge 3.13. Aday bölünmeler

Aday Bölünme	t_{sol}	$t_{sağ}$
1	GELİR = normal	GELİR ϵ {yüksek, düşük}
2	GELİR = düşük	GELİR ϵ {yüksek, normal}
3	GELİR = yüksek	GELİR ϵ {normal, düşük}
4	YAŞ = orta	YAŞ ϵ {genç, yaşlı}
5	YAŞ = genç	YAŞ ϵ {orta, yaşlı}
6	YAŞ = yaşlı	YAŞ ϵ {genç, orta}
7	BORÇ = var	BORÇ = yok
8	BORÇ = yok	BORÇ = var

2.Adım

Her bir aday bölünme için olasılık ve uygunluk ölçütleri hesaplanır (Burada sadece bir aday bölünmenin sol değeri için olasılık hesaplanacak diğerleri değerler ise tablo halinde verilecektir).

t_{sol} dalının GELİR = Normal alanının olasılığını aşağıdaki gibi hesaplanır.

$$P_{sol,gelir=normal} = \frac{2}{10} = 0.2$$

Gelir alanında toplam 2 adet normal değeri vardır ve eğitim kümesinde toplam 10 adet kayıt bulunmaktadır.

P(j | t_{sol}) değerini aşağıdaki gibi hesaplanır.

j değeri, sınıf alanına karşılık gelmektedir. Çizelge 3.12'deki veri kümesine bakıldığında, j değerinin güvenli yada riskli olacağı anlaşılır, bu durumda P(j | t_{sol}) aşağıdaki gibi hesaplanır.

$$P\ (G\ddot{U}VENL\dot{I} \mid t_{sol}) = \frac{2}{2} = 1$$

$$P\ (R\dot{I}SKL\dot{I} \mid t_{sol}) = \frac{0}{2} = 0$$

Eğitim kümesinde GELİR = Normal olup Sınıfı Güvenli olan 2 kayıt vardır. Aynı zamanda GELİR = Normal olan 2 kayıt vardır.

Aday bölünmelerin sağ ve sol tarafları için olasılıklar ve koşullu olasılıklar Çizelge 3.14'de ve Çizelge 3.15'de verilmiştir.

Çizelge 3.14. Sol aday bölünme ile ilgili olasılıklar

Aday Bölünme	T_{sol} daki kayıt sayısı	P_{sol}	Güvenli Sayısı	Riskli Sayısı	P(Güvenli | t_{sol})	P(Riskli | t_{sol})
1 (Gelir = normal)	2	0,2	2	0	1	0
2 (Gelir = düşük)	5	0,5	2	3	0,4	0,6
3 (Gelir = yüksek)	3	0,3	2	1	0,7	0,3
4 (Yaş = orta)	2	0,2	2	0	1	0
5 (Yaş = genç)	4	0,4	1	3	0,25	0,75
6 (Yaş = yaşlı)	4	0,4	3	1	0,75	0,25
7 (Borç = var)	3	0,3	0	3	0	1
8 (Borç = yok)	7	0,7	6	1	0,857	0,143

Çizelge 3.15. Sağ aday bölünme ile ilgili olasılıklar

Aday Bölünme	$T_{sağ}$ daki k. s.	$P_{sağ}$	Güvenli Sayısı	Riskli Sayısı	P(Güvenli | t_{sol})	P(Riskli | t_{sol})
1 GELİR ∈ {yüksek, düşük}	8	0,8	4	4	0,5	0,5
2 GELİR ∈ {yüksek, normal}	5	0,5	4	1	0,8	0,2
3 GELİR ∈ {normal, düşük}	7	0,7	4	3	0,571	0,429
4 YAŞ ∈ {genç, yaşlı}	8	0,8	4	4	0,5	0,5
5 YAŞ ∈ {orta, yaşlı}	6	0,6	5	1	0,83	0,17
6 YAŞ ∈ {genç, orta}	6	0,6	3	3	0,5	0,5
7 (Borç = yok)	7	0,7	6	1	0,857	0,143
8 (Borç = var)	3	0,3	0	3	0	1

Her bir aday bölünme için uygunluk değerinin hesaplanması

Sol ve sağ aday bölünmeler ile ilgili olasılıklar hesaplandıktan sonra her bir aday bölünme için uygunluk ölçütü hesaplanır. Uygunluk ölçüsü aşağıdaki formülle hesaplanır.

$$\Phi(s|t) = 2P_{sol}P_{sağ}\sum_{j=1}^{n}|P(j|t_{sol}) - P(j|t_{sağ})|$$

1.aday bölünme için uygunluk ölçütü aşağıdaki gibi hesaplanır.

$$\Phi(1|t) = 2 * 0{,}2 * 0{,}8 * (|1 - 0{,}5| + |0 - 0{,}5|) = 0{,}32$$

Tüm aday bölünmelerin uygunluk ölçütü Çizelge 3.16'da verilmiştir.

Çizelge 3.16. Her bir aday bölünme için uygunluk ölçütü

Aday Bölünme	P_{sol}	$P_{sağ}$	$2P_{sol}P_{sağ}$	$\Phi(s\|t)$
1	0,2	0,8	0,32	0,32
2	0,5	0,5	0,5	0,4
3	0,3	0,7	0,42	0,11
4	0,2	0,8	0,32	0,32
5	0,4	0,6	0,48	0,56
6	0,4	0,6	0,48	0,24
7	0,3	0,7	0,42	0,72
8	0,7	0,3	0,42	0,72

En büyük uygunluk ölçütünün seçilmesi

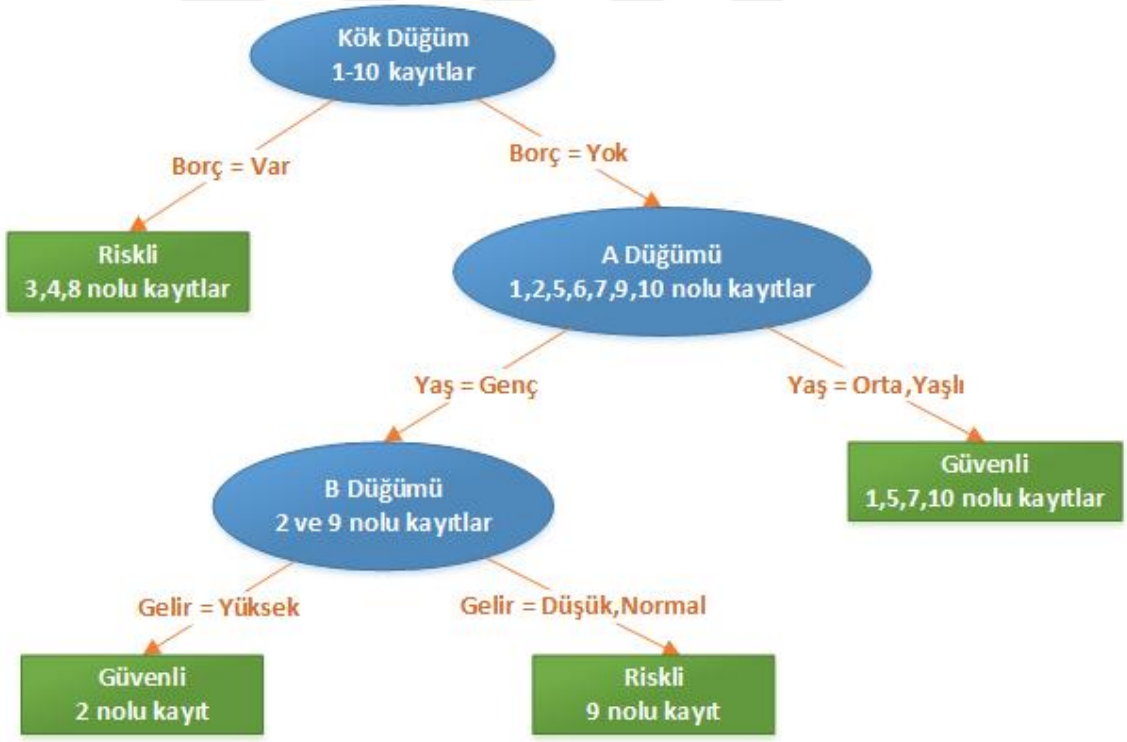
Çizelge 3.16'da görüldüğü gibi uygunluk ölçütü en büyük olan aday bölünmeler 0,72 değerleriyle 7. ve 8. satırdaki aday bölünmelerdir. Bu durumda karar ağacı Borç = Var ve Borç = Yok şeklinde iki alt ağaca ayrılır. Çizelge 3.12'ye bakıldığında Borç = Var niteliğini kapsayan kayıtlar 3, 4 ve 8 nolu kayıtlardır. 1, 2, 5, 6, 7, 9 ve 10 nolu kayıtlar ise Borç = Yok niteliğini kapsar. Böylelikle karar ağacı (3, 4, 8) ve (1, 2, 5, 6, 7, 9) olarak iki ayrı dala ayrılır.

Karar ağacının oluşturulması

Çizelge 3.12'ye bakıldığında Borç = Var niteliği için tüm sınıflar Riskli olduğundan 3, 4 ve 8 nolu kayıtların oluşturduğu dallanma yaprağa dönüşür. Geriye kalan 1, 2, 5, 6, 7 ve 9 nolu kayıtlar ise ayrı bir karar düğümü oluşturur ve bu karar düğümü için algoritmadaki 1. ve 2. adımlar ağaç tamamlanana kadar tekrarlanır.

Şekil 3.12. 1.döngü sonrası oluşan karar ağacı

Yukarıdaki şekildeki karar ağacının karar düğümüne twoing bölünme kuralının 1. ve 2. adımındaki işlemler tekrardan uygulanır ve bu işlemler tüm yapraklar oluşturuluncaya kadar devam eder. Sonuç olarak aşağıdaki karar ağacı elde edilir.



Şekil 3.13. Twoing bölünme kuralı nihai karar ağacı

2. Gini Bölünme Kuralı

Bu bölünme kuralı da twoing bölünme kuralı gibi örnek kümesini sadece iki farklı dala ayırır. Gini bölme kuralı Özkan (2008) tarafından aşağıdaki gibi açıklanmıştır.

a) Her nitelik değeri ikili biçimde gruplandırılır.
b) Her bir nitelik ile ilgili sol ve sağ taraftaki bölünmeler için $Gini_{sol}$ ve $Gini_{sağ}$ değerleri aşağıdaki formüle göre hesaplanır.

$$Gini_{sol} = 1 - \sum_{i=1}^{k} \left(\frac{L_i}{|T_{sol}|}\right)^2$$

$$Gini_{sağ} = 1 - \sum_{i=1}^{k} \left(\frac{R_i}{|T_{sağ}|}\right)^2$$

Yukarıdaki formüllerde yer alan ifadelerin açıklamaları şu şekildedir

k	Sınıfların sayısı
T	Bir düğümdeki örnekler
$\|T_{sol}\|$	Sol taraftaki örneklerin sayısı
$\|T_{sağ}\|$	Sağ taraftaki örneklerin sayısı
L_i	Sol taraftaki i kategorisindeki örneklerin sayısı
R_i	Sağ taraftaki i kategorisindeki örneklerin sayısı

c) j nitelik ve n eğitim kümesindeki satır sayısı olmak üzere aşağıdaki formülle tüm niteliklerin Gini değerleri hesaplanır.

$$Gini_j = \frac{1}{n}\left(|T_{sol}|Gini_{sol} + |T_{sağ}|Gini_{sağ}\right)$$

d) Her nitelik için hesaplanan Gini değerleri arasından en küçük olan seçilir ve bölünme bu nitelik üzerinde gerçekleştirilir.
e) a adımına dönülerek geriye kalan kayıtlarla işlem tekrar edilir.

3.4.2.d. SLIQ

SLIQ algoritması hem sayısal hem de kategorik verileri sınıflandırabilen bir algoritmadır. Algoritma karar ağacının büyüme aşamasında, sayısal verilerin değerlendirilmesindeki maliyeti azaltmak için sayısal verileri sınıflandırmadan önce bu verileri küçükten büyüğe doğru sıraya dizer. SLIQ algoritması CART ve C4.5 algoritmalarının aksine her düğümde sıralama yapmaz, sadece ağacın büyüme aşamasının başlangıcında yapar (Manish *et al.* 1996).

SLIQ algoritması ağacı dallara ayırırken Gini bölünme kuralını kullanırken ağacın budanması işleminde MDL[2] ilkesini kullanır.

3.4.2.e. SPRINT

SPRINT algoritması SLIQ algoritması gibi çalışır ancak SPRINT algoritmasında SLIQ algoritmasının eksiklikleri giderilmiştir.

SLIQ algoritmasında sınıflandırılacak her kayıt başına bir takım verilerin hafızada bulunması gerekmektedir. Sınıflandırılacak veriler ne kadar fazla olursa SLIQ algoritmasında hafızada tutulması gereken veriler artmaktadır. Bu durum ise SLIQ algoritmasında sınıflandırılacak veriyi kısıtlamaktadır. SPRINT algoritması bu kısıtlamaları kaldırdığı gibi hızlı ve ölçeklenebilir bir algoritmadır (Shafer *et al.* 1996).

SPRINT algoritması ile karar ağacı oluşturma

Shafer *et al.* (1996) SPRINT algoritmasıyla karar ağacı oluşturmayı aşağıdaki gibi açıklamışlardır.

Adım1

[2] Minimum Description Length (En Küçük Uzunluk Tanımlaması) (Rissanen 1989)

Çizelge 3.17'deki eğitim kümesinin her bir niteliği için Çizelge 3.18'deki gibi birer nitelik tablosu oluşturulur. Eğer nitelik sayısal değerlerden oluşuyorsa bu değerler küçükten büyüğe doğru sıralanır. Kategorik değerler sıralanmaz.

Çizelge 3.17. SPRINT algoritması için eğitim kümesi

rid	**Yaş**	**Araç Tipi**	**Risk**
0	23	Aile	Yüksek
1	17	Spor	Yüksek
2	43	Spor	Yüksek
3	68	Aile	Düşük
4	32	Kamyon	Düşük
5	20	Aile	Yüksek

Çizelge 3.18. Birinci düğüm için değişken listesi

Yaş	Sınıf	Tid
17	Yüksek	1
20	Yüksek	5
23	Yüksek	0
32	Düşük	4
43	Yüksek	2
68	Düşük	3

Araç Tipi	Sınıf	Tid
Aile	Yüksek	0
Spor	Yüksek	1
Spor	Yüksek	2
Aile	Düşük	3
Kamyon	Düşük	4
Aile	Yüksek	5

Adım 2

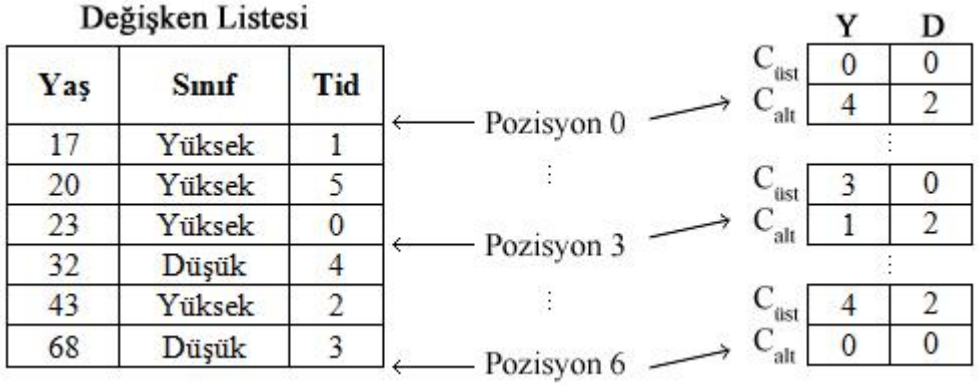
Eğitim kümesinde görüldüğü gibi yaş ve araç tipi diye iki adet nitelik değeri bulunmaktadır. Kök düğümünün hesaplanması için bu iki değer Gini bölünme kuralına tabi tutulur. Sonuç olarak yaş alanı kök düğüm olarak bulunur.

Adım 3

Yaş alanı sayısal verilere sahip olduğu için yaş alanı için bir bölme noktasının hesaplanması gerekmektedir.

Sayısal verilerin için bölme noktasının hesaplanması

Şekil 3.14'de gösterildiği gibi veri kümesi baştan sonra kadar taranır. Her bir verinin üstündekiler ve altındakiler olarak iki gruba ayrıldıktan sonra bu iki ayrım Gini bölünme kuralına tabi tutulur. En küçük Gini değerine sahip olan pozisyon bölme pozisyonu olarak kabul edilir. O pozisyonun bir üstündeki ve bir altındaki değerin aritmetik ortalaması bölünme değeri olarak kabul edilir.



Şekil 3.14. SPRINT algoritmasında sayısal verilerin bölme noktasının bulunması

Bu örnekte en küçük Gini değerine sahip olan pozisyon 3 nolu pozisyon olup, bölme noktası 0 ve 4 nolu kayıtların yaş değerlerinin aritmetik ortalaması olan 27,5 olarak hesaplanır.

Adım 4

Bu adımda karar ağacının birinci seviye düğümleri oluşturulur. 3. Adımda bulunan bölünme noktası ile veri kümesi iki alt düğüme bölünür. 27,5 değerinden küçük olan 1, 5 ve 0 nolu kayıtlar bir düğüm, 4, 2 ve 3 nolu kayıtlar ise başka bir düğüm oluştururlar.

1, 5 ve 0 nolu kayıtların sınıflarının hepsi tek bir sınıfa ait olduğundan bu alt bölünme bir başka düğüm değil, yaprak oluşturur.

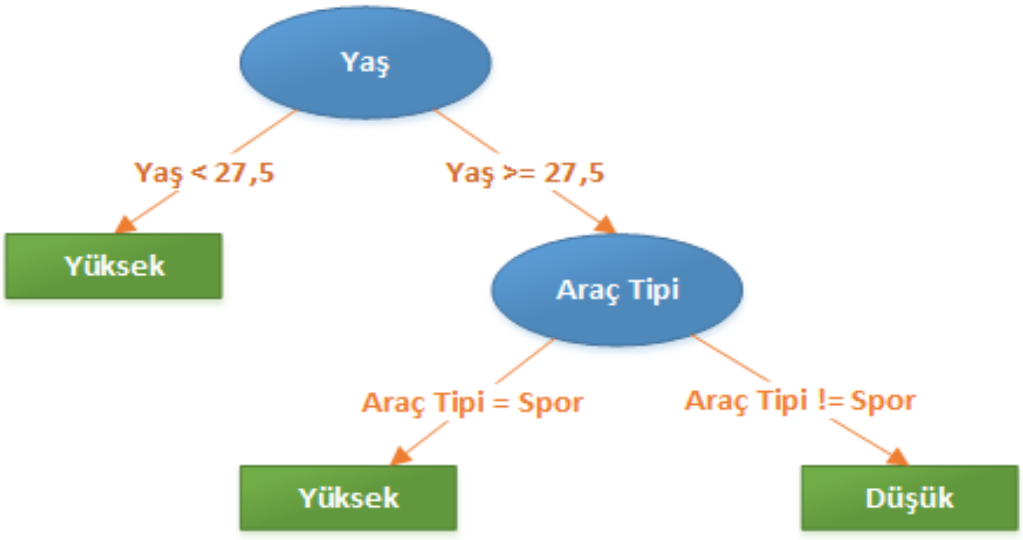
Kategorik verilerin bölme noktasının hesaplanması

Kategorik verilerin bölme noktası hesaplanırken, tüm veri kümesi baştan sonra kadar taranır. Tüm olası ikili bölünmeler bulunur.

Bu örnekte oluşan olası bölünmeler;

1 - Araç Tipi = Spor, Araç Tipi ϵ {Kamyon, Aile}
2 - Araç Tipi = Aile, Araç Tipi ϵ {Kamyon, Spor}
3 - Araç Tipi = Kamyon, Araç Tipi ϵ {Spor, Aile}

şeklindedir. Bu olası bölünmelere Gini bölünme kuralı uygulanır ve en düşük Gini değerine sahip olan bölünme kabul edilir. 1 nolu bölünme en düşük Gini değerine sahip olduğu için seçilir. Ağacın 3 nolu düğümü Araç Tipi = Spor, Araç Tipi ϵ {Kamyon, Aile} şeklinde bölünür ve sonuç olarak Şekil 3.15'deki gibi bir karar ağacı ortaya çıkar.



Şekil 3.15. SPRINT algoritması ile oluşan karar ağacı

3.4.3. Mesafeye dayalı sınıflandırma algoritmaları

Mesafeye dayalı sınıflandırma algoritmaları, sınıflandırma yaparken, sınıflandırılacak verinin, veri kümesindeki diğer verilere olan uzaklığını kullanarak sınıflandırma yaparlar. Uzaklık ölçülürken genellikle Öklid uzaklık bağlantısı kullanılır. Mesafeye dayalı sınıflandırma algoritmalarından en çok bilineni k en yakın komşu algoritmasıdır.

3.4.3.a. k en yakın komşu algoritması

Bu algoritma sınıflandırılması istenilen yeni veri için eğitim kümesinde bulunan ve yeni veriye en yakında bulunan k tane verinin sınıfına bakarak yeni verinin sınıfını tespit eder.

K en yakın komşu algoritmasının adımlarını Özkan (2008) aşağıdaki gibi açıklamıştır.

1. k parametresi belirlenir. K parametresi sınıfı tespit edilecek veriye en yakın komşuların sayısıdır.
2. Eğitim kümesindeki veriler koordinat düzlemi üzerinde listelenir. Sınıfı tespit edilecek olan verinin diğer verilere olan uzaklıkları tek tek hesaplanır.
3. 2. adımda hesaplanan uzaklıklardan en küçük olan k tane veri seçilir.
4. Seçilen verilerin sınıfları belirlenir ve en çok tekrar eden sınıf değeri seçilir.
5. Seçilen sınıf tahmin edilmesi beklenen yeni verinin sınıfı olarak atanır.

k en yakın komşu algoritması sınıfı tespit edilecek veriye en yakın komşuları bulmak için uzaklıkları hesaplarken formülü aşağıda bulunan öklid uzaklık bağıntısını kullanır.

$$\mathrm{d}\,(\mathrm{i},\mathrm{j}) = \sqrt{\sum_{k=1}^{p} \left(x_{ij} - x_{jk}\right)^2}$$

k En Yakın Komşu Algoritması Örnek Uygulama

Çizelge 3.19'daki eğitim kümesini kullanarak (9,5) noktası yani x1 = 9, x2 = 5 verisi için sınıf tayini aşağıdaki gibi yapılır.

Çizelge 3.19. k en yakın komşu algoritması için eğitim verisi

No	X1	X2	Sınıf
1	3	4	Evet
2	4	7	Hayır
3	4	4	Hayır
4	5	11	Evet
5	6	9	Evet
6	8	4	Hayır
7	10	8	Hayır
8	11	2	Evet

Adım 1

k değeri 3 olarak belirlenir. Böylelikle (9,5) noktasına en yakın 3 noktanın sınıflarının bulunması gerekir.

Adım 2

(9,5) noktasından veri kümesindeki tüm noktalara olan uzaklıklar Öklid uzaklık formülü ile hesaplanır. Çizelge 3.20'de uzaklıklar verilmiştir.

Çizelge 3.20. Eğitim kümesindeki verilerin (9,5) noktasına olan uzaklıkları

No	X1	X2	Sınıf	Uzaklık
1	3	4	Evet	6,08
2	4	7	Hayır	5,39

Çizelge 3.20. Eğitim kümesindeki verilerin (9,5) noktasına olan uzaklıkları (Devam)

3	4	5	Hayır	5,10
4	5	11	Evet	7,21
5	6	9	Evet	5
6	8	4	Hayır	1,41
7	10	8	Hayır	3,16
8	11	2	Evet	3,61

Adım 3

İkinci adımda hesaplanan uzaklık değerlerinden en küçük olan k(3) tanesi seçilir. Çizelge 3.20'de 6, 7 ve 8 nolu kayıtların uzaklık değerleri en küçüktür yani en yakın 3 adet komşudur.

Adım 4

Üçüncü adımda seçilen kayıtların sınıf değerleri bulunur. Çizelge 3.20'ye bakıldığında 6 nolu kaydın sınıfı Hayır, 7 nolu kaydın sınıfı Hayır, 8 nolu kaydın sınıfı ise Evet'dir. En çok tekrar eden sınıf Hayır sınıfıdır.

Adım 5

Bu adımda sınıflandırılması istenilen (9,5) noktasının sınıfı 4. Adımda en çok tekrar eden sınıfın değeri olan Hayır değeri olarak atanır.

Verilerin Dönüştürülmesi

K en yakın komşu algoritmasında bazen, veriler arasında büyük uzaklık farkları olabilir. Bu farkların giderilmesi için veriler (0,1) aralığındaki değerlere dönüştürülür. Bu işlem min-max normalleştirme yöntemiyle yapılır. Veri kümesindeki tüm verilere aşağıdaki formül uygulanarak veriler normalleştirirler.

$$X_{yeni} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Ağırlıklı Oylama

k en yakın komşu algoritmasının 5. adımında sınıflandırılması istenilen noktanın sınıfı tespit edilirken, tekrar sayısı fazla olan sınıf kabul edilmektedir. Ağırlıklı oylama yönteminde bu durum değişmektedir. Tekrar eden sınıfların ağırlıkları hesaplanır ve ağrılığı fazla olan sınıf, yeni gelen verinin sınıfı olarak atanır. Sınıfların ağırlıkları ise aşağıdaki formülle hesaplanır.

$$d(i,j)' = \frac{1}{d(i,j)^2}$$

Burada bulunan $d(i,j)$ ifadesi i ve j verileri arasındaki Öklid uzaklığıdır.

3.4.4. İstatistiğe dayalı sınıflandırma algoritmaları

3.4.4.a. Bayesian sınıflandırma algoritması

Bu sınıflandırma tekniği, veri kümesindeki tüm verilere bakar ve yeni verinin hangi sınıfa ait olması gerektiğinin olasılığını hesaplar. Olasılığı en yüksek olan sınıf, yeni verinin sınıfı olarak atanır. Olasılık hesaplanırken Bayes teoremi kullanılır. Bayes teoremi aşağıdaki gibi ifade edilebilir (Anonymous 2012).

$$P(C_1|x_i) = \frac{P(x_i|C_i)P(C_1)}{P(x_i|C_1)P(C_1) + P(x_i|C_2)P(C_2)}$$

- C_1 ve C_2 iki ayrı sınıf
- $P(x_i|C_1)$, x_i nin C_1 sınıfında olma olasılığı
- $P(C_1)$, C_1 sınıfının veri kümesinde bulunma sıklığı

Bayes Sınıflandırma algoritmasının yalancı kodu aşağıdaki gibidir.

Girdiler: Veri kümesi: D
Sınıflar: C_i, i = 1, 2, 3 … n
Sınıflandırılacak veri: R

BAŞLA

DO FOR each C;

Hesapla => $P(C_j|x_i) = \frac{P(x_i|C_j)P(C_j)}{P(x_i)}$

LOOP

R'nin sınıfına en büyük olasılığa sahip olan C'nin sınıfını ata

BİTİR

3.4.4.b. Regresyon

Regresyon analizi bağımlı bir değişken ile bağımsız değişken yada değişkenler arasındaki ilişkinin matematiksel bir denklem şeklinde yazılmasıdır (Taha 1968).

Regresyon analizi sadece bir bağımlı değişken ile bir bağımsız değişken arasında gerçekleşiyorsa, basit regresyon analizi olarak ifade edilirken, birden fazla bağımsız değişken arasında gerçekleşiyorsa, çoklu regresyon analizi olarak ifade edilir. Ayrıca regresyon analizi kurulan denkleme göre de doğrusal ya da doğrusal olmayan regresyon analizi olarak ikiye ayrılabilir (Silahtaroğlu 2013).

x ile bağımlı değişken y arasındaki çoklu regresyon formülü $y = a + b_1x_1 + b_2x_2 + \cdots + b_nx_n + \varepsilon$ şeklinde yazılabilir. Bu denkleme veri madenciliği gözlüğü ile bakıldığında y sınıfları temsil ederken x ise sınıflandırılacak verinin değerini temsil eder.

3.4.4.c. CHAID algoritması

CHAID algoritması aslında bir karar ağacı algoritmasıdır. Ancak istatistikte bilinen Ki-kare testi yöntemi, algoritmanın esasını oluşturduğu için istatistiğe dayalı sınıflandırma algoritmaları arasında yerini almıştır. CHAID algoritması, 1980 yılında Gordon V. Kass tarafından geliştirilmiştir. CART algoritması gibi ikili karar ağaçları üreten bir algoritmadır (Kass 1980).

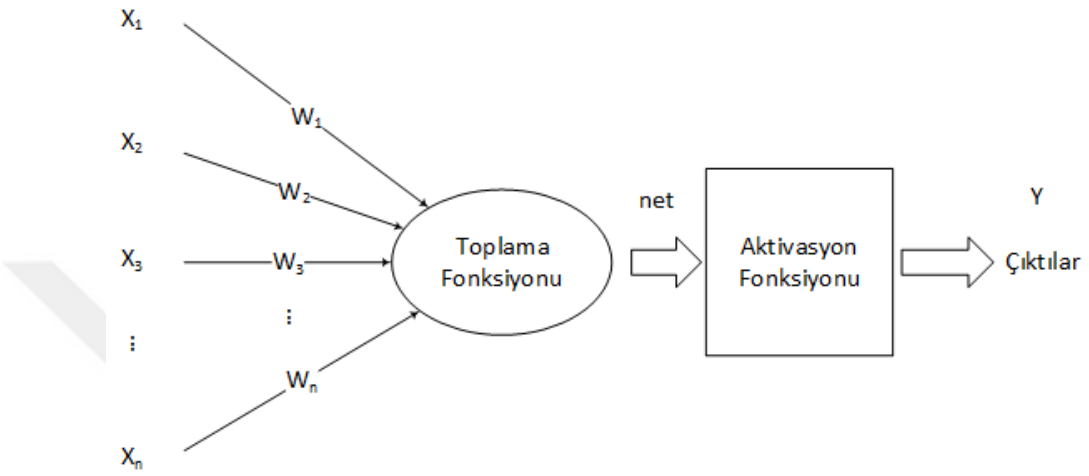
CHAID algoritmasının yalancı kodu aşağıdaki gibidir.

```
Girdiler:    Veri Kümesi
             Değişken Sayısı
             Her bir değişkendeki farklı değer sayısı
BAŞLA
  DO FOR her bir tahminleyici değişken (X)
    IF (X > iki farklı değer)
    DO UNTIL (x = iki faklı değer)
          Tüm değerleri Diğer Değerlerle ikili Ki-Kare testine sok ve en büyük p
                değeri veren ikilileri tek değer olarak etiketle
    Tüm iki değerli değişkenler için sınıf değişkenine göre ki-kare değerini hesapla
    En küçük p değerine sahip değişkenden dal yarat
    LOOP
BİTİR
```

3.4.5. Yapay sinir ağları

Yapay sinir ağları canlı organizmalarda bulunan biyolojik sinir yapısından esinlenerek gerçekleştirilmiştir. Canlılarda bulunan biyolojik sinir sistemlerinin öğrenme özelliklerini örnek olarak çalışan bir bilgi işleme modelidir. Biyolojik sinir sistemlerinin öğrenme, ilişkilendirme, sınıflandırma, genelleme, tahmin gibi işlemlerini gerçekleştirmeyi hedefler.

Şekil 3.16'da görüldüğü gibi, yapay sinir ağlarının yapı taşı olan yapay sinir hücreleri, girdiler, ağırlıklar, birleştirme fonksiyonu, aktivasyon fonksiyonu ve çıktılar olmak üzere beş temel bölümden oluşur (Gürsoy 2009).



Şekil 3.16. Yapay sinir hücresi

Bütün sinir ağı modellerinde X_i giriş değerleri, W_i ağırlık kat sayıları ile çarpılarak toplanır ve net denilen çıkış bulunur:

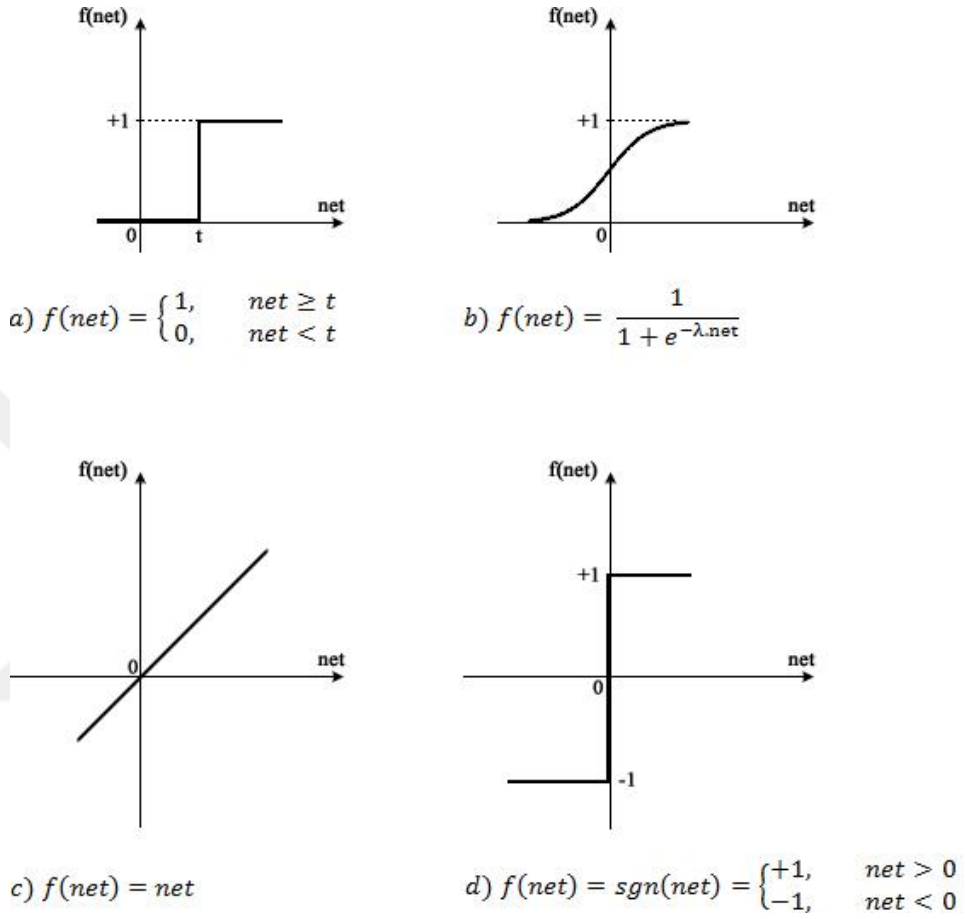
$$net = \sum_i x_i w_i = X.W$$

Yapay sinir hücresi ikili biçimde ifade edileceği zaman net değeri belirli bir eşik değerinden geçirilerek, eşiğin üstünde ise 1, altında ise 0 değeri üretilir.

$$y = f(X.W) = f(net)$$

Burada f fonksiyonu eşikleme yani aktivasyon fonksiyonudur. En çok kullanılan aktivasyon fonksiyonları olan step (a), signum (d), sigmoid (b) ve doğrusal (c) aktivasyon fonksiyonları, Şekil 3.17'de gösterilmiştir. Bunlardan step ve signum

fonksiyonları genellikle örüntü tanıma ve sınıflandırma işlemlerinde kullanılmaktadır (Nabiyev 2005).



$a)\ f(net) = \begin{cases} 1, & net \geq t \\ 0, & net < t \end{cases}$

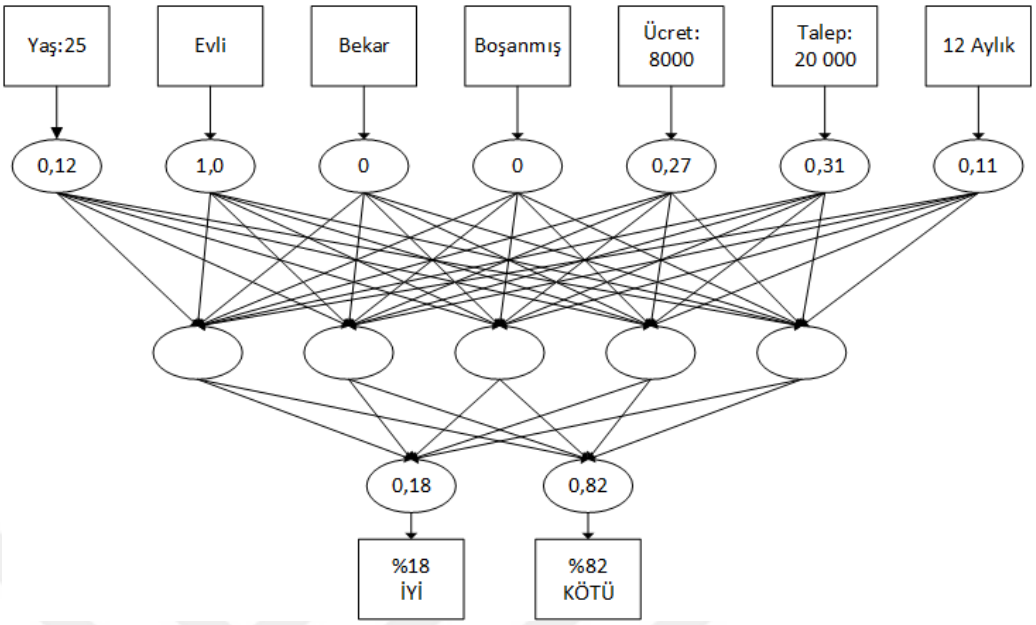
$b)\ f(net) = \dfrac{1}{1 + e^{-\lambda . net}}$

$c)\ f(net) = net$

$d)\ f(net) = sgn(net) = \begin{cases} +1, & net > 0 \\ -1, & net < 0 \end{cases}$

Şekil 3.17. Yapay sinir ağlarında kullanılan aktivasyon fonksiyonları (Nabiyev 2005)

Bir yapay sinir ağı uygulaması, Gürsoy (2009) tarafından Şekil 3.18'deki gibi gösterilmiştir. Uygulamada bir bankanın müşterilerinin kredi başvurularında oluşabilecek risk tahmin edilmektedir. Müşteri bilgileri alınmakta ve bir tahmin modeli kurulduktan sonra yeni başvurular değerlendirilirken bu modelden faydalanılmaktadır.



Şekil 3.18. Yapay sinir ağı uygulaması (Gürsoy 2009)

3.4.6. Genetik algoritmalar

Genetik algoritmalar ilk defa John Holland tarafından 1975 yılında geliştirilmiştir. Holland, Darwin'in evrim - "doğada en iyinin yaşaması" kuramından yola çıkmıştır ve bu süreci bilgisayar ortamına aktarmıştır (Elmas 2007).

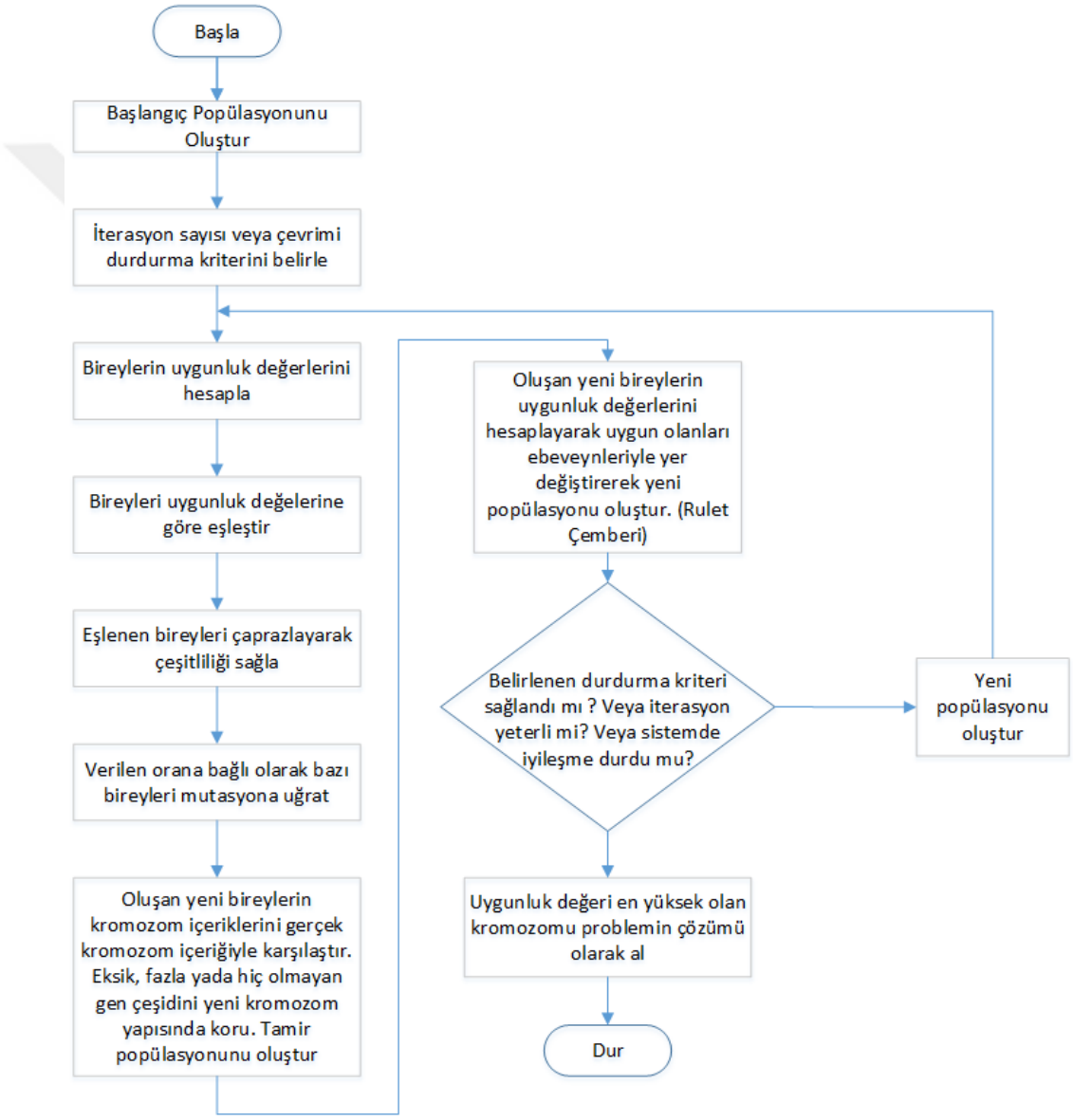
Genetik algoritmalar doğada gözlemlenen evrimsel süreci örnek alan bir arama ve eniyileme yönetimidir. İyi nesillerin kendi yaşamlarını korurken, kötü nesillerin yok olması ilkesine dayanmaktadır (Elmas 2007).

Goldberg (1989)'e göre genetik algoritmalar, rastlantısal arama tekniklerini kullanarak çözüm bulmaya çalışan, parametre kodlama esasına dayanan sezgisel bir arama tekniğidir.

Genetik algoritmalar, geleneksel yöntemlerle çözümü zor yada imkansız olan problemlerin çözümünde kullanılmaktadır. Genetik algoritmalar veri madenciliğinde

kümeleme, tahmin, ilişki kuralları oluşturma ve sınıflandırma gibi alanlarda yaygın olarak kullanılmaktadır (Gürsoy 2009).

Genetik algoritmaların temel işleyişini gösteren akış diyagramı Elmas (2007) tarafından Şekil 3.19'da gösterilmiştir.



Şekil 3.19. Genetik algoritmaların akış diyagramı (Elmas 2007)

3.5. Zamansal Veri Madenciliği

Zaman, verinin en temel yapı taşlarından bir tanesidir. Birçok gerçek yaşam verisi bir takım nesnelerin belirli bir zamandaki durum ve özelliklerini belirtir. Örneğin, bir süpermarket veri tabanında, müşterilerin satın aldıkları ürünler kaydedilir. Bu veriler zaman içerirler. Bu veri tabanındaki her bir kayıt bir zaman damgasına sahiptir. Telekomünikasyon veri tabanında her bir sinyal zamanla ilişkilidir. Üçüncü bir örnek olarak ise hisse senedindeki fiyatlar sabit değildirler zaman ile değişirler (Hsu *et al.* 2008).

Daha önce veri madenciliğini büyük veri yığınlarından anlamlı bilgilerin çıkarılması olarak tanımlamıştık. Zamansal veri madenciliği ise zamansal verileri içeren veri tabanlarından anlamlı bilgilerin çıkarılması işlemi olarak tanımlanabilir.

Zamansal veri madenciliği, zamansal veriler ile gerçekleştirilen veri madenciliğidir. Zamansal veri madenciliğinin birincil hedefi veriler içerisinden ilginç desenlerin keşfedilmesidir (Hsu *et al.* 2008).

Veri tabanları geleneksel olarak zamansal veriler içermezler. Onun yerine zamanda bir noktanın yansımasını saklarlar. Bu nedenle bu veri tabanları anlık veri tabanları olarak adlandırılabilirler. Örneğin bir işçi veri tabanında normal olarak, işyerinde mevcut olan işçilerin bilgileri saklanır, şirket için çalışmış olan işçiler saklanmazlar. Ancak bu anlık veriler ile birçok soru cevaplanamayabilir. Bir şirket yöneticisi, işe alınan ve işten ayrılan işçilerin eğilimleri hakkında bilgi sahibi olmak isteyebilir yada işçilerin etnik çeşitliliklerinin zaman içerisinde nasıl değiştiğini bilmek isteyebilir. Bu tip veri madenciliği zamansal veriler gerektirirler. Zamansal veri tabanında veri birden çok zaman noktasına sahiptir (Dunham 2003). Çizelge 3.21'de zamansal veri tabanının bir örneği gösterilmektedir.

Çizelge 3.21. Zamansal veri tabanı örneği

Tarih	Adı	T.C.	Adres	Maaş
02.12.2002	Elif	123456789	Orta Mah. No:10	5000
08.12.2002	Elif	123456789	Orta Mah. No:10	5200
12.10.2002	Elif	123456789	Yeni Mah. No:13	5200

Çizelge 3.21'de belirtildiği üzere, Elif adlı kişi 02.12.2002 tarihinde 5000 lira maaş ile işe alınmıştır. 08.12.2002 tarihinde ise maaşı 200 lira artarak 5200 olmuştur. 12.10.2002 tarihinde ise adresi değişmiştir. Bir kişi için tek bir kaydın ve zamanın tutulduğu zamansal olmayan veri tabanlarının aksine bu zamansal veri tabanında bir kişi için 3 farklı zamana ait kayıtlar bulunmakta ve bu kayıtlar o kişi hakkında bilgiler vermektedir.

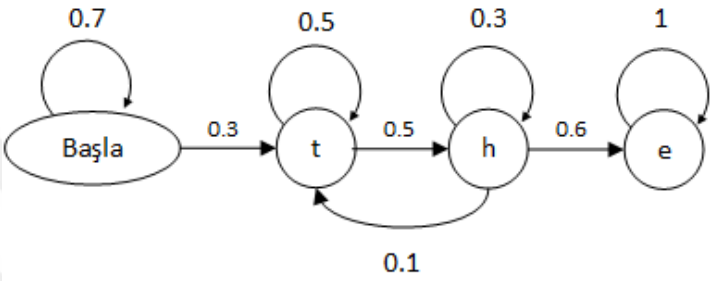
Zamansal veri tabanlarında hisse senedi fiyatları, döviz kuru, biyomedikal ölçüm verileri gibi zaman içeren veriler saklanırlar. Zamansal veriye bir diğer örnek EEG (Elektroensefalografi) verileri gösterilebilir. EEG beynin elektriksel etkinliğinin değerlendirilmesi amacıyla yapılan bir işlemdir. Beyin dokusunda yer alan sinir hücrelerine ilişkin elektriksel sinyaller kafatasındaki saçlı deriye iletilirler. Bu sinyaller kafa derisinin belirli bölgelerine yerleştirilen elektrotlarla bilgisayara aktarılırlar. Bu sinyaller bir kişi için sürekli değiştiğinden böyle bir uygulama zamansal veri madenciliği alanına girmektedir.

3.5.1. Zamansal olayların modellenmesi

Zamansal olayların sırasını modellemek için bir çok farklı teknik bulunmaktadır. Bunlardan en yaygınları Markov modeli, Saklı Markov modeli ve yinelemeli yapay sinir ağlarıdır (Dunham 2003).

3.5.1.a. Markov modeli

Şekil 3.20'de görüldüğü üzere Markov modeli desen keşfinde kullanılabilen yönlendirilmiş grafiklerdir. Grafikteki her bir düğüm bir olay dizisini tanımada, bir durum ile ilişkilidir. Düğümlerden çıkan her bir geçiş ise bir olasılık ile ilişkilidir (Dunham 2003).



Şekil 3.20. Basit markov modeli (Dunham 2003)

Yukarıdaki Markov modelinde başlangıç düğümünden t düğümüne geçiş olasılığı 0.3 dür. Bir düğümden çıkan geçişlerin olasılıkları toplamı 1 olmalıdır. Başlangıç yada bitiş düğümleri gösterilmeyebilir. Markov modelinin en temel özelliği geçiş olasılığının önceki durumlardan bağımsız olmasıdır. Yani Markov modeli hafızasız bir modeldir.

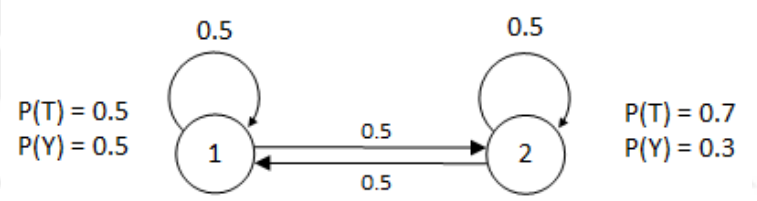
Markov modeli günümüzde konuşma tanıma, doğal dil işleme gibi bir çok uygulamalarda kullanılmaktadır.

3.5.1.b. Saklı markov modeli

Saklı Markov Modeli, Markov modelinin bir uzantısıdır. MM ile HMM (Hidden Markov Model) arasındaki en temel fark HMM'de sistemin hangi durumda olduğunun bilinmemesidir. Ancak elde edilen gözlemler ile hangi durumda olduğu tahmin edilebilir. Bir diğer fark ise ekstra olasılık değerlerinin var olmasıdır. Ekstra olasılık değerlerinin var olması modelde gizli kısımların olduğunu göstermektedir (Dunham 2003).

HMM, MM'de olduğu gibi ses tanıma, doğal dil işleme, el yazısı tanıma, jest tanıma, biyoenformatik gibi alanlarda kullanılmaktadır.

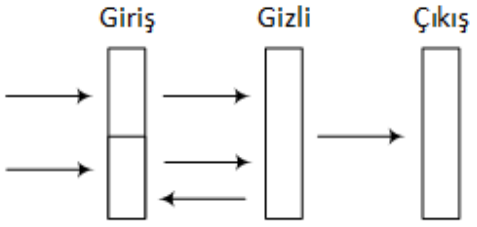
Şekil 3.21'de iki adet bozuk paranın havaya atılma durumları Saklı Markov Modeli ile gösterilmiştir. Bu paralarda bir tanesi hilesiz para olup yazı ve tura gelme olasılığı eşittir. Diğer para ise hileli para olup tura gelme olasılığı 0,7 iken yazı gelme olasılığı ise 0,3 dür. Geçiş olasılıkları ise iki para olduğundan 0,5 dir. Örnekte görüldüğü gibi gizli ihtimaller, durumdan çıkacak sonucu belirlemek için kullanılırken, gizli olmayan değerler ile oluşacak bir sonraki durum belirlenir. Yani birinci paradan sonra ikinci paranın atılması gizli olmayan bir olasılıktır. Ancak paranın yazı yada tura gelme olasılığı gizli bir olasılıktır (Dunham 2003).



Şekil 3.21. Saklı markov modeli (Dunham 2003)

3.5.1.c. Geri dönüşümlü yapay sinir ağları

Şekil 3.22'de görüldüğü gibi geri dönüşümlü yapay sinir ağlarında, ağın çıktıları, belirli bir şekilde ağa girdi olarak geri gönderilir. Bu tür ağlar, geri dönüşümlü yapay sinir ağları, olarak tanımlanır (Öztemel 2012).



Şekil 3.22. Geri dönüşümlü yapay sinir ağı

İleri beslemeli yapay sinir ağları zamansal olayları modellemek için kullanılmaz. Çünkü zaman mekanizmasına sahip değillerdir. Ancak ileri düzey yapay sinir ağları mimarisi, tanımlama ve tahmin problemlerinde kullanılabilir. Gizli katman ya da çıkış katmanındaki düğümlerin çıkışı, daha önceki bir katmana giriş olabilir. Böylece geri dönüşümlü bir yapay sinir ağı oluşmuş olur ve bu yapay sinir ağları zamanı kaydedebilir ve zamansal tahmin uygulamalarında kullanılabilir. Ancak geri dönüşümlü yapay sinir ağlarının kullanımı ve eğitilmesi oldukça zordur. Geleneksel ileri beslemeli yapay sinir ağlarının aksine, geri dönüşümlü yapay sinir ağlarının çıktı üretmesi için gereken süre belirsizdir. Çünkü gizli katman ve çıkış katmanındaki düğümler sistem dengeleninceye kadar durmadan aktif yada pasif olacaklardır (Dunham 2003).

3.6. Mekânsal Veri Madenciliği

Tobler (1979)'ın "her şey birbiriyle alakalıdır, ancak yakın olan şeyler birbirleriyle daha alakalıdır" kanunundan yola çıkarak, mekânsal veri madenciliğine, yakın, uzak, doğusu, batısı gibi ilişkiler kullanarak, mekânsal ve mekânsal olmayan veriler arasından bir takım ilişkiler çıkartma işlemidir denilebilir (Hsu *et al.* 2008).

Mekânsal veri, mekan yada bölge bileşenlerini içeren verilerdir. Mekânsal veri, adres, enlem, boylam gibi veriler olabilir. Mekânsal veriye, mekânsal operatörler olan near, north, south, adjacent, contained in gibi operatörlerle ulaşılabilir. Mekânsal veri, mekânsal yada mekânsal olmayan verileri kaydedebilen veri tabanlarında saklanır. Mesafe bilgisinin mekânsal veri ile ilişkili olmasından dolayı mekânsal veri tabanları genellikle özel veri yapıları kullanılarak saklanılmaktadır (Dunham 2003).

Mekânsal veri, coğrafi bilgi sistemleri gibi birçok güncel bilgi teknolojileri sistemleri için gereklidir. Coğrafi Bilgi Sistemleri (CBS), dünya yüzeyindeki coğrafi bölgelerle alakalı bilgileri saklamak için kullanılır. Coğrafi bilgi sistemlerinin, hava ile ilgili uygulamalar, alt yapı ihtiyaçları, afet yönetimi gibi uygulama alanları vardır (Dunham 2003).

3.6.1. Mekânsal sorgular

Mekânsal verinin karmaşıklığından dolayı, mekânsal sorgu gerçekleştirmek ve optimize etmek için birçok çalışma yapılmıştır. Mekânsal olmayan veriye ulaşmak için kullanılan sorgularda, standart karşılaştırma operatörleri olan $<$, $>$, $\leq$, $\geq$, $=$, $\neq$ kullanılırken, mekânsal veriye ulaşmak için ise near, north, south, adjacent, contained in, overlap ya da intersect gibi sorgu operatörleri kullanılmaktadır. Aşağıdaki iki tane mekânsal sorgu örneği verilmiştir (Dunham 2003).

- Find all houses near Mohawk Elemantary School
- Find the nearest fire station to 9631 Moss Haven Drive in Dallas

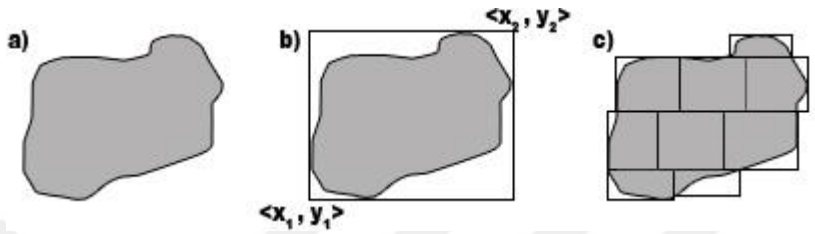
3.6.2. Mekânsal veri yapıları

Mekânsal verilerin benzersiz özelliklerinden ötürü, mekânsal veriyi indeksleme yada kaydetmek için bir çok veri yapısı dizayn edilmiştir.

Mekânsal olmayan veri tabanı sorguları B-tree gibi geleneksel indeksleme yapılarını kullanarak veriye tam erişim sağlarlar. Ancak mekânsal sorgular, mekânsal nesnenin göreceli yerine göre yakınlık ölçüleri kullanabilir. Bu mekânsal sorguları verimli bir şekilde gerçekleştirebilmek için uzayda birbirine yakın olan nesnelerin disk üzerinde kümelenmesi önerilmektedir. Bu amaçla söz konusu olan coğrafi bölge yakınlığa göre hücrelere bölünebilir ve sonra bu hücreler diskteki bloklarla ilişkilendirilebilir. İlgili veri yapısı bu hücrelere göre inşa edilir (Dunham 2003).

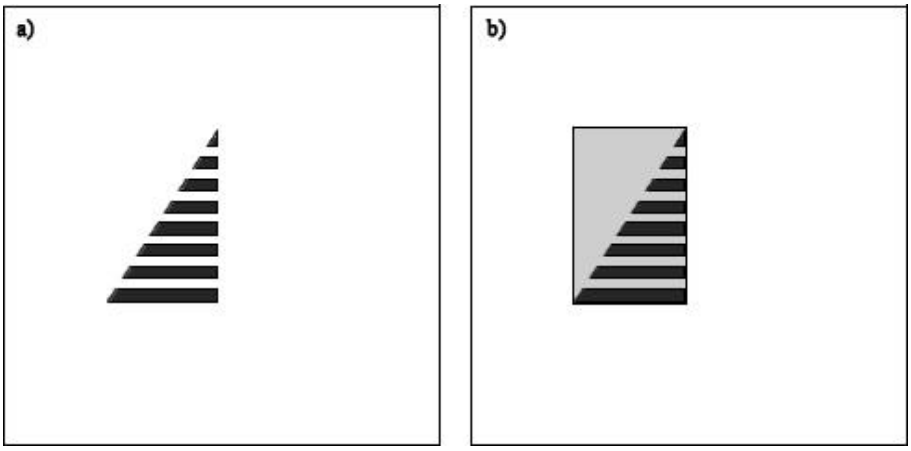
Mekânsal nesneleri modelleyebilmek için kullanılan en yaygın teknik MBR (minimum bounding rectangle) tekniğidir. Şekil 3.23'de MBR tekniğinin kullanımı gösterilmektedir. Şekilde bir gölün sınırları gösterilmektedir. Eğer bu göl geleneksel koordinat sistemiyle modellenmek istenilseydi Şekil 3.23.b'de görüldüğü gibi, gölü kapsayan bir dikdörtgen gerekirdi. Böylelikle MBR'nin bir mekânsal objeyi modellemek için kullanılabileceğini anlaşılabilmektedir. Bu yönteme alternatif olarak,

Şekil 3.23.c'de göl bir dizi küçük dikdörtgenler vasıtasıyla modellenmektedir. Bu yöntem modellenecek nesneye daha yakın ve uyumlu bir modellemedir, ancak birden fazla MBR kullanılmasını gerektirir. Şekil 3.23.b'de gösterildiği gibi bir MBR bitişik olmayan iki köşenin koordinatlarıyla kolaylıkla gösterilebilmektedir (Dunham 2003).



Şekil 3.23. MBR örneği (Dunham 2003)

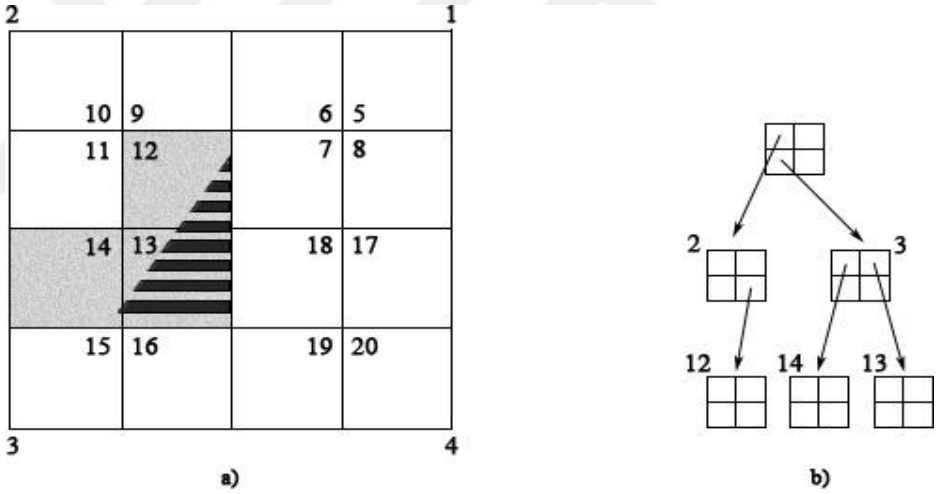
Şekil 3.24.a'da bir üçgen basit bir mekânsal nesne olarak sunulmaktadır. Şekil 3.24.b'de üçgen için bir MBR görülmektedir. Mekânsal veri madenciliği aktivitelerine yardım etmek için mekânsal göstergeler kullanılabilir. Mekânsal veri yapılarının bir yararı, nesneleri bölgelere göre kümelemesidir. Bu göstermektedir ki, n boyutlu uzayda birbirlerine yakın bulunan nesneler, veri yapısı ve disk üzerinde birbirlerine yakın olma eğilimindedirler. Böylece, bu yapılar bir algoritmanın arama alanını sınırlayarak, işlem yükünü azaltmakta kullanılabilir. Ek olarak, mekânsal sorgular, bu veri yapıları kullanılarak etkili bir şekilde gerçekleştirilebilir (Dunham 2003).



Şekil 3.24. Mekansal nesne örneği (Dunham 2003)

3.6.2.a. Dörtlü ağaç (Quad tree)

Mekânsal veriler için önerilen ilk veri yapılarından biri dörtlü ağaçtır. Dörtlü ağaç mekânsal nesneyi hiyerarşik bir ayrıştırma yaparak sunmaktadır. Şekil 3.24'deki mekânsal nesnenin dörtlü ağaç kullanılarak nasıl modellendiği Şekil 3.25'de gösterilmektedir. Şekilde görüldüğü gibi MBR'ler hiyerarşik bir şekilde 4'lü gruplara ayrılmaktadır. Her bir hücre sağ üst köşeden başlayarak saat yönünün tersine, hiyerarşik bir şekilde numaralandırılır. Şekilde görüldüğü gibi önce büyük, daha sonra küçük kareler numaralandırılmaktadır. Mekânsal nesneyi 12, 13 ve 14 nolu hücreler temsil etmektedir. Şekil 3.25.b'de görüldüğü gibi ağaç kurulurken sadece dolu olan hücreler modellenmektedir (Dunham 2003).

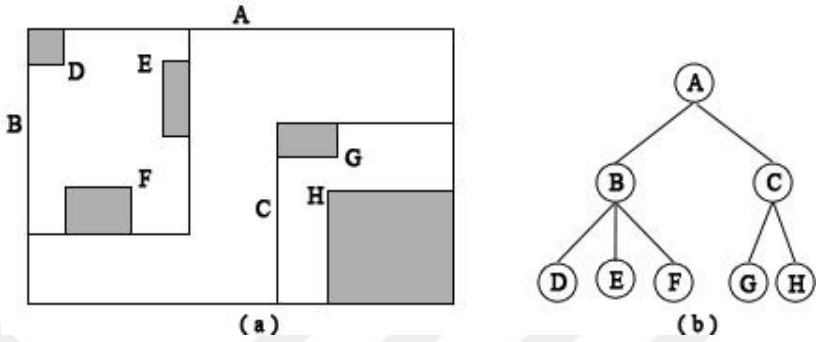


Şekil 3.25. Dörtlü ağaç örneği (Dunham 2003)

3.6.2.b. R-ağaç (R-tree)

R-Ağaç, poligonlar, dikdörtgenler, coğrafi koordinatlar gibi çok boyutlu verileri indekslemek için kullanılan bir veri yapısıdır. Verileri MBR kullanarak modelleyen bir yaklaşımdır. Ağaçtaki her bir katman küçük dikdörtgenlere karşılık gelir. Bir R-ağaçta hücreler üst üste gelebilir. Nesneler bir hücre içerisinde yer alan MBR ile temsil edilir. R-ağacın büyüklüğü kaç adet nesne olduğuyla ilişkilidir. Şekil 3.24'deki üçgen şekli R-ağaç ile gösterildiğinde sadece kök düğümünden oluşan bir ağaç oluşur. Ancak Şekil

3.26'da karışık bir şekil vardır. Bu şekilde 5 adet nesne sunulmuştur. Coğrafi alan kök düğümüdür ve Şekil 3.26.b'de gösterilmiştir (Dunham 2003)

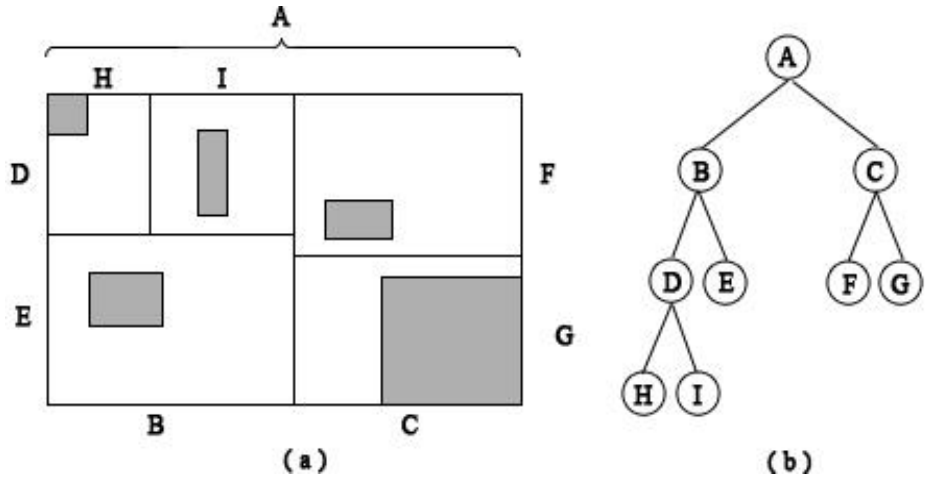


Şekil 3.26. R-ağacı örneği (Dunham 2003)

Mekânsal operatörleri gerçekleştirmek için kullanılan algoritmalarda R-tree kullanılması nispeten daha kolaydır. Verilen nesne ile kesişen tüm nesneleri bulunmak istenilirse, yapılan sorguda sadece ağacın üst seviyelerinde arama yapılır, böylece kesişmeyen alt ağaçlara bakılmaz (Dunham 2003).

3.6.2.c. k-D ağacı (k-D tree)

k-D ağacı sadece mekânsal verileri modellemek için değil, bir çok veriyi modellemek için kullanılabilir. k-D ağacı, ikili arama ağacının farklı bir çeşididir. Ağaçtaki her bir seviye bir özelliği indeksler. Şekil 3.27'de, R-ağacında kullanılan mekânsal verinin aynısı kullanılarak k-D ağacı oluşturulmuştur. Bölünmelerde MBR kullanılmaz. A hücresi tüm bölgeleri kapsar ve ağacın kökünü oluşturur. Mekânsal veri bütün hücrelerde tek bir nesne kalacak şekilde parçalara bölünür. Bu örnekte A bölgesi ilk olarak B ve C bölgelerine bölünmüştür. Daha sonra B bölgesi D ve E, C bölgesi ise G ve F bölgelerine bölünmüştür. Son olarak ise D bölgesi H ve I olarak iki ayrı bölgeye bölünerek, her bir nesne bir hücre içerisine alınarak modellenmiştir (Dunham 2003).



Şekil 3.27. k-D ağacı örneği (Dunham 2003)

3.6.3. Mekânsal sınıflandırma algoritmaları

Mekânsal veri madenciliğinin hedefi, mekânsal veri tabanlarından mekânsal ve mekânsal olmayan özellikleri birleştirerek gizli bilgileri keşfetmektir. Mekânsal veri madenciliği genellikle geleneksel veri madenciliğinde kullanılan algoritmalara, mekânsal özellik kazandırılarak gerçekleştirilir (Fayyad *et al.* 1996).

Mekânsal olmayan veriler için kullanılan sınıflandırma algoritmaları geliştirilerek, mekânsal verilerde kullanılabilir hale getirilmiş ve neticede birçok mekânsal sınıflandırma algoritması oluşturulmuştur.

Mekânsal veri madenciliğinin en belirgin özelliği nesneler arasındaki mekânsal ilişkiyi dikkate almasıdır (Egenhofer and Sharma 1993).

3.6.3.a. ID3 uzantısı

Mekânsal olmayan veriler için kullanılan ID3 algoritmasında bir takım değişiklik yapılarak, algoritmanın mekânsal veriler için kullanılabilir hale getirilmesi neticesinde oluşmuş bir algoritmadır. Algoritma sadece sınıflandırılacak nesnenin özelliklerini dikkate almaz, aynı zamanda o nesnenin komşusu olan nesnelerin özelliklerini de

dikkate alır. Söz konusu mekânsal sınıflandırma kuralları üreten algoritmanın kod taslağı Çizelge 3.22'de gösterilmektedir (Ester *et al.* 1997)

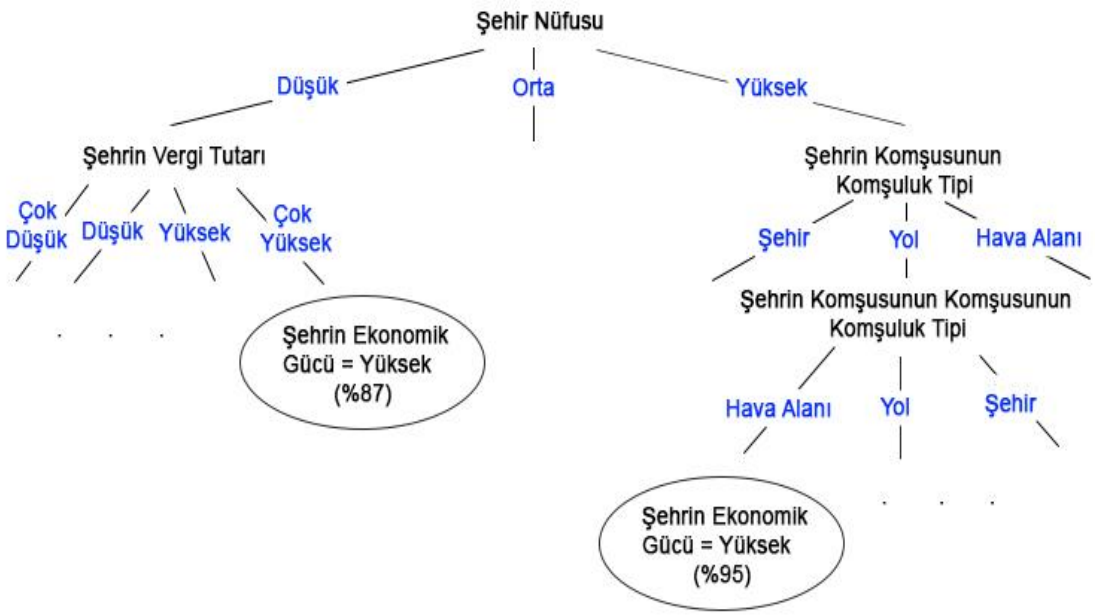
Çizelge 3.22. Mekansal ID3 algoritması (Ester *et al.* 1997)

```
discover_spatial_classification_rules(db:Set_of_Objects;sel:NonSpatialPredicate;
class_attr:Attribute; pred:NeighborhoodRelation; max_length:Int)
    focus:=select(db,sel);
    neighborhood:=get_nGraph(focus,pred);
    paths:=create_nPaths(focus,neighborhood,larger_distance,max_length);
    classify(class_attr,neighborhood,EMPTY_RULE,paths, max_length);
end  // discover_classification_rules;

classify(class_attr:Attribute;neighborhood:NeighborhoodGraph;
rule:ClassificationRule; paths:set_of_paths; max_length:Int)
    max_info_gain:=0.0;
    max_attr:=NULL;
    for i from 1 to max_length do
        for each generalized attribute (Aj,i) not used in rule do
              info-gain:=calculate_information_gain(Aj,class_attr,i,paths);
              if info_gain > max_info_gain then
                    max_attr:=Aj;
                    max_neighbors:=i;
                    max_info_gain:=info_gain;
              end if // info_gain > max_info_gain;
        end for // each attribute Aj not yet used;
    end for // i from 0 to max_length - 1;

    if max_attr ≠ NULL and max_info_gain > ε then
        for each value of max_attr do
            extended_rule:=rule + "max_attr,max_neighbors,value";
            classify(class_attr,neighborhood, extended_rule,paths,max_length);
        end for // each value of max_attr;
    else
        print rule;
    end if // max_attr ≠ NULL and max_info_gain > ε;
end // classify;

calculate_information_gain (attr,class_attr: Attribute; index:Int;paths:set_of_paths);
    for each path in paths do
          consider attr of the index-th object of path and class_attr of the first object of
          path for the calculation of the information gain
    end for // each path in paths;
end // calculate_information_gain;
```

Nesneler kesişme, doğu, batı, yakın, uzak gibi bir takım ilişkileri karşılıyorlarsa komşu nesne olarak nitelendirilirler. Örneğin bir kentin ekonomik gücü komşu şehirlerin

türlerine göre sınıflandırılabilir. Hem sınıflandırılacak şehrin özellikleri (şehrin nüfusu, vergi miktarı vs.) hem de sınıflandırılacak şehrin komşularının özellikleri (komşu şehrin türü, komşu şehrin komşusunun türü vs.) mekânsal sınıflandırma algoritması tarafından dikkate alınır. Algoritma, mekânsal ilişkiyi nitelik olarak ele alır, ancak mekânsal verinin önemli bir özelliği olan, mekânsal öz ilişkiyi analiz etmez (Grossman 2001).

Şekil 3.28'de ID3 uzantısı olan algoritma kullanılarak oluşturulan karar ağacı ve mekânsal sınıflandırma kuralı verilmiştir.



Eğer Şehrin Nüfusu = Düşük **VE** Şehrin Vergi Tutarı = Çok Yüksek
İSE Şehrin Ekonomik Gücü = Yüksek (87 %)

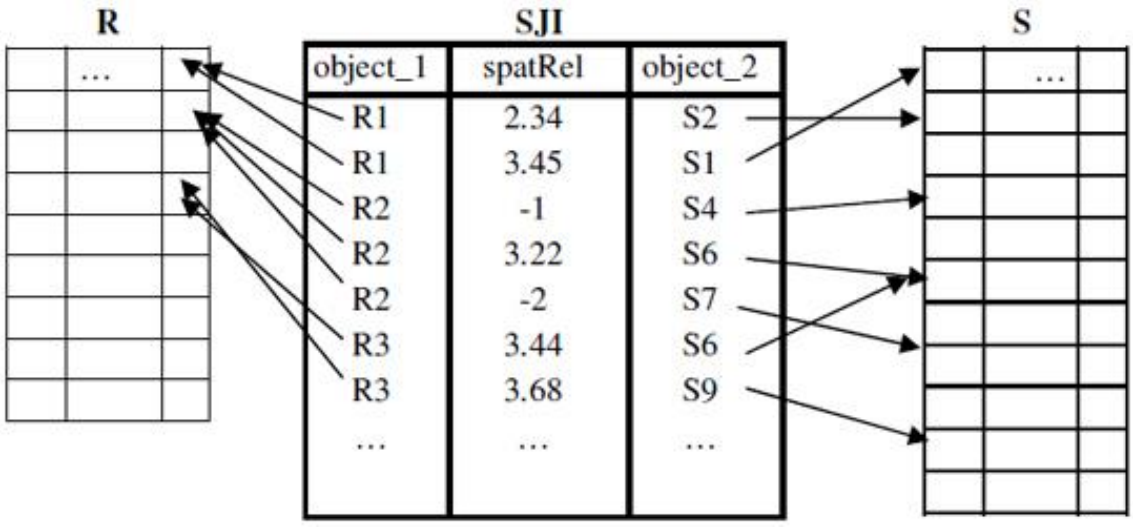
Eğer Şehrin Nüfusu = Yüksek **VE** Şehrin Komşusunun Komşuluk Türü = Yol **VE** Şehrin Komşusunun Komşusunun Komşuluk Türü = Hava Alanı
İSE Şehrin Ekonomik Gücü = Yüksek (95 %)

Şekil 3.28. Mekânsal karar ağacı ve kurallar (Ester *et al.* 1997)

3.6.3.b. SCART algoritması

SCART algoritması iki bölümden oluşmaktadır. Birinci bölüm mekânsal birleştirme indekslerinin kullanılması, ikinci bölüm ise bu metotların ilişkisel veri madenciliğine uygulanmasıdır.

İlişkisel veri modelinin aksine mekânsal ilişkileri hesaplamak daha karmaşıktır. Bu ilişkileri hesaplamak için birçok mekânsal birleştirme operatörü gerekir. Bu gereksinim de beraberinde maliyet getirir. SCART algoritmasında mekânsal ilişkileri hesaplamak için ikincil bir yapı kullanılmaktadır. Bu yapı mekânsal birleştirme indeksi olarak adlandırılır (SJI) ve ilişkisel veri tabanı yapılarında kullanılan join işleminin bir uzantısıdır. Şekil 3.29'da görüldüğü gibi SJI, mekânsal ilişkileri S ve R tematik katmanlardaki eşleşen nesnelere referans gönderen bir ikincil tablodur. İlişki eğer topolojikse spatRel özelliği negatif değer içerir. Aksi halde spatRel özelliğinde uzaklık değerleri saklanır (Chelghoum *et al.* 2002).



Şekil 3.29. Mekânsal birleştirme indeksi (Chelghoum *et al.* 2002)

SCART (Spatial CART) algoritması, CART algoritmasının bir uzantısıdır. Bu nedenle SCART olarak adlandırılmıştır. SCART bilgi kazancı için Twoing metodunu kullanır. CART algoritmasından farkı hedef nesne ile mekânsal ilişkilere sahip olan bir takım

komşu nesnelerin var olma kriterine göre düğümler dallandırılırlar. Algoritma diğer nesnelerle olan komşuluk ilişkilerini tam olarak verebilmektedir. Böylece, bilgi kazancının hesabı, hedef nesneler ile komşu nesnelerin özellikleri ve onların uzaklıklarını veya topolojik ilişkilerini birleştirir.

Çizelge 3.23. SCART algoritması yalancı kodu (Chelghoum *et al.* 2002)

Giriş Parametreleri:
Target_table: analiz nesneleri
Neighbor_table: tematik katman nesneleri
Spatial_join_index: mekânsal birleştirme tablosu indeksi
Target_attribute: tahmin edilecek veri
Predictive_attributes: hedef niteliği tahmin etmekte kullanılacak değerler
Saturation_condition: algoritmayı sonlandırma şartı
Output:
İkili karar ağacı

1.Adım:
Tüm hedef nesneleri köke at

2.Adım:
Best_gain = 0
For each predictive_attribute
For each attribute_value
If predictive_attribute **ait** target_table
Info_gain = bilgi kazancını hesapla // CART ile aynı
Else If predictive_attribute **ait** neighbor_table
For each mekansal ilişki *spatRel*
Info_Gain =Bilgi kazancını ayrıştırma kriterine göre hesapla
If Info_gain > Best_gain
Ayrıştırma kriterini kaydet
3.Adım:
If o anki yaprak sonlandırma noktasına ulaşmadıysa
Düğümü böl
Step 4:
Bir sonraki kod numarasına göre aktif olan düğümü, yeni düğümle değiştir
2. ve 3. adımları yeni düğüm için uygula
//algoritma tüm alt dallar doyuma ulaşıncaya kadar çalışacaktır.

3.7. Zamansal Mekânsal Veri Madenciliği

Konumlandırma teknolojilerindeki ve konum tabanlı servislerdeki gelişmeler, zamansal-mekânsal verilerin hızlı bir şekilde birikmesine neden olmuştur. Hareketli

nesneleri modellemek, indekslemek ve sorgulamak için yeni teknikler geliştirilmiştir. Günümüzde veri tabanı teknolojileri, zamansal mekânsal uygulamaların geliştirilmesinde merkezi bir rol oynamaktadır. Zamansal-mekânsal veri tabanlarından bilgi çıkartma işlemi gün geçtikçe daha önemli hale gelmektedir (Hsu *et al.* 2008).

Zamansal-mekânsal veri madenciliği, zamansal ya da mekânsal veri madenciliğinin üç boyutlu uzaya genelleştirilmesi olarak kabul edilebilir. Zamansal veri ve mekânsal veri, zamansal mekânsal veriye uyarlanabilirken, zamansal-mekânsal veri karmaşık ilişkiler içerdikleri için bu ilişkiler basit olarak zamansal ya da mekânsal boyuta ayrı ayrı bakarak keşfedilemezler. Zamansal veya mekânsal veri madenciliğine nazaran zamansal-mekânsal veri madenciliği daha karmaşıktır. Coğrafi alanın karmaşıklığı, zamansal-mekânsal yapı içerisinde verinin konumlandırılması ve zamansal-mekânsal öz ilişki gibi durumlar, zamansal-mekânsal veri madenciliğini zorlaştırır. Zamansal-mekânsal veri tabanlarında, her bir nesne diğer nesnelerle karmaşık bir etkileşim içerisindedir. Bu etkileşim modellenen ortamın geçmiş, şimdi ve gelecekteki durumlarıyla alakalıdır (Hsu *et al.* 2008).

3.7.1. Zamansal-mekânsal veri madenciliğinin önemi

Zamansal-mekânsal veri tabanlarında veri madenciliği yaparken, zamansal mekânsal verinin çoklu durumunu dikkate almalıdır. Zamansal ve mekânsal veri madenciliği teknikleri birleştirilerek, anlamlı zamansal-mekânsal desenler çıkartılabilir (Hsu *et al.* 2008)

Zamansal-mekânsal veri madenciliğinde yapılmış bir çok çalışma zamansal yada mekânsal veri madenciliği tekniklerinin, zamansal-mekânsal verilere uyarlanmasıyla yapılmıştır. Ancak zamansal-mekânsal veri, zamansal veya mekânsal boyuta ayrı ayrı bakılarak keşfedilemeyecek ilişkiler içerir (Hsu *et al.* 2008).

Hsu *et al.* (2008) zamansal-mekânsal veri madenciliğinin önemini aşağıdaki örnekle açıklamışlardır.

Güneydoğu Asya'daki hava olaylarını ve afetleri gösteren Çizelge 3.24'de bir zamansal-mekânsal veri tabanı örneği verilmiştir. Bu zamansal-mekânsal veri tabanındaki bölgeler Şekil 3.30'daki harita üzerinde gösterilmiştir. Verilerden ve haritadan anlaşılacağı üzere, Güneydoğu Asya'daki değişik bölgelerde gerçekleşen, hava olaylarının ve afetlerin etkileşim ilişkisi bulunmak istenmektedir.

Çizelge 3.24. Zamansal-mekânsal veri tabanı örneği (Hsu *et al.* 2008)

ID	Tarih	Bölge	Olay
xxx	26 Temmuz 1965	R2	Orman Yangını
xxx	28 Temmuz 1965	R1	Sis
xxx	30 Temmuz 1965	R3	Atmosfer Basıncı ↓
xxx	2 Ağustos 1965	R4	Sağanak Yağmur
xxx	25 Şubat 2005	A	Deprem
xxx	26 Şubat 2005	A	Tsunami
xxx	28 Mart 2005	B	Deprem

Şekil 3.30. Zamansal mekânsal veri madenciliği örnek harita (Hsu *et al.* 2008)

Var olan mekânsal veri madenciliği tekniklerini kullanarak aşağıdaki mekânsal ilişki kurallarını bulunulabilir.

- **S1:** Denize yakın bir yerde deprem olursa, tsunami olma olasılığı yüksektir.
- **S2:** Atmosferik basıncı yüksek olan bölgelerde, deprem olma ihtimali daha yüksektir.
- **S3:** Eğer R2 bölgesinin yakınlarında orman yangını oluşmuşsa, R1 bölgesinde sis olma olasılığı yüksektir.
- **S4:** Eğer R3 bölgesindeki atmosfer basıncında bir düşüş olursa, R4 bölgesinde her zaman sağanak yağış gerçekleşir.
- **S5:** Eğer R2 bölgesi yakınlarında sis oluşursa, R3 bölgesinde atmosfer basıncı büyük olasılıkla düşer.

Ancak bu mekânsal kurallar, olaylar içerisindeki zamansal ilişkiler hakkında her hangi bir bilgi sunmamaktadır. Bu olaylar içerisindeki zamansal ilişkileri çıkartmak için zamansal veri madenciliği tekniklerini uygulanıldığında aşağıdaki zamansal ilişki kuralları bulunur.

- **T1:** Depremler, daima yüksek atmosfer basıncından sonra ya da süresince gerçekleşir.
- **T2:** Eğer bir orman yangını varsa sonrasında sis oluşacaktır. Sonra atmosfer basıncı düşer, sonra yağmur yağar.

T1 ve T2'de gösterilen zamansal kurallara bakıldığında, bu kuralların da yetersiz bilgi içerdikleri görülmektedir. Bu nedenle yapılan veri madenciliği neticesinde yeterli bilgi sahibi olunmak isteniliyorsa ve veri tabanı zamansal ve mekânsal bilgiler içeriyorsa, kullanılacak veri madenciliği tekniği sadece zamansal ya da sadece mekânsal olmamalı, hem mekânsal hem de zamansal kuralları üreten zamansal-mekânsal veri madenciliği tekniğini kullanılmalıdır. Zamansal-mekânsal veri madenciliği kurallarında, bölge ve önceki ilişkiler birlikte bulunmalıdır.

- **ST1:** Yüksek atmosfer basıncı devam ederken yada kısa süre sonra yüksek atmosfer basıncı olan yerlerin yakınlarında deprem oranı yüksektir.
- **ST2:** R2 bölgesinin yakınlarında sis oluşmadan önce, R1 bölgesinde daima orman

yangını oluşur. Sonra R3 bölgesinde atmosfer basıncı düşer, sonra R4 bölgesinde yağmur oluşur.

- **ST3:** Mart-Nisan aylarında, Asya'nın güneyindeki bir bölgede orman yangını olursa, ardından o bölgenin güneydoğusundaki komşularında sis ve yağmur oluşur.

Zamansal veri madenciliği sonucunda oluşan kurallar, mekânsal veri madenciliği sonucunda oluşan kurallar ve zamansal-mekânsal veri madenciliği sonucunda oluşan kurallar karşılaştırıldığında, zamansal-mekânsal veri madenciliği sonucunda oluşan kuralların daha fazla bilgi verici kurallar olduğu açıkça görülmektedir. Zamansal-mekânsal kurallar sadece farklı bölgelerdeki değişik olayları göstermezler, aynı zamanda bu bölgelerdeki olayların değişim sırasını da gösterirler. Bu nedenle, zamansal mekânsal kurallar, yapılan işi anlama ve doğru tahmin yapma konusunda karar vericiler için daha faydalı ve kullanışlıdırlar.

3.7.2. CR-Ω+ sınıflandırma algoritması

CR-Ω+ algoritması Kora-3 algoritmasının uzantısı olan Kora-Ω ve CR+ algoritmalarının birleşiminden oluşmaktadır. Algoritmanın yalancı kodu aşağıda verilmiştir (Calderón *et al.* 2010).

Çizelge 3.25. CR-Ω+ sınıflandırma algoritması (Calderón *et al.* 2010)

Öğrenme Aşaması

1. Veri kümesindeki her bir sınıf için aşağıdaki tanımlamaları hesapla;
   - $\theta_1^+(C_i)$ sınıftaki en karakteristik desenin pozitif tanımlaması
   - $\theta_1^-(C_i)$ sınıfın negatif tanımlaması
2. Tüm sınıfları için ortak nötr özelliklerinin endüktif tanımını hesapla θ^0

Yeniden Öğrenme Aşaması

3. Veri kümesindeki her bir sınıf için aşağıdaki tanımlamaları hesapla;
   - $\theta_2^+(C_i)$ sınıftaki daha az karakteristik desenlerin pozitif tanımlaması

Sınıflandırma Aşaması

1. Sınıflandırılacak her bir desen için, pozitif karakteristiği, negatif karakteristiği ve tamamlayıcı özellikleri kaydet

Çizelge 3.25. CR-Ω+ sınıflandırma algoritması (Devam)

2. Tanımlanmış her bir sınıfa, sınıflandırılmış desenlerin üyeliklerini gösteren aşağıdaki çözüm kuralını uygula.

$$S(C_i, P) = \left|F_p \cap \theta_1^+(C_i)\right| - \left|F_p \cap \theta_1^-(C_i)\right| + \frac{1}{2}\left|F_p \cap \theta_2^+(C_i)\right|$$

Calderon *et al.* (2010) CR-Ω+ sınıflandırma algoritmasını, Çizelge 3.26'daki eğitim kümesini kullanarak aşağıdaki örnekle açıklamışlardır.

Çizelge 3.26. CR-Ω+ sınıflandırma algoritması için eğitim kümesi (Calderón *et al.* 2010)

Saat ID	Ay	Bölge	C1	C2	C3
8	Temmuz	Santiago Tepalcapa	1	0	0
8	Mayıs	Santiago Tepalcapa	1	0	0
8	Mayıs	Santiago Tepalcapa	1	0	0
1	Ocak	Seccion Parques	1	0	0
2	Ocak	Seccion Parques	1	0	0
4	Nisan	Bosques del Lago	0	1	0
6	Nisan	Bosques del Lago	0	1	0
8	Mayıs	Centro Urbano	0	1	0
5	Temmuz	Ensuenos	0	1	0
5	Temmuz	Santiago Tepalcapa	0	1	0
6	Temmuz	Centro Urbano	0	0	1
6	Haziran	Centro Urbano	0	0	1
6	Mart	Centro Urbano	0	0	1
5	Şubat	Cumbria	0	0	1
4	Haziran	Cumbria	0	0	1

Çizelge 3.26'daki eğitim kümesinde bulunmayan P1 = [8, Şubat, Santiogo Tepalcapa] değerinin hangi sınıfa ait olduğu, CR-Ω+ sınıflandırma algoritmasıyla aşağıdaki şekilde tahmin edilebilir.

1.Adım

Eşik değerleri belirlenir. $\beta_1^+ = 3,\ \beta_1'^+ = 1,\ \beta_1^- = 3,\ \beta_1'^- = 1$

2.Adım

Her bir özellik için bir alt küme oluşturulur. Alt kümeler 1 elemanlı olamaz 2 veya daha fazla elemanlı olmalıdır. Her bir alt kümeye bakılır, o özellik seti o sınıf için en az 3 (β_1^+) defa görülüyor ve diğer sınıflar içinde ise en fazla 1 ($\beta_1'^+$) defa görülüyorsa o özellik seti o sınıf için pozitif karakteristik olur. Bunun tam tersi de negatif karakteristiktir. Her bir sınıf için pozitif ve negatif karakteristikler bulunur.

$\theta_1^+(C_1) = \{[8, \text{Santiago Tepalcapa}]\}$, $\theta_1^-(C_1) = \{[6, \text{Centro Urbano}]\}$

$\theta_1^+(C_2) = \{\}$, $\theta_1^-(C_2) = \{[8, \text{Santiago Tepalcapa}], [6, \text{Centro Urbano}]\}$

$\theta_1^+(C_3) = \{[6, \text{Centro Urbano}]\}$, $\theta_1^-(C_3) = \{[8, \text{Santiago Tepalcapa}]\}$

3.Adım

Tamamlayıcı özellikler bulunur. Yeni eşik değerleri $\beta_2^+ = 2,\ \beta_2'^+ = 1$ olarak tanımlanır. Yeni eşik değerine göre alt kümeler tekrardan bulunur.

$\theta_2^+(C_1) = \{[8, \text{Mayıs}, \text{Santiago Tepalcapa}],\ [8, \text{Mayıs}],$
$[\text{Mayıs}, \text{Santiago Tepalcapa}],\ [8, \text{Santiago Tepalcapa}],$
$[\text{Ocak}, \text{Seccion Parques}]\}$

$\theta_2^+(C_2) = \{[\text{Nisan}, \text{Bosques del Lago}],\ [5, \text{Temmuz}]\}$

$\theta_2^+(C_3) = \{[6, \text{Centro Urbano}]\}$

4.Adım

Sınıfı tahmin edilecek P1 = [8, Şubat, Santiogo Tepalcapa] kümesinin alt kümeleri ile 2. ve 3. adımlarda bulunan kümelerin kesişim sayıları bulunarak aşağıdaki tablo oluşturulur.

F_{P1} ∩	$\theta_{\mathbf{1}}^{+}(C_1)$	$\theta_{\mathbf{1}}^{-}(C_1)$	$\theta_{\mathbf{2}}^{+}(C_1)$	$\theta_{\mathbf{1}}^{+}(C_2)$	$\theta_{\mathbf{1}}^{-}(C_2)$	$\theta_{\mathbf{2}}^{+}(C_2)$	$\theta_{\mathbf{1}}^{+}(C_3)$	$\theta_{\mathbf{1}}^{-}(C_3)$	$\theta_{\mathbf{2}}^{+}(C_3)$
||=	1	0	1	0	1	0	0	1	0

5.Adım

Her bir sınıf için aşağıdaki formül uygulanır.

$$S(C_i, P) = \left|F_p \cap \theta_1^{+}(C_i)\right| - \left|F_p \cap \theta_1^{-}(C_i)\right| + \frac{1}{2}\left|F_p \cap \theta_2^{+}(C_i)\right|$$

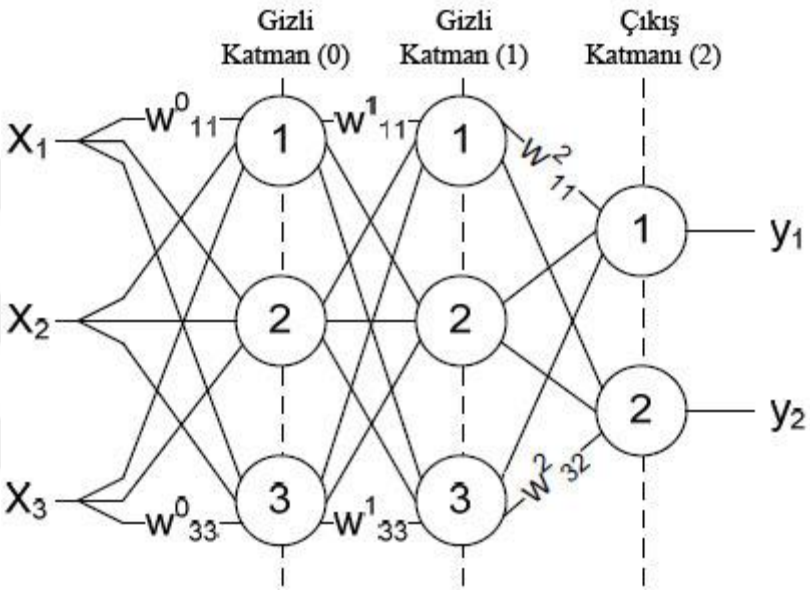
C_i	$\theta_1^{+}(C_i)$	$\theta_1^{-}(C_i)$	$\theta_2^{+}(C_i)$	İşlem	$S(C_i, P_1)$
C_1	1	0	1	$1 - 0 + 0{,}5 * 1$	1,5
C_2	0	1	0	$0 - 1 + 0{,}5 * 0$	-1
C_3	0	1	0	$0 - 1 + 0{,}5 * 0$	-1

Sonuç olarak karakteristiği en yüksek olan sınıf, tahmin edilecek verinin sınıfı olarak seçilir. [8, Şubat, Santiogo Tepalcapa] = C_1

3.7.3. Complex-BP uzantısı

Complex-BP uzantısı, karmaşık değerli geri yayılım algoritması ile eğitilmiş, karmaşık değerli yapay sinir ağlarını esas alan bir zamansal-mekânsal sınıflandırma yöntemidir. Zamansal-mekânsal verinin sınıflandırılmasında kullanılan bir yöntemdir. Robotik, bilgisayar görüşü ve medikal veri analizi gibi alanlarda kullanılabilir.

Bu sınıflandırma algoritması Şekil 3.31'de gösterilen çok katmanlı bir topolojiye sahip olan yapay sinir ağlarını kullanır. Yapay sinir ağına giren her bir giriş, çıkış, ağırlık ve eşik değerleri karmaşık sayılar ile ifade edilir. Girilen verinin zaman boyutu karmaşık sayının faz kısmına, mekân kısmı ise karmaşık sayının genlik kısmına dönüştürülerek kullanılır (Zahradník and Skrbek 2013).



Şekil 3.31. Çok katmanlı bir yapay sinir ağı (Zahradník and Skrbek 2013)

Algoritma karmaşık sayılarla işlem yaptığı için giriş verilerinin dönüştürülmesi, çıkış verilerinin dönüştürülmesi, ağırlıkların değiştirilmesi, eşik değerinin değiştirilmesi, aktivasyon fonksiyonu gibi çok katmanlı bir yapay sinir ağında yapılması gereken işlemler, bu algoritma için karmaşık sayılarla yapılabilir hale getirilmelidir.

Bu algoritma yapay sinir ağının aktivasyon fonksiyonunu iki farklı şekilde kullanır. Birincisinde aktivasyon fonksiyonuna gelen karmaşık sayı, sanal kısım ve gerçek kısım olarak ayrılır. Elde edilen iki reel sayı aktivasyon fonksiyonuna girilir ve aktivasyon fonksiyonu neticesinde bir karmaşık sayı elde edilir. İkinci uygulamada ise aktivasyon fonksiyonuna gelen karmaşık sayı, gen ve faz olarak iki reel sayıya dönüştürülür. Elde edilen iki sayı aktivasyon fonksiyonuna girilir ve neticesinde bir karmaşık sayı elde edilir (Zahradník and Skrbek 2013).

Sonuç olarak bu algoritma zamansal-mekânsal verinin zaman kısmını karmaşık sayının faz kısmı ile, mekan kısmını ise karmaşık sayının genlik kısmı ile temsil edip, karmaşık değerli yapay sinir ağları vasıtasıyla sınıflandırma geçekleştirdiği için öğretmeli bir zamansal-mekânsal sınıflandırma algoritmasıdır.

3.7.4. İlişkisel olasılık ağaçları

Zamansal-mekânsal ilişkisel olasılık ağaçlarına geçmeden önce ilişkisel olasılık ağaçlarını bilmekte fayda vardır. Bu nedenle bu bölümde ilişkisel olasılık ağaçlarından bahsedilmiştir.

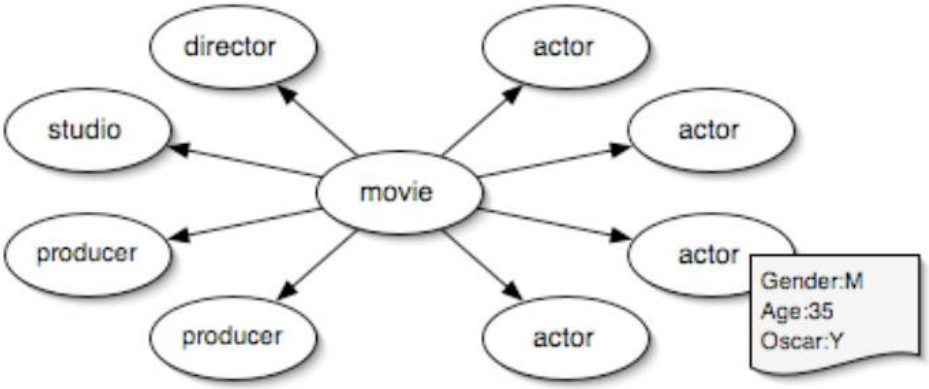
İlişkisel olasılık ağaçları makine öğrenmesi ve veri madenciliği alanında çokça kullanılmaktadır. Ağaç modelleri, veriyi sezgisel ve seçici olarak sunmasından dolayı genellikle kolay yorumlanmaktadırlar. Bu durum ağaç modellerinin bilgi keşfi alanında ilgi çeken bir model olmasını sağlamaktadır. Geleneksel ağaç öğrenme algoritmaları, homojen ve istatistiksel olarak bağımsız olan veri kümeleri için tasarlanmıştır. Ancak ilişkisel olasılık ağaçlarıyla homojen olmayan ve birbirine bağlı ilişkisel veriler üzerinde sınıflandırma yapılabilir (Neville *et al.* 2003).

İlişkisel olasılık ağaçları, olası özellik değerleri üzerindeki olasılık dağılımlarını tahmin ederler. Bu işlem geleneksel önermeli öğrenme metodundan biraz farklıdır. İlişkisel öğrenme algoritmaları, ilişkili nesnelerin etkilerini dikkate almak için tanımlanmış niteliklerin genellikle ötesine bakarlar. Örneğin; bir filmin gişe başarısını tahmin etmek için, ilişkisel model sadece filmin özelliklerini dikkate almaz aynı zamanda aktörlerin, yönetmenin, yapımcının ve stüdyo özelliklerini de dikkate alır. Hatta model daha ileriye gidip, yönetmen tarafından yapılan diğer filmler gibi özellikleri dikkate alır (Neville *et al.* 2003).

İlişkisel veri, geleneksel sınıflandırma tekniklerinin iki varsayımını ihlal etmektedir. Birincisi, önermeli veriler için tasarlanmış algoritmalar veriyi bağımsız ve özdeş varsayarlar. İkincisi, önermeli veriler için tasarlanmış algoritmalar veri örneklerinin

homojen bir yapıda kaydedildiğini varsayar, fakat ilişkisel veri örnekleri genellikle çok çeşitli ve karmaşıktır. Örnek verilecek olursa, bazı filmlerde 10 adet aktör varken bazılarında 1000 aktör olabilir. Bir ilişkisel sınıflandırma tekniği, hem öğrenme hem de sonuç çıkartma için homojen olmayan bağlı veri örnekleriyle uğraşmak zorundadır (Neville *et al.* 2003).

İlişkisel olasılık ağacı üretmek için verinin ağaç üretmek için hazır olması gerekmektedir. Bu işlem ya ön hazırlık aşamasında yada dinamik olarak öğrenme aşamasında uygulanır. Bu işlem genellikle ilişkisel veriyi modelleme için önermeselleştirme anlamına gelmektedir. Heterojen veri örnekleri birçok değerin tek bir değere birleştirilmesiyle homojen hale gelir. Bu aşamadan sonra geleneksel makine öğrenmesi algoritmaları, dönüştürülmüş veriye uygulanabilir. Şekil 3.32'de makine öğrenme algoritmalarının kullanabileceği veriye bir örnek verilmiştir (Neville *et al.* 2003).



(a)

Receipts >$2mil	Mode Actor Gender	Avg Actor Age	Mode Actor Oscar	Mode Studio Loc	...
+	F	28.50	N	USA	...
+	M	33.34	Y	USA	...
-	F	51.00	N	USA	...
+	M	22.79	N	Canada	...
...	...	...	...	...	...

(b)

Şekil 3.32. (a) ilişkisel veri örneği, (b) önermeselleştirilmiş ilişkisel veri kümesi (Neville *et al.* 2003)

3.7.5. Zamansal-mekansal ilişkisel olasılık ağaçları

Gerçek dünya, insanlar, yerler, eşyalar ve olaylar gibi nesneler arasındaki ilişkilerden meydana gelmektedir. İstatistiksel ilişkili öğrenme, endüktif lojik programlama ve ilişkisel bilgi keşfi metotları birçok gerçek dünya uygulamasında kanıtlanmış başarılara sahiptir. Ancak bu yaklaşımlar ya verinin zamansal yönünü dikkate almazlar ya da belirli bir yaklaşımı veri kümesine uydururlar. Diğer bir çok yaklaşım verinin sadece zamansal yönünü modellerler. Zamansal-mekânsal ilişkisel olasılık ağaçları ise hem zamansal yönü hem de mekânsal yönü modellerler (Mcgovern *et al.* 2008)

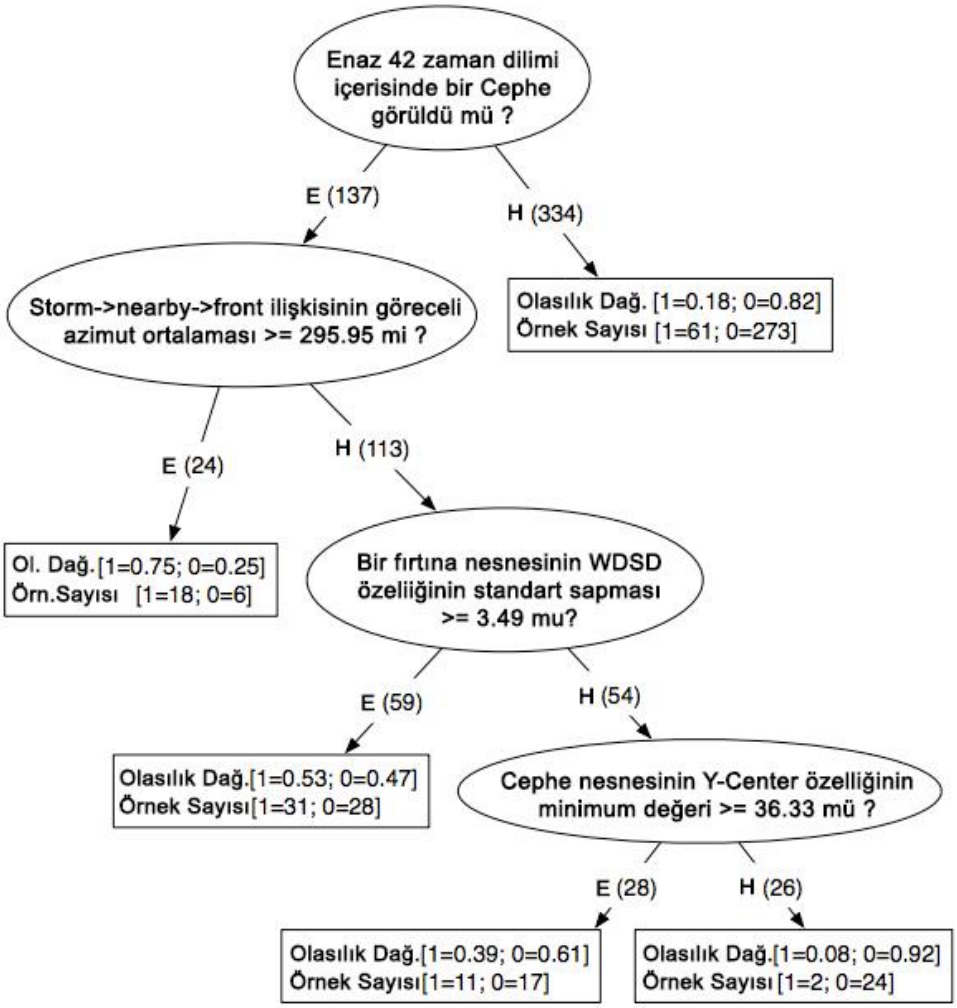
Zamansal-mekânsal ilişkili olasılık ağaçları zamansal-mekânsal ilişkili verileri öğrenebilen olasılık tahmin ağaçlarıdır. Zamansal-mekânsal ilişkili olasılık ağaçları "A Nesnesi, B nesnesinin etrafını sarıyor mu ?", "A nesnesi B nesnesinden önce var mıydı ?", "A nesnesi B nesnesine doğru yaklaşıyor mu?" gibi zamansal, mekânsal yada zamansal-mekânsal sorulara cevap verir (Troutman 2010).

Zamansal-mekânsal olasılık tahmin ağaçları ile diğer karar ağaçları, özellikle ilişkisel olasılık ağaçları arasındaki en önemli fark, zamansal-mekânsal olasılık tahmin ağaçlarının Çizelge 3.27'deki zamansal-mekânsal dallanma tiplerini kullanabilmesidir (Mcgovern *et al.* 2008).

Çizelge 3.27. Ağaç oluşturulurken seçilebilecek SRPT dallanma tipleri

	Dallanma	Açıklama
1	Exists (Bulunma)	Nesne yada ilişki grafikte bulunuyor mu?
2	Temporal Exist (Zamansal bulunma)	En az t adımda nesne yada ilişki bulunuyor mu?
3	Attribute Value (Öznitelik değeri)	Obje yada ilişkinin var olduğunu varsayarak, bu özniteliğin değeri x değerinden büyük yada eşit mi?
4	Temporal Gradient (Zamansal eğilim)	Bir nesnenin veya t ilişkisinin zamana göre kısmi türevli öznitelik değeri v değerinden büyük yada eşit mi?
5	Count (Miktar)	b ana dallanmasının eşleşen örnek sayısı en az v kadar mı?
6	Structural(Yapı)	b ana dallanması bir nesneyle ilişkili mi ?
7	Temporal Ordering (Zamansal Sıralama)	A dallanmasında eşleşen örnekler, b dallanmasındakilerle zamansal olarak ilişkili mi? Zamansal sıralama türleri: önce, kavuşmak, üst üste gelme, eşittir, başlar, biter, boyunca

Şekil 3.33'de bir zamansal-mekânsal olasılık tahmin ağacı verilmiştir. Bu olasılık ağacında oval düğümler dallama düğümleridir, köşeli düğümler ise sınıflandırma için olasılık dağılım fonksiyonunu verir. Storm→nearby→front ilişkisi bir fırtına nesnesinin, bir cephe nesnesiyle yakınında (nearby) ilişkisi ile bağlı olduğunu gösterir. Kenarlardaki etiketler, kaç örneğin dallanırken ayrıldığını gösteren evet yada hayır dallanmalarıdır. Bir yaprak düğümü iki tane bilgi içerir. Birincisi olasılık dağılımı, ikincisi ise o yapraklardaki her bir sınıfın örnek sayısını gösterir.



Şekil 3.33. Bir gök gürültülü fırtınanın kasırgasal olup olmadığını sınıflandıran, cephe veri kümesiyle oluşturulmuş zamansal-mekânsal olasılık ağacına bir örnek (Troutman 2010)

Çizelge 3.28'de zamansal-mekânsal ilişkisel olasılık ağacı algoritmasının iki ana fonksiyonu gösterilmektedir. GrowSRPT fonksiyonu özyinelemeli olarak çalışmakta ve kullanıcının girdiği α (p-değeri eşiği), K (örnek sayısı), D (maksimum ağaç derinliği) ve S (dallanma dizisi) değişkenleri oluşturulacak ağacın büyüklüğünü belirlemektedir. Zamansal-mekânsal ilişkisel olasılık ağacı yapabilmek için, eğitim verisi için önceden etiketlenmiş bir diyagrama ihtiyacımız vardır. Ağaç oluşturulduğu sırada, grafik dizisi aşağı doğru ilerlerken her dallanmada özyinelemeli olarak ayrışır. Öğrenme algoritmasının esas işi bu dallanmaları seçmektir. Tüm olası dallanma dizisinden dallanmalar davranışsal olarak örneklenir ve en iyi dallanma seçilir. FindBestSplit fonksiyonu en iyi dallanmayı seçerek ağacı oluşturur (Troutman 2010).

Çizelge 3.28. Zamansal-mekânsal ilişkisel olasılık ağacı algoritmasının kod taslağı

```
GrowSRPT  Fonksiyonu
Giriş
    K = örnek sayısı, D = ağacın maksimum derinliği, α  = p-değeri eşiği, G = eğitim
      verisi, S = dallanma dizisi, d = o anki ağaç derinliği
Çıkış
   Zamansal-mekânsal ilişkili olasılık ağacı

if d ≤ D then
   dugum = FindBestSplit(G,K, α)  // En iyi dallanmayı bulur
   if dugum <> Ø then
      evetAltdizi, hayırAltdizi = VeriyiBol(dugum, G)
      dugum.ekleEvetDalı(GrowSRPT(K,D, α,evetAltdizi, d+1))  // özyineli çağrı
      dugum.ekleHayırDalı(GrowSRPT(K,D, α,evetAltdizi, d+1)) // özyineli çağrı
     Return dugum
   End
End

Retun new LeafNode(G) // yaprak düğümü oluştur
```

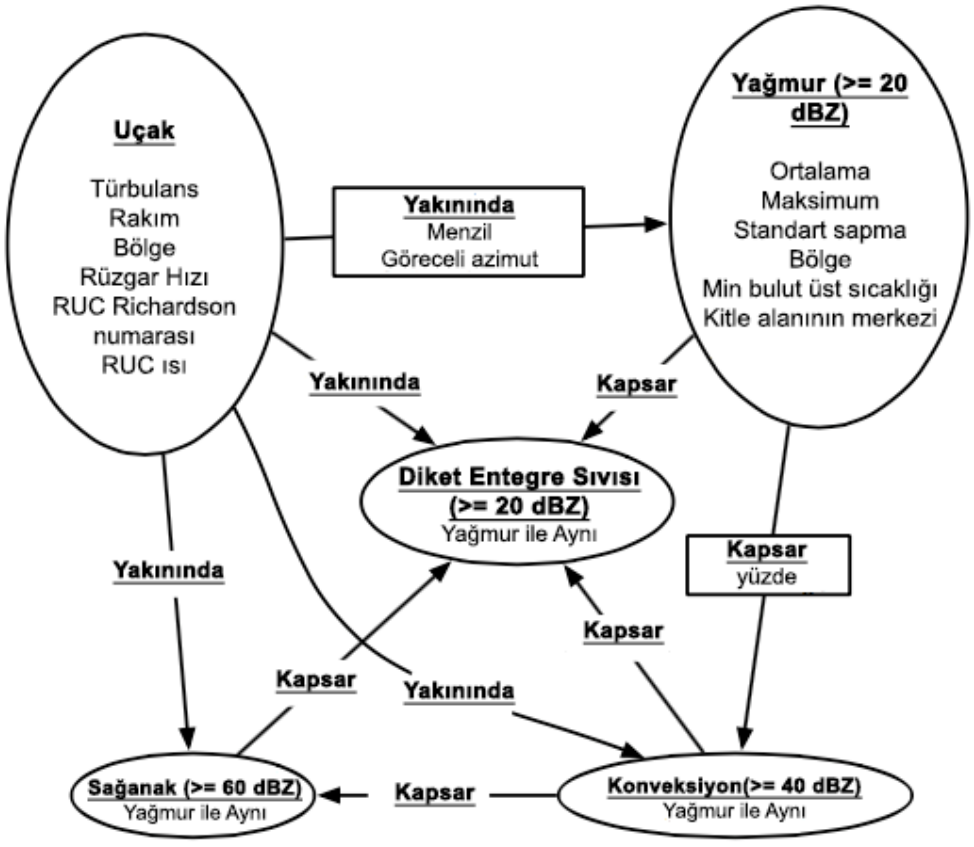
```
FindBestSplit Fonksiyonu
Giriş
   G = eğitim verisi, K = örnek sayısı, α  = p-değeri eşiği, S = dallanma dizisi
Çıkış
   İç düğüm yada Ø

eniyi = Ø
for i = 1 to K do
   dallanma = new InternalNode(RandomDistiction(S))
   fit, p = Calcx²(dallanma,G)          // ki-kare testi
   if p ≤ α and fit > eniyi.fit then eniyi = dallanma
end
```

3.7.6. Zamansal-mekânsal rastgele orman algoritması

Rastgele orman algoritması Breiman tarafından oluşturulmuştur. Bireysel ağaçlar oluşturmak için C4.5 karar ağaçlarını kullanır. Farklı eğitim kümeleri oluşturulurken önyükleme ve rastgele özellik seçimi kullanılır. Her seviyedeki özniteliği belirlerler, önce bütün ağaçlarda hesaplamalar yapılarak nitelik belirlenir, ardından bütün ağaçlardaki nitelikler birleştirilerek en fazla kullanılan öznitelik seçilir. Seçilen nitelik

ağaca dahil edilerek diğer seviyelerde aynı işlemler tekrarlanır (Troutman 2010).

Zamansal-mekânsal ilişkili rastgele orman algoritmasında, zamansal-mekânsal ilişkili olasılık ağaçları kullanılır. Zamansal-mekânsal ilişkili rastgele orman için kullanılacak eğitim verileri Şekil 3.34'de gösterildiği gibi zamansal-mekânsal ilişkili grafiklerdir. Şekilde görüldüğü gibi nesneler elips şeklinde, ilişkiler ise çizgi ve dikdörtgen olarak sunulmaktadır. Hem nesnelerin hem de ilişkilerin kendileriyle alakalı özellikleri bulunmaktadır (Timothy *et al.* 2013).



Şekil 3.34. Zamansal-mekânsal ilişkili rastgele orman algoritması için veri örneği (Timothy *et al.* 2013)

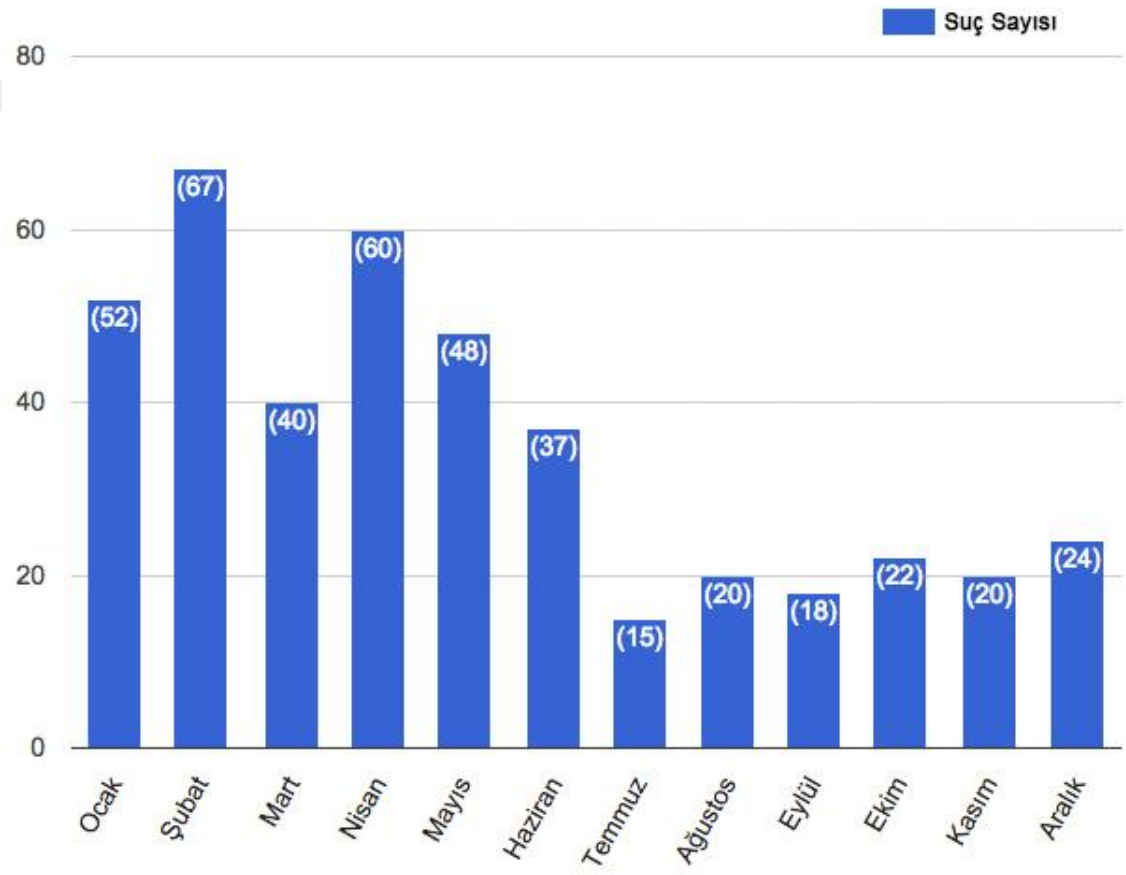
4. BULGULAR ve TARTIŞMA

Bu çalışmada suç tahmini ve analizini gerçekleştirmek için PHP programlama dili vasıtasıyla web tabanlı bir uygulama gerçekleştirilmiştir. Uygulamanın web tabanlı olması uygulamaya erişilebilirlik özelliği katmaktadır. Yani interneti olan herhangi bir cihaz ile uygulamaya erişilebilir. Uygulama web tabanlı olduğu gibi ayrıca mobil uyumludur. Uygulamanın mobil uyumlu olması farklı cihazlarda kullanım kolaylığı sağlamaktadır. Ayrıca sınıflandırma işlemindeki formun, Şekil 4.1'deki gibi sihirbaz formu şeklinde adım adım olması formun herkesin anlayabileceği bir form olmasını sağlamıştır.

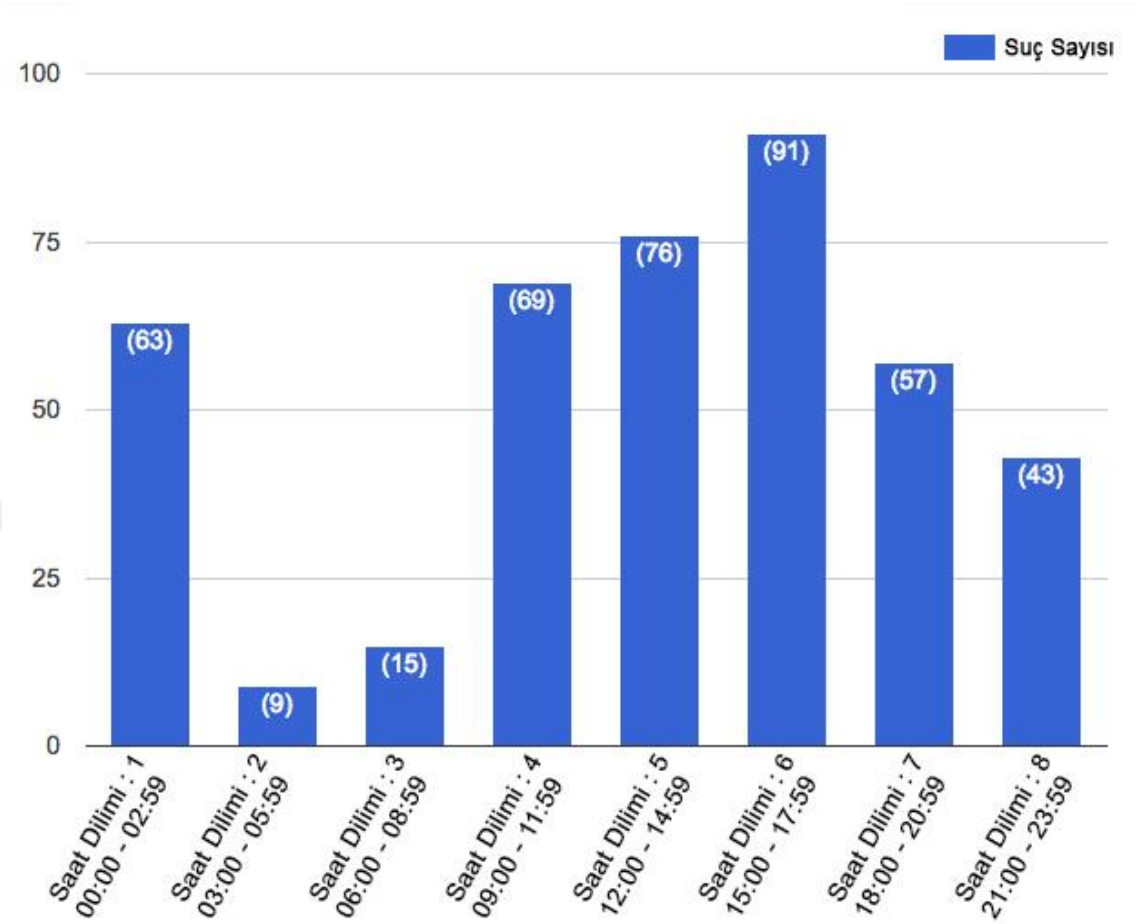
Şekil 4.1. Sınıflandırma formu sihirbaz formu şeklindedir

Uygulamaya veri madenciliği yapabilmek için istenilen sayıda veri kümesi eklenebilmektedir. Eklenilen veri kümelerine CR-Ω+ uygulanarak suç tahmini işlemi gerçekleştirilebilir. Uygulamada veri giriş işlemi tek veya Excel belgesi ile çoklu bir şekilde yapılabilir. Eklenilen her bir veri kümesi üzerinde düzenleme, silme işlemleri gerçekleştirebilmektedir. Uygulamadaki yardım menüsünde bulunan videolarda tüm yapılan işlemler uygulamalı olarak anlatılmıştır. Yapılan bu uygulamada ayrıca suç verilerinin analizleri gerçekleştirebilmektedir. Seçilen veri kümesinin, seçilen alanları pasta yada çubuk grafik olarak son kullanıcıya sunularak suç verilerinin analiz işlemi

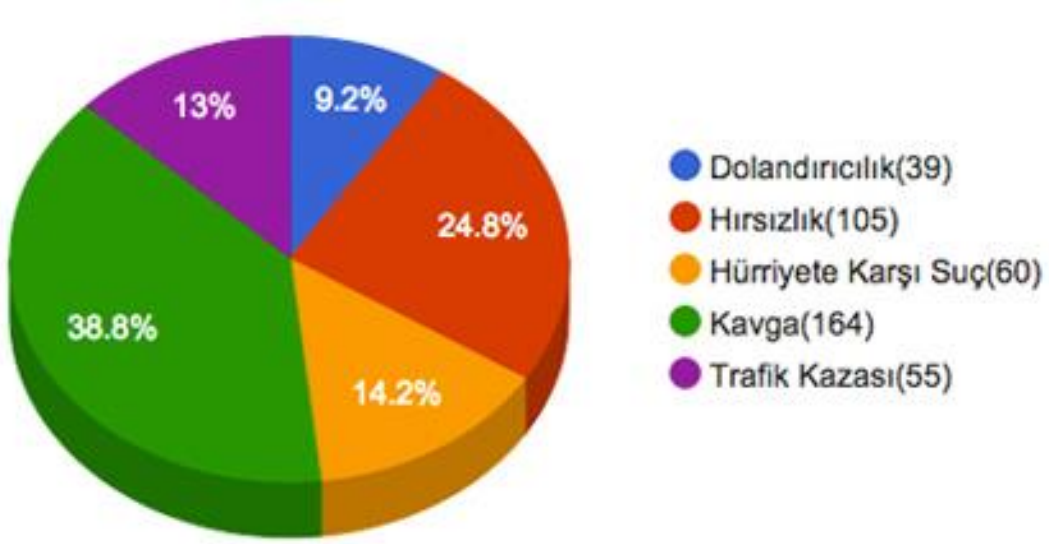
gerçekleştirilebilir. Şekil 4.2, 4.3, 4.4 ve 4.5'deki grafiklerde veri analiz çalışması sonucunda çıktı olarak alınan grafikler verilmiştir. Veri analiz çalışmasında seçilen herhangi bir yada birden fazla alanın veri analizi, grafikler vasıtasıyla gerçekleştirilebilir. Veri analizi işlemi neticesinde suç verisinin nasıl bir veri olduğu anlaşılabilir. Sistemde kayıtlı herhangi bir veri kümesi için analiz işlemi gerçekleştirilebilmektedir.



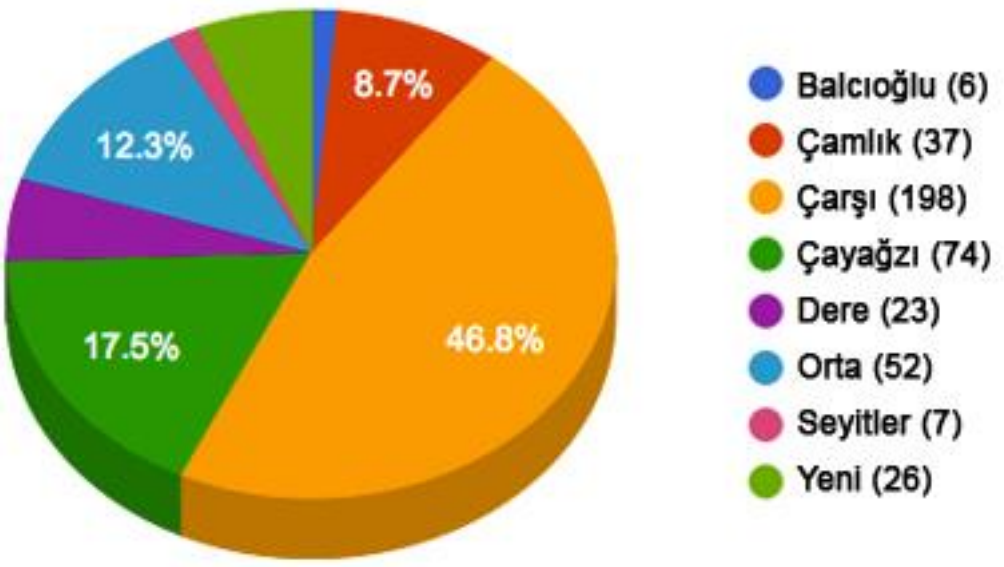
Şekil 4.2. Suç verilerinin aylara göre dağılımı



Şekil 4.3. Suç verilerinin saat dilimine göre dağılımı

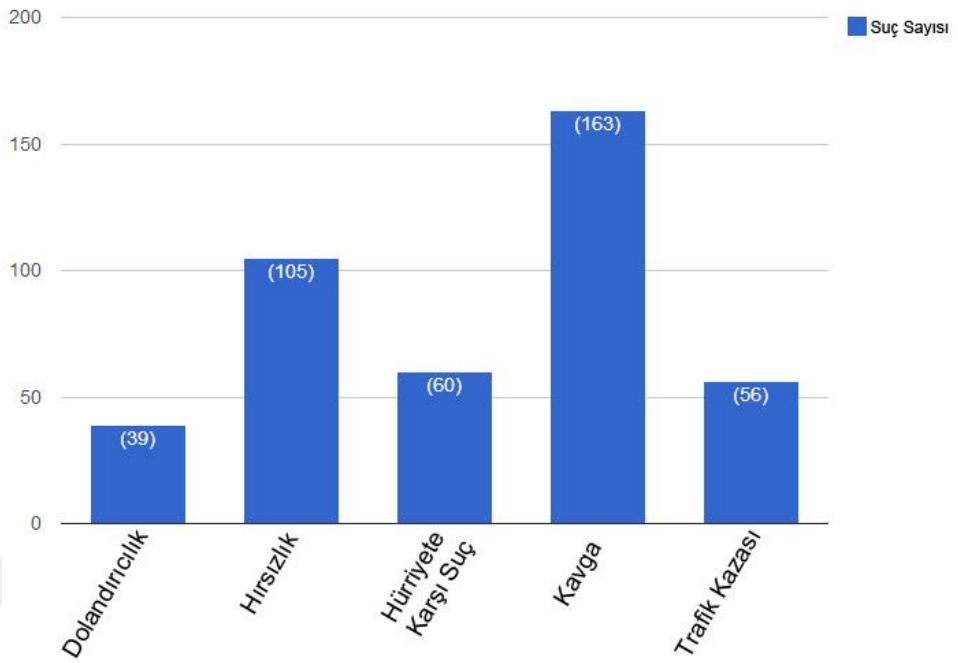


Şekil 4.4. Suç verilerinin tür alanına göre pasta grafiği

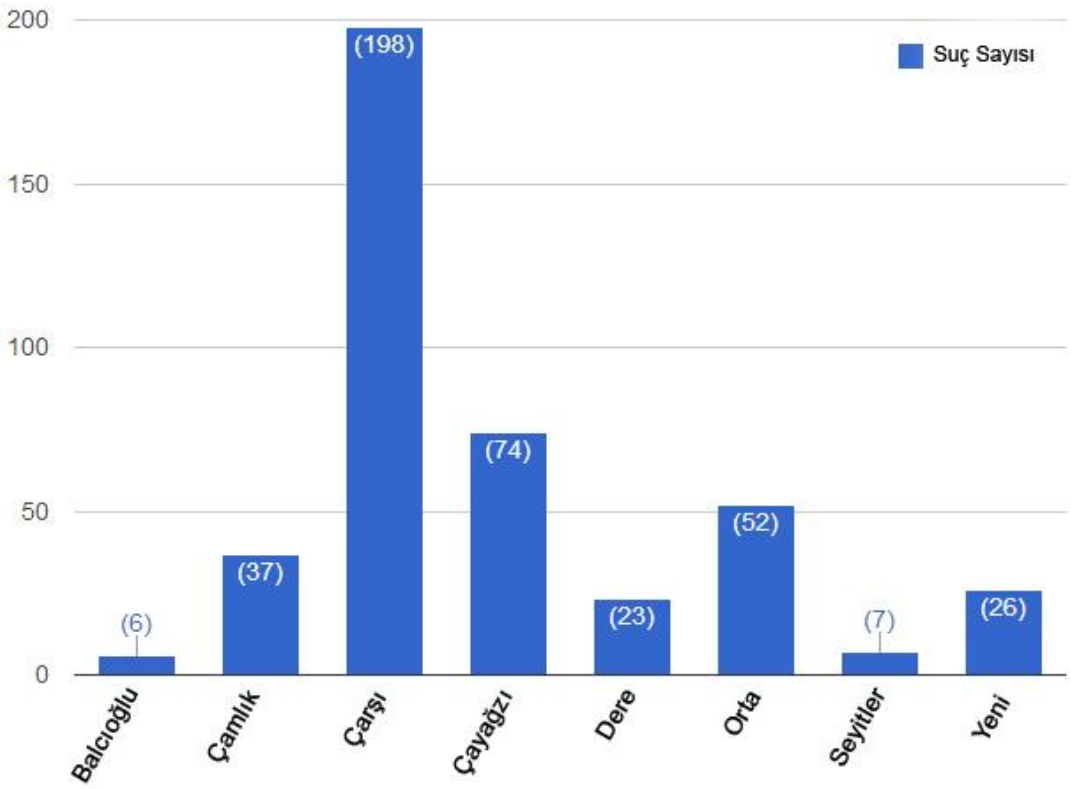


Şekil 4.5. Suç verilerinin mahalle alanına göre pasta grafiği

Çalışmada 423 suç verisinden rastgele alınan 30 suç verisi test veri kümesi olarak kullanılmıştır. 393 suç verisi CR-Ω+ sınıflandırma algoritmasına tabi tutularak eğitim kümesindeki verilerin sınıflandırması gerçekleştirilmiştir. Sınıflandırma iki şekilde yapılmıştır. İlk sınıflandırmada suçun türü sınıflandırılmıştır, ikinci sınıflandırma da ise suçun gerçekleştiği mahalle sınıflandırılmıştır. Şekil 4.6'daki grafikte görüldüğü üzere suç türü alanında dolandırıcılık, hırsızlık, hürriyete karşı suç, kavga ve trafik kazası olmak üzere 5 adet suç türü bulunmaktadır. Şekil 4.7'deki grafikte görüldüğü üzere mahalle alanında ise Balcıoğlu, Çamlık, Çarşı, Çayağzı, Dere, Orta, Seyitler ve Yeni olmak üzere 8 adet mahalle bulunmaktadır.



Şekil 4.6. Suç türüne göre suç verilerinin dağılımı



Şekil 4.7. Mahalle alanına göre suç verilerinin dağılımı

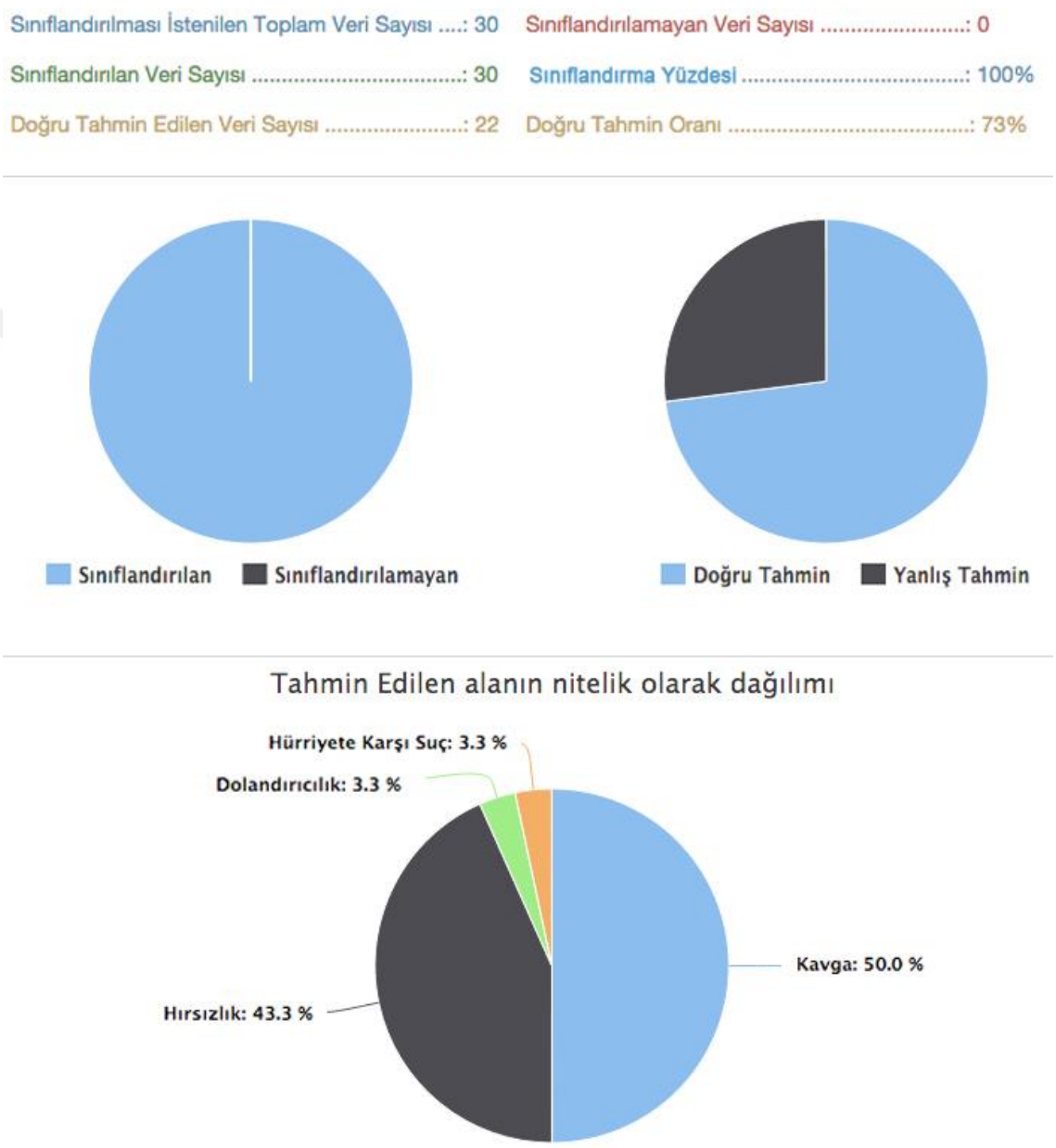
İki sınıflandırma çalışmasında da algoritma Bölüm 3.7.2'de anlatıldığı gibi bir takım sabit değerler almakta ve bu sabit değerler algoritmanın sınıflandırma yüzdesini ve doğru tahmin yüzdesini değiştirmektedir. Birinci sınıflandırma çalışmasında sabitler $\beta_1^+ = 5,\ \beta_1'^+ = 2,\ \beta_1^- = 5,\ \beta_1'^- = 2\ ,\ \beta_2^+ = 2,\ \beta_2'^+ = 5$ olarak alınmış ve Çizelge 4.1'deki otuz adet test verisi, algoritmanın tahminde bulunması için algoritmaya tanıtılmıştır.

Çizelge 4.1. Birinci tahmin işlemi için algoritmaya verilen test kümesi

No	Ay	Mahalle	Saat Dilimi
1	Ocak	Orta	4
2	Ocak	Çarşı	1
3	Ocak	Yeni	5
4	Ocak	Çayağzı	1
5	Ocak	Çayağzı	6
6	Şubat	Yeni	5
7	Şubat	Çarşı	1
8	Şubat	Çarşı	5
9	Şubat	Çamlık	8
10	Şubat	Orta	1
11	Şubat	Çayağzı	6
12	Mart	Çarşı	6
13	Nisan	Çarşı	2
14	Nisan	Orta	8
15	Nisan	Çarşı	1
16	Nisan	Orta	8
17	Nisan	Çarşı	3
18	Nisan	Çarşı	5
19	Mayıs	Çarşı	3
20	Mayıs	Çarşı	2
21	Haziran	Çarşı	4
22	Haziran	Çarşı	7
23	Haziran	Çayağzı	4
24	Haziran	Dere	1
25	Temmuz	Çayağzı	1
26	Ağustos	Çarşı	5
27	Ağustos	Çamlık	8
28	Aralık	Çarşı	1
29	Aralık	Çarşı	7
30	Aralık	Çarşı	8

Şekil 4.8'deki ekran görüntüsünde görüldüğü gibi ilk sınıflandırma çalışmasında verilerin tamamı sınıflandırılmıştır. Sınıflandırılan veriler içerisinde tahmin edilen sınıf

ile gerçek verinin sınıfı karşılaştırılmış ve algoritmanın doğru tahmin oranı %73 olarak belirlenmiştir.



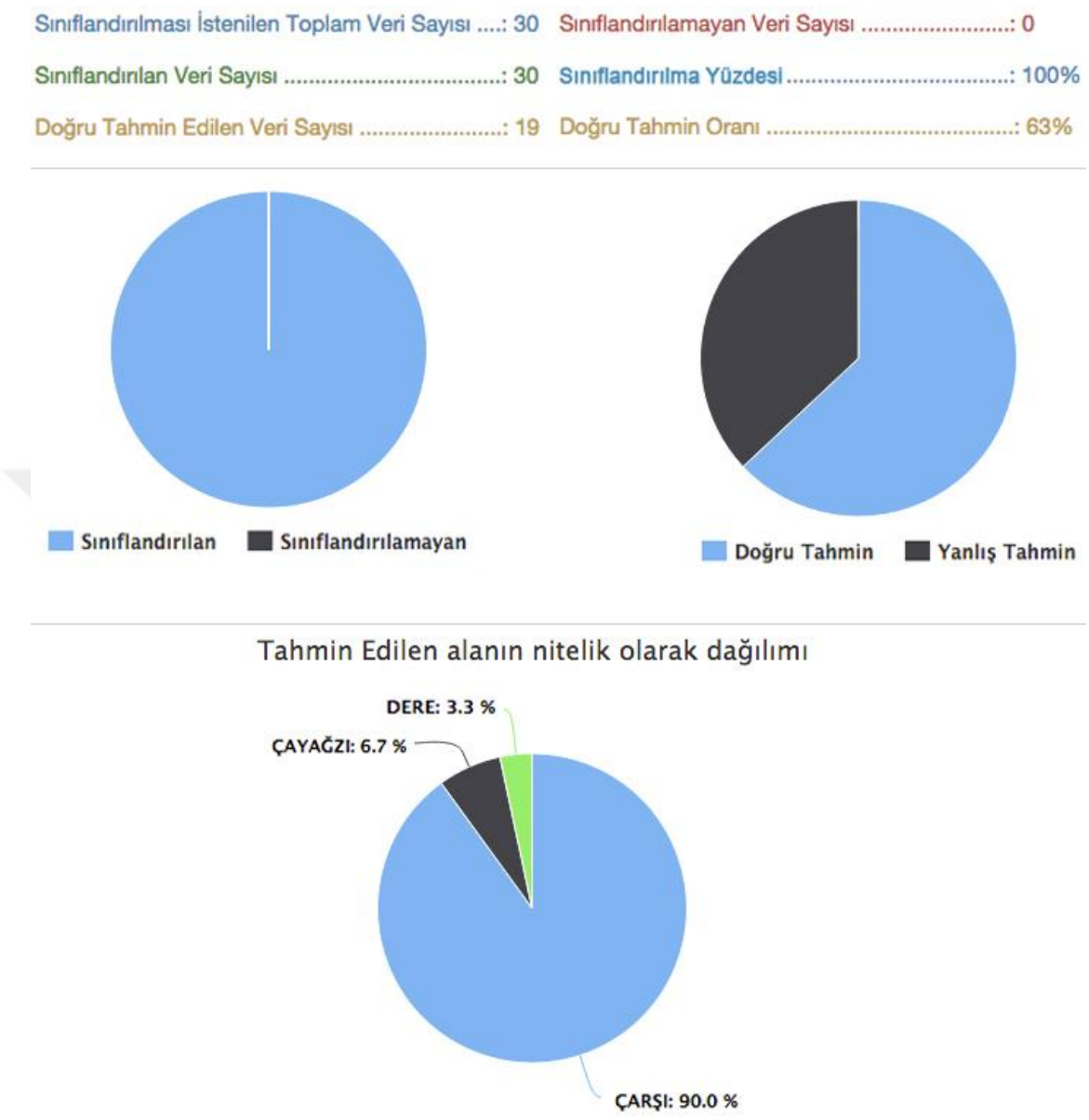
Şekil 4.8. Birinci işlem sonucu uygulamanın vermiş olduğu rapor

İkinci sınıflandırma çalışmasında ise Çizelge 4.2'deki test verisi algoritmaya tanıtılmıştır. Sabitler $\beta_1^+ = 4,\ \beta_1'^+ = 2,\ \beta_1^- = 4,\ \beta_1'^- = 2\,,\ \beta_2^+ = 2,\ \beta_2'^+ = 7$ olarak seçilmiştir. Şekil 4.9'daki ekran görüntüsünde görüldüğü gibi otuz adet test verisinin tamamı sınıflandırılmıştır. Sınıflandırılan veriler içerisinde tahmin edilen sınıf ile

gerçek verinin sınıfı karşılaştırılmış ve algoritmanın doğru tahmin oranı %63 olarak belirlenmiştir.

Çizelge 4.2. İkinci tahmin işlemi için algoritmaya verilen test kümesi

No	Ay	Suç Türü	Saat Dilimi
1	Ocak	Hırsızlık	6
2	Ocak	Hürriyete Karşı Suç	6
3	Şubat	Kavga	6
4	Şubat	Trafik Kazası	6
5	Şubat	Kavga	8
6	Şubat	Hırsızlık	7
7	Şubat	Trafik Kazası	4
8	Mart	Dolandırıcılık	4
9	Mart	Kavga	6
10	Mart	Kavga	1
11	Mart	Hırsızlık	5
12	Mart	Hırsızlık	4
13	Nisan	Hürriyete Karşı Suç	7
14	Nisan	Kavga	6
15	Nisan	Hürriyete Karşı Suç	5
16	Nisan	Trafik Kazası	7
17	Nisan	Dolandırıcılık	5
18	Nisan	Kavga	4
19	Mayıs	Hırsızlık	4
20	Mayıs	Hürriyete Karşı Suç	8
21	Mayıs	Hürriyete Karşı Suç	7
22	Mayıs	Kavga	8
23	Haziran	Hırsızlık	6
24	Haziran	Hürriyete Karşı Suç	3
25	Haziran	Trafik Kazası	6
26	Temmuz	Kavga	7
27	Ağustos	Hırsızlık	5
28	Eylül	Trafik Kazası	5
29	Ekim	Kavga	7
30	Aralık	Kavga	1



Şekil 4.9. İkinci işlem sonucu uygulamanın vermiş olduğu rapor

İki çalışmadaki sonuçlara bakınca ortaya çıkan uygulamanın, kolluk kuvvetleri için yararlı bir uygulama olduğu aşikardır. Uygulamayı kullanan kişiler tahmin edilmesini istedikleri verileri Excel belgesine, programın gösterdiği şekilde girdikten sonra, oluşturdukları Excel belgesini programa yükleyerek çeşitli durumlar için simülasyonlar oluşturabilirler. Kolluk kuvvetlerinin öncelikli amaçları halkın canını ve malını korumaktır. Bu amaca ulaşmak için kolluk kuvvetleri, suçu aydınlatmak, olması muhtemel suçu tahmin etmek, önünü almak ve şüphelileri yakalamakla yükümlülerdir.

Geliştirilen bu uygulama kolluk kuvvetlerindeki yöneticiler için karar aşamasında yardımcı bir unsur olacaktır.

Artvin Emniyet Müdürlüğünden alınan suç verilerinin sayısının ve suç verisi içerisindeki alanların az olması tahmin işlemini güçleştirmiştir. Şekil 4.6'da suç verilerinin türüne göre dağılımı, Şekil 4.7'de ise suç verilerinin mahalle alanına göre dağılımları verilmiştir. Bu dağılımlara baktığımızda verilerin orantısız olduğu gözükmektedir. CR-Ω+ algoritması veriler arasındaki alt kümelerin kesişme sayılarıyla ilişki kurup tahminde bulunduğu için veriler içerisinde bulunan orantısızlık, tahmin oranının düşüklüğüne neden olmaktadır.

Algoritma giriş parametreleri olan β_1^+, $\beta_1'^+$, β_1^-, $\beta_1'^-$, β_2^+, $\beta_2'^+$ değerlerine göre tahmin işlemi gerçekleştirmektedir. Algoritma veri kümesinde bulunan tüm verilerin alt kümelerine bakar, o özellik seti, o sınıf için en az β_1^+ defa görülüyor ve diğer sınıflar için ise en fazla $\beta_1'^+$ defa görülüyor ise o özellik seti o sınıf için pozitif karakteristiktir. Bunun tam tersi durumda ise negatif karakteristiktir. Buradan algoritmanın son kullanıcıdan aldığı parametrelere göre tahmin gerçekleştirdiği anlaşılmaktadır. Yani girilen parametreler algoritmanın tahmin oranına etki etmektedir. Dolayısıyla daha fazla veri, daha orantılı veriler ve daha düzgün giriş parametreleri ile algoritmanın tahmin oranının yüksek olacağı görülmektedir. Bu nedenle algoritmanın orantısız verilerde çalışamaması ve düzgün giriş parametrelerine ihtiyaç duyması, algoritmanın dezavantajlarıdır.

5. SONUÇ ve ÖNERİLER

Kolluk kuvvetlerinin öncelikli amaçları halkın canını ve malını korumaktır. Bu amaca ulaşmak için kolluk kuvvetleri, suçu aydınlatmak, olması muhtemel suçu tahmin etmek, önünü almak ve şüphelileri yakalamakla görevlidir. Kolluk kuvvetleri amaçlarına ulaşmak için birçok kaynak kullanırlar. Teknolojinin gelişmesiyle birlikte suç analizi, suç tahmini gibi çalışma alanları ortaya çıkmıştır. Bu çalışmada da Artvin Emniyet Müdürlüğü'nden alınan suç verileri ile suç tahmini ve analizi yapılmıştır. Alınan sonuçların kolluk kuvvetlerinin planlama ve taktik çalışmalarında kullanılması amaçlanmıştır. Alınan veriler CR-Ω+ sınıflandırma algoritmasına uygulanmış ve yapılan birinci çalışma neticesinde suç verileri %73 oranında doğru tahmin edilmiş, ikinci tahmin çalışmasında ise %63 oranında doğru tahmin edilmiştir. Ayrıca bu çalışma için geliştirilen web tabanlı uygulama ile suç analiz işlemleri başarılı bir şekilde yerine getirilmiştir.

Suç tahmini ve analizi gerçekleştirmek için yapılan uygulamada şuanda tek bir sınıflandırma algoritması bulunmaktadır ancak uygulama daha fazla algoritma eklenebilecek şekilde yapılmıştır. Gelecekte bu uygulamaya daha fazla sınıflandırma algoritması eklenerek, eklenen algoritmaların sınıflandırma ve tahmin performansları incelenebilir.

Artvin ilinin nüfusunun düşük olması alınan suç verilerinin sayısının az olmasına neden olmuştur. Veri sayısının az olması ve verilerin türe ve mahalleye göre dağılımlarının birbirlerine yakın olmayışı sınıflandırma ve tahmini zorlaştırmıştır. Daha fazla veri ve daha dengeli veriler ile daha iyi tahmin sonuçları alınabilecektir. Algoritmanın kural tabanlı olmayışı her simülasyon işlemi için tüm verilerin algoritma tarafından tekrardan işlenmesine neden olmaktadır. Ayrıca algoritmanın bir takım giriş parametreleri istemesi de sınıflandırma ve tahmin işlemlerinde doğruluk oranını artırıp azaltabilmektedir. Bu durumlar giderilerek algoritmanın geliştirilmesi sağlanabilir.

Çalışmada suç verilerinin elde edilmesi konusunda zorluklar yaşanmıştır. Kullanılan suç verileri için Erzurum Emniyet Müdürlüğü, Emniyet Genel Müdürlüğü ve Artvin Emniyet Müdürlüğü ile görüşmeler yapılmış, herhangi bir kişiye ait suç verisi talep etmememize rağmen sadece Artvin Emniyet Müdürlüğünden veriler alınabilmiştir. Artvin Emniyet Müdürlüğünden alınan verilerin sayısının az olması algoritmanın tahmin yüzdesinin düşük olmasına neden olmuştur. Daha fazla veri ile daha başarılı sınıflandırma işlemi yapılabileceği görülmektedir.

Sonuç olarak gerçekleştirilen suç tahmin ve analiz yazılımı kolluk kuvvetlerindeki yöneticilerin taktik ve planlama yapması için yararlı bir uygulamadır. Bu yazılım kullanılarak belirli bir ayda, saat diliminde ve mahallede tahmin edilmesi istenilen veriler tek tek yada Excel belgesi vasıtasıyla toplu bir şekilde girilerek tahmin işlemi gerçekleştirilebilmektedir. Program çıktıları kullanılarak bir takım durumlar simüle edilebilir ve simülasyon sonucuna göre ilgili birimler gerekli görülen bölgeye, zamanında yönlendirilerek suç oranlarının düşmesi sağlanabilir.

KAYNAKLAR

Akpınar, H., 2000.Veri tabanlarında bilgi keşfi ve veri madenciliği. İ.Ü. İşletme Fakültesi Dergisi, 29 (1), 1-22.

Alpaydın, E., 2000. Zeki veri madenciliği: ham veriden altın bilgiye. Bilişim 2000 Eğitim Semineri, İstanbul.

Anonymous, 2011. Weka programı, http://www.cs.waikato.ac.nz/ml/weka/index.html (05.11.2011).

Anonymous, 2012. Bayes teoremi, https://tr.wikipedia.org/wiki/Bayes_teoremi (04.11.2012).

Anonymous, 2013. ID3 algorithm, http://en.wikipedia.org/wiki/ID3_algorithm (09.12.2013).

Baysakoğlu, A., 2005. Veri madenciliği ve çimento sektöründe bir uygulama. Akademik Bilişim Konferansı, Gaziantep.

Birant, D., Kut A., Ventura M., Altınok H., Altınok B., Altınok E. ve Ihlamur M., 2010. İş zekası çözümleri için çok boyutlu birliktelik kuralları analizi. Akademik Bilişim Konferansı, Muğla.

Breiman, L., Friedman J., Olshen R. and Stone C., 1998. Classification and Regression Trees. Chapman&Hall /CRC, 368 p, USA.

Cabena, P., Hadjinian P., Stadler R., Verhees J. and Zanasi A., 1998. Discovering Data Mining: From Concept to Implementation, Prentice Hall, 224 p, NJ, USA.

Calderón, S., Calvo H., Martínez-Hernández V. and Moreno-Armendáriz M., 2010. The CR-Ω+ Classification Algorithm for Spatio-Temporal Prediction of Criminal Activity, Journal of Applied Research and Technology, 8 (1), 5-25.

Chapman, P., Clinton J., Kerber R., Khabaza T., Reinartz T., Shearer C. and Wirth R., 2000. Step-by-step data mining guide. CRISP-DM Consortium, 73s, USA.

Chelghoum, N., Zeitouni K. and Boulmakoul A., 2002. A Decision tree for multi-layered spatial data. Advances in Spatial Data Handling, Richardson D. Oosterom P. Springer, Berlin, 1-10.

Dunham, M., 2003. Data Mining Introductory and Advanced Topics. Prentice Hall, 315s, New Jersey, USA.

Egenhofer, M. and Sharma J., 1993. Topological Relation Between Regions in R2 and Z2. 5th International Symposium, SSD'93, Singapore.

Elmas, Ç., 2007. Yapay Zeka Uygulamaları. Seçkin Yayıncılık, 424 s, Ankara.

Ester, M., Kriegel H. and Sander J., 1997. Spatial data mining a database approach. 5th International Symposium, SSD '97, Berlin, Germany.

Fayyad, U., Shapiro G., Smyth P. and Uthurusamy R., 1996. Advance in Knowledge Discovery and Data Mining. AAAI Press / MIT Press, 611s, CA, USA.

Goldberg, D.E., 1989. Genetic Algorithm in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, 432p, MA, USA.

Grossman, R.L., Kamath C., Kegelmeyer P., Kumar V. and Namburu R., 2001. Data Mining for Scientific and Engineering Applications. Kluwer Academic Publisher, 605 p, MA, USA.

Gürsoy, U., 2009. Veri Madenciliği ve Bilgi Keşfi. Pegem Akademi, 192 s, Ankara.

Güvenç, E., 2001. Yüksek Öğrenimde Öğrenci Performansının Veri Madenciliği Teknikleri ile Belirlenmesi. Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Boğaziçi Üniversitesi, İstanbul.

Han, J. and Kamber M., 2000. Data Mining: Concepts and Techniques. Morgan Kaufmann, 743 p, CA, USA.

Hsu, W., Lee M. and Wang J., 2008. Temporal and Spatio-Temporal Data Mining. Igi Publishing, 292 p, Newyork, USA.

Ingebrigtsen, N., 2013. The differences between data, information and knowledge. http://www.infogineering.net/data-information-knowledge.htm (20.12.2013).

Kantardzic, M., 2011. Datamining Concepts, Models, Method and Algorithm. Wiley-IEEE Press, 284 p, NJ, USA.

Kass, G.V., 1980. An exploratory technique for investigating large quantities of categorical data. Applied Statistics, 29 (2), 119-127.

Larose, D., 2005. Discovering Knowledge in Data : An Introduction to Datamining. Wiley, 240 p, New Jersey, USA.

McGovern, A., Hiers C.N., Collier M., Gagne D. and Brown R., 2008. Spatiotemporal relational probability trees: an introduction. Eighth IEEE International Conference on Data Mining, Pisa, Italy.

Nabiyev, V., 2005. Yapay Zeka. Seçkin Yayıncılık, 766 s, Ankara.

Neville, J., Jensen D., Friedland L. and Hay M., 2003. Learning relational probability trees. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, DC, USA.

Özçınar, H., 2006. KPSS Sonuçlarının Veri Madenciliği Yöntemiyle Tahmin Edilmesi. Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Pamukkale Üniversitesi, Denizli.

Özkan, Y., 2008. Veri Madenciliği Yöntemleri. Papatya Yayıncılık, 224 s, İstanbul.

Öztemel, E., 2012. Yapay Sinir Ağları. Papatya Yayıncılık, 232 s, İstanbul.

Quinlan, J.R., 1986. Induction of decision trees. Journal of Machine Learning, 1(1), 81-106.

Quinlan, J.R., 1996. Improved use of continuous attributes in C4.5. Journal of Artificial Intelligence Research, 4 (1), 77-90.

Schumann, J. A., 2005. Data Mining Methodologies in Educational Organizations. PhD Thesis, University of Connecticut, Connecticut, USA.

Shannon, C.E., 1948. A mathematical theory of communication. The Bell System Technical Journal, 27 (3), 379-423.

Shaffer, J., Agrawal R. and Mehta M., 1996. SPRINT: A scalable parallel classifier for data mining. VLDB '96 Proceedings of the 22th International Conference on Very Large Data Bases, USA.

Silahtaroğlu, G., 2008. Kavram ve Algoritmalarıyla Temel Veri Madenciliği. Papatya Yayıncılık, 172 s, İstanbul.

Silahtaroğlu, G., 2013. Veri Madenciliği: Kavram ve Algoritmaları. Papatya Yayıncılık, 304 s, İstanbul.

Taha, H., 2000. Yöneylem Araştırması (Ş.Alp Baray, Şakir Esnaf Çev.). Literatür Yayıncılık, 900 s, İstanbul.

Tan, P., Steinbach M. and Kumar V., 2005. Introduction to Data Mining. Addisson Wesley , 568 p, NY, USA.

Timothy, A., McGovern A., Williams J. and Abernathy J., 2013. Spatiotemporal relational random forests. 13th International Conference on Data Mining Workshops, USA.

Troutman, N., 2010. Enhanced Spatiotemporal Relational Probability Trees and Forests. MS Thesis, University of Oklahoma, Oklahoma, USA.

Zahradník, J. and Skrbek M., 2013. Spatio-temporal data classification using CVNNs. Simulation Modelling Practice and Theory, 33, 81-88.

ÖZGEÇMİŞ

Muhammed Çağrı AKSU, 1987 yılında Erzurum'un Oltu ilçesinde doğdu. Lise öğrenimini Oltu YDA Lisesinde tamamladıktan sonra, 2005 yılında Harran Üniversitesi Bilgisayar Mühendisliği bölümünü kazandı. 2009 yılında lisans eğitimini tamamladı. Eğitimine, 2009 yılında başladığı, Atatürk Üniversitesi, Bilgisayar Mühendisliği bölümünde yüksek lisans yaparak devam etmektedir. Kasım 2009'dan itibaren Artvin Çoruh Üniversitesi, Enformatik Bölümü'nde Araştırma Görevlisi olarak görev yapmaktadır.