

T.C.
YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
İSTATİSTİK ANABİLİM DALI

**VERİ MADENCİLİĞİ SÜREÇLERİ VE
KARAR AĞAÇLARI ALGORİTMALARI İLE BİR UYGULAMA**

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Saygın DİLER
DANIŞMAN: Yrd. Doç. Dr. Sanem ŞEHRİBANOĞLU

VAN-2016

T.C.
YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
İSTATİSTİK ANABİLİM DALI

**VERİ MADENCİLİĞİ SÜREÇLERİ VE
KARAR AĞAÇLARI ALGORİTMALARI İLE BİR UYGULAMA**

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Saygın DİLER

VAN-2016

KABUL VE ONAY SAYFASI

İstatistik Anabilim Dalı'nda Yrd. Doç. Dr. Sanem ŞEHRİBANOĞLU danışmanlığında, Saygın Diler tarafından sunulan “**VERİ MADENCİLİĞİ SÜREÇLERİ VE KARAR AĞAÇLARI ALGORİTMALARI İLE UYGULAMA**” isimli bu çalışma Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili hükümleri gereğince 29/02/2016 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile başarılı bulunmuş ve Yüksek Lisans Tezi olarak kabul edilmiştir.

Başkan: Doç. Dr. H. Eray ÇELİK

İmza:

Üye : Yrd. Doç. Dr. Sanem ŞEHRİBANOĞLU

İmza:

Üye : Yrd. Doç. Dr. Naci MURAT

İmza:

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun .../.../ 2016 tarih ve/..... sayılı kararı ile onaylanmıştır.

İmza
Prof. Dr. Suat ŞENSOY
Enstitü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Saygın DİLER

ÖZET

VERİ MADENCİLİĞİ SÜREÇLERİ VE KARAR AĞAÇLARI ALGORİTMALARI İLE BİR UYGULAMA

DİLER, Saygın

Yüksek Lisans Tezi, İstatistik Anabilim Dalı

Tez Danışmanı : Yrd. Doç. Dr. Sanem ŞEHRİBANOĞLU

Şubat 2016, 118 sayfa

Bu tez çalışmasında, veri madenciliği süreci, yöntemleri ve sınıflandırma yöntemleri altında yer alan karar ağaçları algoritmaları incelenmiştir. TÜİK tarafından her yıl düzenli olarak yapılan ve ilk defa 2013 yılında il düzeyinde yapılan Yaşam Memnuniyeti Araştırması B grubu mikro verileri kullanılarak karar ağacı algoritmaları IBM SPSS Modeler 14.2 programı üzerinde uygulama yapılmıştır.

IBM SPSS Modeler 14.2 programında yer alan; C5.0 (ID3, C4.5), C&RT, CHAID ve QUEST algoritmalarının teorisi ile birlikte çalışma mantıkları anlatılmıştır. Bu algoritmaların farklılıkları ve ortak yönleri tez içerisinde belirtilmiştir. Yaşam Memnuniyeti araştırmasına ait ham verilerinin analizinde C&RT ve CHAID algoritması ile uygulamalar gerçekleştirilmiştir.

Anahtar kelimeler: Veri madenciliği, Karar ağacı algoritmaları, Yaşam memnuniyeti araştırması.

ABSTRACT

DATA MINING PROCESS AND AN EXAMPLE WITH DECISION TREE ALGORITHMS

DİLER, Saygın

M. Sc., Statistics

Supervisor : Asst. Prof. Dr. Sanem ŞEHRİBANOĞLU

February 2016, 118 pages

Within this research process of the data mining process, the methods of such and under the technical located decision trees have been examined. Life satisfaction is a research regularly conducted every year by TURKSTAT, firstly in province 2013. Group B of data making decision tree algorithms with IBM SPSS Modeler 14.2 were used.

C5.0 (ID3, C4.5), C&RT, CHAID and QUEST algorithms with the logic and the theory are described and is located in IBM SPSS Modeler 14.2. These algorithms are expressed in theses with both differences and common aspects. Applications with C&RT and CHAID algorithms to analyze the raw data of the Life Satisfaction survey were also conducted.

Key words: Data mining, Decision tree algorithms, Life satisfaction research.



ÖN SÖZ

Bu tez çalışmasında, her türlü ilgi ve yardımlarını esirgemeyen danışmanım Sayın Yrd. Doç. Dr. Sanem ŞEHRİBANOĞLU'na teşekkür ederim. Ayrıca tez jürimde bulunan Sayın Doç. Dr. H. Eray ÇELİK'e ve Sayın Yrd. Doç. Dr. Naci MURAT teşekkürlerimi sunarım. Türkiye İstatistik Kurumu'nda görev yapan ve desteklerini esirgemeyen çalışma arkadaşlarıma teşekkür ederim. Eğitim hayatım boyunca her zaman desteklerini hissettiğim babam Mustafa DİLER'e, Annem Türkan DİLER'e, abim Haldun DİLER'e ve biricik eşim Zuhal DİLER'e teşekkür ederim.

Şubat 2016
Saygın DİLER



İÇİNDEKİLER

	Sayfa
ÖZET	i
ABSTRACT	iii
ÖN SÖZ	v
İÇİNDEKİLER	vii
ÇİZELGELER LİSTESİ	ix
ŞEKİLLER LİSTESİ	xi
SİMGELER VE KISALTMALAR	xiii
EKLER	xv
1. GİRİŞ	1
2. VERİ MADENCİLİĞİ	6
2.1. Veri Madenciliği ve İstatistik	7
2.2. Veri Madenciliği Neler Yapamaz?	10
2.3. Veri Madenciliği Süreci (CRISP-DM)	10
2.3.1. İş hedeflerinin belirlenmesi (Business understanding)	11
2.3.2. Veriyi anlama (Data understanding)	11
2.3.3. Veri hazırlığı (Data preparation)	12
2.3.4. Modelleme (Modelling)	12
2.3.5. Değerlendirme (Evaluation)	12
2.3.6. Uygulama/ dağıtım (Deployment)	13
2.4. Metodoloji	14
3. VERİ MADENCİLİĞİ MODEL VE ALGORİTMALARI	15
3.1. Sınıflandırma (Classification)	15
3.2. Kümeleme (Clustering)	17
3.3. Birliktelik Kuralları (Association Rules)	18
4. KARAR AĞAÇLARI	20
4.1. Avantaj ve Dezavantajlarıyla Karar Ağaçları	21
4.2. Sınıflandırma İşleminin Doğruluğu	22
4.2.1. Holdout	23
4.2.2. K-fold cross validation	23
4.2.3. Leave-one-out	24

4.2.4. Bootstrap.....	25
4.3. Sınıflandırma Performanslarının Değerlendirilmesi.....	25
4.3.1. Karışıklık tabloları.....	25
4.3.2. ROC eğrileri	26
4.4. Karar Ağacı Algoritmaları.....	27
4.4.1. ID3 algoritması.....	30
4.4.2. C4.5 algoritması.....	40
4.4.3. C5.0 algoritması.....	40
4.4.4. C&RT algoritması	41
4.4.4.1. Twoing algoritması.....	42
4.4.4.2. Gini algoritması	49
4.4.5. CHAID algoritması.....	50
4.4.6. QUEST algoritması	55
5. UYGULAMA	58
5.1. IBM SPSS Modeler 14.2	58
5.1.1. IBM SPSS modeller kullanımı	59
5.2. Yaşam Memnuniyeti Araştırması	67
5.3. Uygulamanın Gerçekleştirilmesi	68
5.3.1. Veri setinin bölünmesi (training ve testing)	70
5.3.2. Otomatik sınıflama işlemcisi ile kullanılacak modelin belirlenmesi.....	70
5.3.3. CHAID algoritması.....	72
5.3.4. CHAID algoritmasının başarısının test edilmesi	82
5.3.5. CHAID algoritmasındaki önemli değişkenler için web grafiği.....	82
5.3.6. C&RT algoritması	83
5.3.7. C&RT algoritmasının başarısının test edilmesi.....	87
5.3.8. C&RT ve CHAID algoritmasının karşılaştırılması	88
6. SONUÇ VE TARTIŞMA	90
KAYNAKLAR.....	91
EKLER	96
ÖZGEÇMİŞ.....	97

ÇİZELGELER LİSTESİ

Çizelge	Sayfa
Çizelge 4.1. Karışıklık matrisi (Witten ve ark, 2011)	26
Çizelge 4.2. ID3 algoritması uygulanacak olan hava durumu verileri	32
Çizelge 4.3. Niteliklerin alabileceği değerler	32
Çizelge 4.4. Kök düğüm seçiminde bilgi kazancı	36
Çizelge 4.5. Güneşli hava durumu değerleri	37
Çizelge 4.6. Güneş hava durumuna göre bilgi kazancı	38
Çizelge 4.7. Bulutlu hava durumu değerleri.....	38
Çizelge 4.8. Yağmurlu hava durumu değerleri.....	39
Çizelge 4.9. C&RT algoritması uygulanacak veriler	43
Çizelge 4.10. C&RT algoritması için aday bölünmeler	43
Çizelge 4.11. Kök düğüm seçimi için ϕ (s t) değeri	45
Çizelge 4.12. İnşaat sektörü verileri	46
Çizelge 4.13. Bilişim sektörü	46
Çizelge 4.14. C&RT algoritması için aday bölünmeler	47
Çizelge 4.15. Bilişim niteliği için ϕ (s t) değeri	47
Çizelge 4.16. Gelir düzeyi verileri	48
Çizelge 4.17. C&RT algoritması için aday bölünmeler	48
Çizelge 4.18. Gelir niteliği için ϕ (s t) değeri	48
Çizelge 4.19. CHAID nitelikleri.....	51
Çizelge 4.20. Sağlık ve mutluluk değerleri için çapraz tablo	52
Çizelge 4.21. Gelir ve mutluluk değerleri için çapraz tablo	53
Çizelge 4.22. Umut ve mutluluk değerleri için çapraz tablo	54
Çizelge 4.23. Karar ağacı algoritmalarının karşılaştırılması.	57
Çizelge 5.1. CHAID ve C&RT ağacındaki değişkenler	89



ŞEKİLLER LİSTESİ

Şekil	Sayfa
Şekil 2.1. www.sciencedirect.com’da yayımlanan bilimsel makaleler (1996-2014).	7
Şekil 2.2. CRISP-DM döngüsü (Larose, 2005).	13
Şekil 2.3. CRISP-DM alt başlıkları (SPSS, 2000).	13
Şekil 2.4. Veri madenciliği çalışmasında kullanılan metodoloji (Alpaydın, 2000).	14
Şekil 3.1. Tanımlayıcı ve tahmin edici modeller (Kaya ve Köymen, 2008).	16
Şekil 4.1. Karar ağacı örneği (Silahtaroğlu, 2013).	21
Şekil 4.2. Model değerlendirme (Sınıflama Doğruluğu), (Souza ve ark., 2002).	23
Şekil 4.3. ROC eğrisi (Witten ve ark., 2011).	27
Şekil 4.4. ID3 ile kök düğüm seçildikten sonra veri setinin dallanması.	37
Şekil 4.5. ID3 Algoritması ile oluşturulan karar ağacı.	39
Şekil 4.6. C&RT ile kök düğüm seçildikten sonra veri setinin dallanması.	45
Şekil 4.7. C&RT algoritması ile oluşturulan örnek karar ağacı.	49
Şekil 4.8 CHAID algoritması ile oluşturulan örnek karar ağacı.	54
Şekil 5.1. Tercih edilen veri madenciliği programları (Muenchen, 2015).	59
Şekil 5.2 IBM SPSS modeller arayüzü.	60
Şekil 5.3. IBM SPSS modeller’da stream örneği.	67
Şekil 5.4. Genel mutluluk düzeyi.	69
Şekil 5.4. Genel mutluluk düzeyi.	69
Şekil 5.5. Reclassify işlemcisi ile dönüştürme.	69
Şekil 5.6. Verilerin partiton işlemcisi ile bölünmesi.	70
Şekil 5.7. Auto classifier işlemcisi.	71
Şekil 5.8 Auto classifier işlemcisi sonuçları.	72
Şekil 5.9. CHAID algoritmasına göre önemli değişkenler.	72
Şekil 5.10. CHAID algoritması ile oluşturulan karar ağacının genel görünümü.	73
Şekil 5.11. CHAID kök düğüm (node0) bölünmesi.	74
Şekil 5.12. CHAID node1 bölünmesi.	75
Şekil 5.13. CHAID node2 bölünmesi.	76
Şekil 5.14. CHAID node3 bölünmesi.	78
Şekil 5.16. CHAID node5 bölünmesi.	81
Şekil 5.17. CHAID algoritmasının test edilmesi.	82

Şekil	Sayfa
Şekil 5.18. CHAID algoritmasındaki önemli değişkenler için web grafiği.....	83
Şekil 5.19. C&RT algoritmasına göre önemli değişkenler.....	84
Şekil 5.20. C&RT algoritması ile oluşturulan karar ağacının genel görünümü.	84
Şekil 5.21. C&RT node0 bölünmesi.....	85
Şekil 5.22. C&RT node1 bölünmesi.....	86
Şekil 5.23. C&RT node2 bölünmesi.....	87
Şekil 5.24. C&RT algoritmasının test edilmesi.	88
Şekil 5.25. C&RT ve CHAID algortimalarının karşılaştırılması.	89



SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

Açıklama

%

Yüzde

Kısaltmalar

Açıklama

TÜİK

Türkiye İstatistik Kurumu

YMA

Yaşam Memnuniyeti Araştırması

C&RT

Classification and Regression Trees

CHAID

Chi-Square Automatic Interaction Detector

QUEST

Quick, Unbiased, Efficient Statistical Tree

AID

Automatic Interaction Detector

ID3

Iterative Dichotomiser 3

MARS

Multivariate Adaptive Regression Splines

SLIQ

Supervised learning in Quest

SPRINT

Scalable Parallelizable Induction of Decision Trees



EKLER

Ek

Sayfa

Ek 1. Oluřturulan Karar Aęaęlarında Kullanılan Deęiřkenlerin Soruları 96





1. GİRİŞ

Bilgisayar sistemlerindeki baş döndürücü ilerlemeler, aynı zamanda fiyatlarının ucuzlaması, işlemcilerinin ve kapasitelerinin artması; bilgisayarlarda büyük miktarda veri saklanabilmesine ve daha kısa sürede işlem yapılabilmesine olanak sağlamaktadır. Bu değişimler sonucunda teknoloji yaygınlaşmakta ve veriler toplanıp saklanabilmektedir. Böylece detaylı ve doğru bilgi üretilmektedir (Alpaydın, 2000).

Marketlerde, hastanelerde ve belediyelerde yapılan her işlem anında veritabanlarında yerini alır. Bütün bu kayıtlar bir yığın halinde depolanırken içlerinde bazı gizli bilgileri de barındırır. Bütün bu veriler, veritabanlarından çıkarılmayı bekleyen maden gibi durmaktadır. Kısacası etrafımızda bir sürü veri varken bu veriler bilgiye dönüşmeyi beklemektedir (Silahtaroglu, 2013).

Veriler üzerinden çözümlemeler yapmak amacıyla çeşitli istatistiksel ve matematiksel yöntemler kullanılabilmekte ancak veri sayısı arttıkça ortaya sorunlar çıkmaktadır. Özellikle ilişkisel veritabanları üzerinde bu çözümlemeleri yapmak zorlaşacaktır. Bu tür veriler üzerinde çözümlemeler yapabilmek için hem yeni veritabanı kavramlarına hem de yeni çözümleme yöntemlerine gereksinim duyulduğundan; veriyi yönetmek için “veri ambarı” ve verileri çözümleyerek “yararlı bilgiye” erişilmesini sağlayan “veri madenciliği” kavramları ortaya atılmıştır (Özkan, 2008).

Veri madenciliği ile ilgili çok çeşitli tanımlar yapılmaktadır. Bunlardan bazıları aşağıda verilmiştir;

Alpaydın (2000)’a göre, çok sayıdaki veri içinden gelecekle ilgili tahmin yapmamızı sağlayacak olan ilgili bağıntı ve kuralların bilgisayar programları kullanılarak elde edilmesidir.

Akpınar (2000)’a göre İstatistiksel ve matematiksel teknikler yardımıyla örüntü tanıma teknolojileri kullanılarak depolama ortamlarında saklanmış bulunan verilerin elenmesi ile anlamlı yeni korelasyon, örüntü ve eğilimlerin keşfedilmesidir.

Silahtaroglu (2013)’e göre ise, önceden hakkında fikir sahibi olunmayan, geçerli ve uygulanabilir bilgilerin büyük veritabanlarından elde edilmesi ile bu bilgiler doğrultusunda kararları belirlenmesinde kullanılır.

Lakshmanan ve ark. (2015) göre veri madenciliği, veritabanlarında tanımlanamayan bilgiyi keşfetmek için kullanılan bir teknolojidir.

Song ve Lu (2015)'e göre ise veri madenciliği, büyük veri kümelerinden mevcut bilgileri elde etmek ve bu bilgileri kolayca yorumlayabilmek için görselleştirmek için kullanılmaktadır.

Veri madenciliği, veritabanı ve veri ambarı teknolojisi, istatistik, makine öğrenmesi, yüksek performanslı bilgi işlem, örüntü tanıma, yapay sinir ağları, veri görselleştirme, bilgi alma, görüntü ve sinyal işleme ve mekansal ve zamansal veri analizi gibi farklı disiplinler ile birlikte gelişimini gerektirir (Mitra ve Acharya, 2003; Han ve Kamber, 2006).

Tüm bu tanımların ışığında diyebiliriz ki elde edilecek bilginin önceden bilinmiyor olması yani gizli kalmış tahmin edilemeyen bir bilgi olması ve işlem yapılan veri setinin çok büyük olması veri madenciliği tanımının ortak noktasıdır.

Aynı zamanda veri madenciliği için literatürde çok yaygın kullanılan bir terim de; veri tabanlarında bilgi keşfidir (Knowledge Discovery from Databases-KDD).

Veri madenciliği; veri ile içli dışlı olan hemen hemen bütün sektörlerde kullanılmaktadır. Bankacılık, pazarlama, sigortacılık, sağlık gibi değişik alanlar bunlardan bazılarıdır. Özetlenecek olursa;

Pazarlama

- Müşterilerin satın alma alışkanlıklarının belirlenmesi,
- Müşterilerin demografik özelliklerinin arasında bağıntıların tespit edilmesi,
- Müşteri ilişkileri yönetimi, müşterilerin elde tutulması,
- Satış tahmini.

Bankacılık

- Farklı finansal göstergeler arasındaki gizli ilişkilerin bulunması,
- Kredi kartı dolandırıcılıklarının tespiti,
- Kredi kartı harcamalarına göre müşteri gruplarının belirlenmesi,
- Kredi taleplerinin değerlendirilmesi.

Sigortacılık

- Yeni poliçe alacak müşterilerin tahmin edilmesi,
- Sigorta dolandırıcılıklarının tespiti,
- Riskli müşteri gruplarının belirlenmesi (Akpınar, 2000).

İçinde veri bulunan her alanda kullanılabilen veri madenciliği, veri tabanlarında bilgi keşfinin bir parçasıdır. Amaç gizli kalmış bilgileri bilgisayar programları yardımıyla ortaya çıkarmaktır. Veri madenciliği yöntemleri genel gördükleri işleme göre sınıflama, kümeleme ve birliktelik kuralları olmak üzere üç başlık altında toplanmaktadır (Özkan, 2008). Veri madenciliğinin uygulama alanı çok geniş olmakla beraber ve bu alanda yayımlanan birçok yayın bulunmaktadır.

Edelstein (1999) tarafından yayımlanan “Introduction to Data Mining and Knowledge Discovery”, Berry ve Linoff (2004) tarafından “Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management” ve Han ve Kamber (2006) tarafından “Data Mining: Concepts and Techniques” yayımlanan kitaplar veri madenciliği yöntemlerini, tekniklerini ve sürecini; veri madenciliği ile yapılabilecek ve yapılamayacak işlemleri anlatan kaynaklardan bazılarıdır.

Souza ve arkadaşları (2002) tarafından yayımlanan makalede sınıflandırma işlemi sonrasında algoritmalarının doğruluğunun test edilmesine yönelik geliştirilen yöntemlere değinilmiştir.

Özkan (2008) ve Silahtaroğlu (2013), veri madenciliği üzerine Türkçe kitaplar yazmıştır.

Alpaydın (2000) bilişim eğitim seminerinde yayımlanan dokümanda veri madenciliğinin örnek uygulama alanlarından, metodolojisinden ve tekniklerinden bahsetmiştir.

TÜİK uzmanı Babadağ (2003) tarafından yazılan uzmanlık tezinde veri madenciliği yöntemlerinin veri kalitesini arttırılmaya yönelik çalışmaları anlatılmıştır.

Oğuzlar (2003), veri madenciliği sürecinde daha başarılı sonuçlar elde edilmek için veri ön işlemin önemine değinmiştir.

Tüzüntürk (2010) ise veri madenciliğinin istatistik bilimi olan ilişkisine, benzerliklerine ve farklılıklarına değinmiştir.

Bu tez çalışmasında incelenen karar ağaçları, sınıflandırma yönteminin altında incelenmektedir. Karar ağaçları, hedef değişken üzerinde hangi faktörlerin etken olduğunu belirlemek ve bu faktörlerin geçmiş ilişkilerini ortaya koymakta yardımcı olan bir yöntemdir (Kayri, 2014). Ağaç oluşturulması için geliştirilmiş birçok algoritma vardır. Bunlardan bazıları;

Karar ağaçları için geliştirilen ilk algoritma 1970'lerin başında Morgan ve Sonquist tarafından geliştirilen AID algoritmasıdır (Emel ve Taşkın, 2005). G. V. Kass tarafından 1980'de geliştirilen CHAID algoritması istatistik tabanlı bir algoritmadır ve kategorik veriler için dallandırma işlemini ki-kare testi ile yapar (Silahtaroglu, 2013). Leo Breinman ve arkadaşları tarafından 1984 yılında hem sayısal hem de kategorik verilerle çalışabilen C&RT algoritması geliştirilmiştir (Özkan, 2008). J. Ross Quinlan tarafından 1986 yılında geliştirilen ID3 algoritması daha sonra 1993 yılında geliştirilerek C4.5 ve C5.0 algoritmaları ortaya atılmıştır (Çalış ve ark., 2014).

Rokack ve Maimon (2008) karar ağaçları ile teori ve uygulamasına odaklanmış veri madenciliği kitabı hazırlamıştır.

Shumei (2007), makalesinde C&RT ve CHAID karar ağacı algoritmalarını karşılaştırmıştır.

Ülkemizde yapılan bazı doktora çalışmalarında; Çakır (2008), veri madenciliğinde sınıflandırma yöntemlerini karşılaştırmıştır. Dondurmacı (2011), veri madenciliğinde regresyon ağaları ile uygulama gerçekleştirilmiştir. Yakut (2012), veri madenciliği tekniklerinden C5.0, Destek vektör makineleri ve yapay sinir ağlarının sınıflandırma başarılarının bir uygulama ile karşılaştırmıştır. Köktürk (2012), bazı veri madenciliği yöntemlerinden k-en yakın komşuluk, yapay sinir ağları ve karar ağaçlarını karşılaştırmıştır, Kuzey (2012) ise destek vektör makineleri ve karar ağaçları kullanılarak bir uygulama gerçekleştirilmiştir.

Çalış ve arkadaşları (2014), karar ağaçları algoritmaları ile uygulama üzerine bir makale hazırlamıştır.

Miller ve arkadaşları (2014), CHAID karar ağaçları ile genç yetişkinlerin metabolik sendromları erken teşhisi üzerine çalışma yapmıştır.

Lakshmanan ve arkadaşları (2015), akciğer tüberkülozu tedavisi için karar ağaçları ile veri madenciliği etkinlikleri üzerinde örüntü değerlendirmesi üzerine bir çalışma hazırlamıştır.

Song ve Lu (2015), sınıflandırma ve tahminleme uygulamaları için karar ağacı metotları üzerine bir çalışma yayımlamıştır.

Brandao ve arkadaşları (2015), veri madenciliği yöntemleri kullanılarak gebelik ile ilişkili risklerin tahmin edilmesine yönelik çalışma yayımlanmıştır.

Chen (2016), hileli finansal durumların tespiti için veri madenciliđi yaklaşımları çalışması yayımlanmıştır.

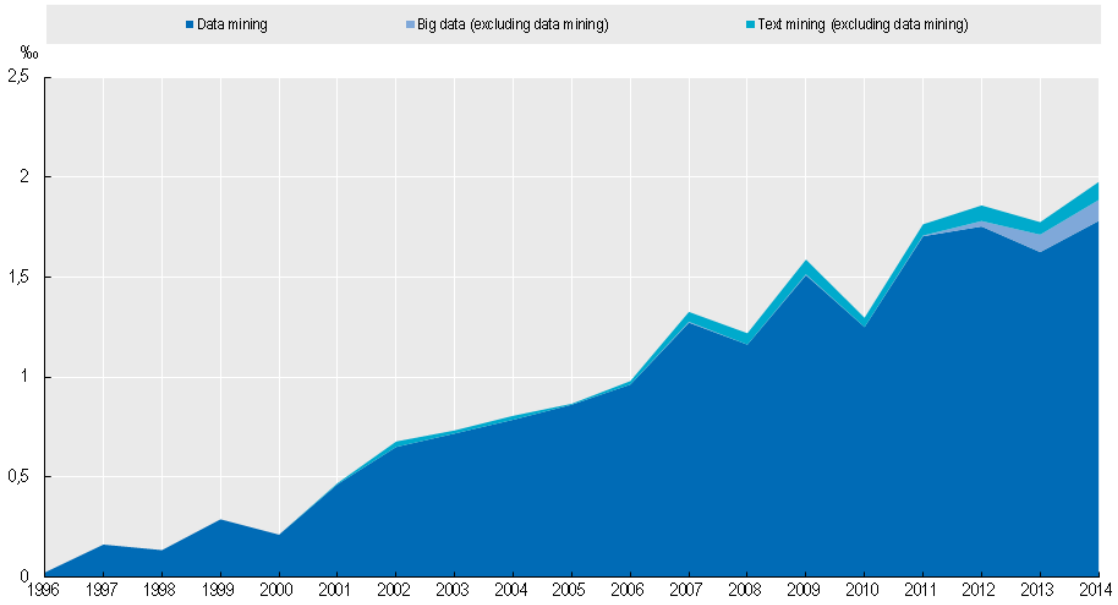


2. VERİ MADENCİLİĞİ

Terim olarak ilk defa 1960'larda istatistikçilerin kullandığı Data Fishing (veri avlama) veya Data Dredging (veri tarama) veri madenciliğinin ataları olarak anılabilir. İstatistikçiler bu terimleri detaylı analizlere başlamadan önce kötü verileri bulup ayıklamak için kullanmışlardır. Dilimize veri madenciliği olarak çevrilmiş olan Data Mining kavramı ise 1990'larda veritabanı uzmanları tarafından literatüre kazandırılmıştır (Pektaş, 2013).

Veri madenciliği bilgi teknolojilerinin doğal bir evrimi olarak görülebilir. Veritabanı sistemleri; veri toplama, veritabanı oluşturma, veri yönetimi ve yüksek veri analizi aşamalarından geçerek günümüze gelmiştir. 1960'lardan itibaren veritabanı ve bilgi teknolojileri basit dosya işlemlerinden güçlü veritabanı yapılarına doğru gelişim göstermiştir. 1970'li yıllarda veritabanı sistemlerindeki gelişme; ağ veri tabanı sistemleri, ilişkisel veri tabanı sistemleri, sorgu dilleri sayesinde verilere esnek erişim imkanı oluşturmuştur. 1980'lerden itibaren ise güçlü veritabanları ortaya çıkmış, internet tabanlı sistemler gelişim göstermiş, bilgi teknolojileri endüstrisinin gelişmesinde rol oynamıştır. Bunların yanında güçlü bilgisayarların da ortaya çıkmasıyla; eskiden veriyi elde etmek için harcanan çabanın yerini ileriye dönük tahminlerde bulunmak için ileri düzey analizlere bırakmıştır (Han ve Kamber, 2006).

Veri madenciliği üzerine yapılan araştırmalar, yazılan makaleler her geçen gün artmaktadır. OECD kütüphanesinden elde edilen şekil 2.1'den de görüleceği üzere 1995 yılından 2014 yılına kadar yayımlanan makale sayısında ciddi derecede artış bulunmaktadır. Veri madenciliği ilintili yazılan makalelerin yanı sıra *büyük veri* (Big Data) ve *metin madenciliği* (Text Mining) ile ilgili yapılan araştırmalarda da artış bulunmaktadır. OECD Şekil 2.1'deki verilerin hesaplamasını Science Direct isimli makale veritabanında veri madenciliği ilintili yayımlanan bilimsel makaleleri inceleyerek elde etmiştir (OECD, 2014)



Şekil 2.1. www.sciencedirect.com’da yayımlanan bilimsel makaleler (1996-2014).

2.1 Veri Madenciliği ve İstatistik

Veri madenciliğinde; veri tabanı, makine öğrenme ve istatistiksel bakış açısı olmak üzere üç farklı bakış açısı bulunmaktadır. Bu farklı bakış açılarına göre dökümanlar ve bilgisayar uygulamaları geliştirilmektedir. Veri madenciliğinin popülaritesi ve önemi yaygınlaşmaya başladıkça bu alanda ortaya konulan bilgisayar programları hızla artmaktadır. Ancak istatistik bu alanda yapılan çalışmalarda temel bir rol üstlenmektedir (Oğuzlar, 2003).

Veri madenciliği ve istatistik biliminde; ortak olan veriyi anlamlı hale getirerek ortaya bilgi çıkarmaktır, yani “veriden öğrenilmesi”, “verinin bilgiye dönüştürülmesi” hedeflenmektedir. Başka bir deyişle her iki yaklaşım belirsizlikleri çözmek ve gelecekte karşılaşılabilecek olaylar ile ilgili bilgi keşfi için kullanılır (Tüzüntürk, 2010).

Verilerin toplanması, özetlenmesi, grafik ve tablolarla sunulması, analiz edilerek kitle hakkında anlamlı bilgiler elde edilmesi ve yorumlar yapılması sırasında kullanılan yöntemlerin hepsine birden istatistik denir (Şenesen, 2013). Veri madenciliğinin amacı ile istatistik biliminin amacı birbirine paraleldir; buda verilerden bilgiyi keşfetmektir. Veri madenciliğinde kullanılan model ve yöntemler arasında istatistiksel tabanlı birçok algoritma mevcuttur, mesela karar ağaçlarında dallandırma işlemlerini gerçekleştirirken bağımlı değişken kategorik ise ki-kare, sürekli ise F testini kullanan CHAID algoritması

buna örnek olarak verilebilir. İkisinde de temel olan, veri ve bilgidir. Bundan dolayı istatistik ve veri madenciliğinin benzer yönleri bulunmaktadır. Veri madenciliği işlenen verilerden bulunan bilgileri eyleme geçirecek kararlar için kullanılmaktadır. Veri madenciliği bir açıdan istatistik biliminin teknolojiyle bütünleşmesi sonucu oluşan bir arayüzdür (Babadağ, 2003).

Veri madenciliği sürecinde yerinde ve efektif kararlar alınması için istatistik eğitimi almış birinin olması aşağıda bahsedilen konular için faydalı olacaktır:

- Çalışmada çıkabilecek problemlerin çözümlerine karar verilmesi,
- Sorunların cevaplanabilmesi için gerekli olan verilerin seçiminin ve düzenlenmesinin yapılması,
- Gerekli çözümlemenin yapılabilmesi ve doğru yöntemin belirlenebilmesi için verilerin özelliklerinin doğru bir şekilde belirlenmesi,
- Çalışılan konunun içeriği ile ilgili olası durumlarına göre yapılması gerekenin önerilmesi ve analiz sonucunda elde edilen bilgilerin yorumlanmasıdır (Feelder ve ark, 2000).

Veri madenciliğinin istatistik ile benzer taraflarıyla birlikte birçok farklılıkları da bulunmaktadır. Bunlar;

Bir hipotez, verileri analiz ederek önerinin geçerliliğini (kabul veya ret edilmesini) test eder (Berry ve Linoff, 2004). İstatistik doğrulayıcı bir süreç iken veri madenciliği teoremin doğruluğu üzerinde durmaz. Pek tabi bu da veri madenciliği yazılımlarının kendiliğinden örüntüleri bulacağı ve bilgiyi keşfedeceği anlamına gelmez. Aksine, bilgisayar programları veri madencisinden kendisine net bir şekilde yönlendirmesini bekler. Bunların yanında, veri madenciliği istatistik bilimiyle kıyaslandığında değişkenler arasındaki ilişkiler için daha az varsayımlarla kısıtlanmıştır (Tüzüntürk, 2010).

İstatistiksel araştırmada rastgele çekilen örneklem, ana kütle hakkında tahmin yapmak için kullanılır, araştırmanın örneklemin kendisi ile ilgilenmesi pek karşılaşılan bir durum değildir. İstatistik bireysellik ile uğraşmak yerine daha genel bilgilerin toplanmasına odaklanırken veri madenciliği detaylı, özel, gizli ve yerel bilgilerin toplanmasına odaklanır. Yani veri madenciliği tümünden gelim ile ilgilenirken, istatistik tüme varım ile ilgilenir (Tüzüntürk, 2010).

İstatistik biliminde önemli yeri olan hipotez testi, veri madenciliğinde önemli değildir. Çünkü veri madenciliği sürecinde hipotez testi ile veya ana kütlelerin sonuçlarının genelleştirmesi ile başlanılmaz. Veri madenciliğinde büyük hacimli veriler ile bazen de ana kütlede ki verilerin hepsi ile analiz yapılmaktadır. Bundan dolayı çıkarım yapmada kullanılan anlam düzeyinin önemi kalmaz (Zhau ve Luan, 2006).

Veri madenciliği ile istatistik arasında diğer bir farklılık ise verilerin toplanması ile ilgilidir. İstatistik araştırmaları için hazırlanan anket soru formları; belirli bir amaca yöneliktir. Sorular hazırlanır pilot çalışmaları yapılır ve daha sonra örneklem üzerinde alan çalışması ile veriler derlenir ve hazırlanan anketin amacına yönelik analiz yapılır. Ancak veri madenciliğinde, veriler bilgi keşfi sürecini uygulamak için toplanmaz, başka amaçlar için toplanan veriler üzerinde veri madenciliği çalışması yapılır (Zhau ve Luan, 2006).

Verinin hacmi yani büyüklüğü istatistik ile veri madenciliği arasındaki en göze çarpan farklılık olarak değerlendirilmektedir. İstatistikte veri setinin büyüklüğü yüzler veya binler ile ifade edilirken, veri madenciliğinde bu rakam milyon veya milyar ile ifade edilse abartı olmayacaktır.

Verilerin ön işleme ile çeşitli işlemlerden geçirilmesi veri madenciliğinde verinin kalitesi için gereklidir. Böylece temiz verilerden doğru çıktılar elde edilebilir. Fakat veri madenciliği sürecinde verilerin manipilasyonu oldukça zaman alıcı bir süreçtir hatta veri madenciliği sürecinin %70-80'i bu kısımda gerçekleşmektedir. Ancak istatistiksel araştırmalarda durum böyle değildir birçok durumda verilerin ön işlemeye tabi tutulmadan direkt analize geçilmektedir. Bu da iki disiplini ayıran önemli bir farktır (Tüzüntürk, 2010).

Veri madenciliği algoritmaları yardımıyla büyük hacimli veri setleri arasından gizli kalmış örüntüler veya ilişkiler çıkarılabilir. Ancak sadece başarılı bir veri madenciliği işlemi yapmak yeterli olmaz, son bir karar vericinin olması şarttır. Konuya hâkim olmak ve veriyi tanımak da veri madenciliği için çok önemlidir. Yapılan işlemler sonucunda ilişkileri bulmak veri madenciliği sürecinin bir parçasıdır. Bulunan sonuçların anlamlı veya faydalı olup olmadığı uzman kişiler tarafından karar verilmelidir. Konuya hâkim karar verici olmadığında veri madenciliği yöntemleri ile elde edilen veriler istihbarat düzeyinde kalır ve değerli olsalar bile, henüz yararlı bilgi varlığı olarak kabul edilemez (Babadağ, 2003).

2.2 Veri Madenciliği Neler Yapamaz?

Veri madenciliği bir araç ya da yönetilmesi gereken bir disiplindir. Analitik yöntemlerin, verinin ve işin iyi anlaşılması gibi ihtiyaçları ortadan kaldırmaz. Veri madenciliği iş analistine; verideki örüntü ve ilişkileri bulmasında yardımcı olur, örüntünün iş için değerini söylemez. Dahası veri madenciliği ile ortaya çıkarılmış örüntüler gerçek dünyada doğrulanmalıdır (Edelstein, 1999).

Unutulmamalıdır ki veri madenciliği ile bulunan tahmini ilişki ve örüntüler bir davranışın veya aksiyonun ille de sebebi olmayabilir. Örneğin, gelir düzeyi 5 bin TL ile 6.5 bin TL arasında ve belirli dergi veya gazetelere üye olan erkeklerin üretilen bir ürünü alma eğilimde oldukları veri madenciliği ile belirlenmiş olabilir. Bu bilgi ürünü pazarlama noktasında avantaj olabilir ancak bu kriterler bahsi geçen özelliklere sahip kişilerin üretilen ürünü alması için belirleyici faktör olarak varsayılmamalıdır (Edelstein, 1999).

Veri madenciliği veri toplama, veriler üzerinde sorgulamalar yapma veya gelişmiş analiz teknikleri kullanarak istatistiksel sonuç üretmenin ilerisindedir. Restoran zincirinin şubelerinin yaptığı ciro, en fazla satılan ürünlerinin tespiti, yoğunluk yaşanan zaman dilimleri gibi analizler veya bir perakende şirkette; devamlılık gösteren müşteriler, bölgelere göre performans düşüklüğünü ya da performans artışını saptamak veri madenciliği değildir. Gelir ve yaş arasındaki ilişki incelendiğinde bir model oluşturarak gelir ve yaş arasındaki ilişkiyi tanımlamakta değildir. Yüzlerce sayısal ve/veya kategorik değişkenlerin olduğu veri setine performanslı bilgisayarlar aracılığı ile çeşitli algoritmalar uygulanarak modellerin kurulabildiği bilgi keşfi süreci veri madenciliğidir. Algoritmalar, gelir ile yaş arasındaki ilişkiyi ortaya koyan doğrusal regresyon modelinden daha karmaşık olsa da amaç aynıdır, eldeki verilerden tahmin veya tanımlamalar yaparak bilgiye ulaşmaktır (Erşahin, 2008).

2.3 Veri Madenciliği Süreci (CRISP-DM)

Veri madenciliğinin farklı bilim alanları ile olan etkileşimi ve uygulama alanlarındaki çeşitlilikten, büyük hacimli verilerle çalışmasından ve veri madenciliği süresi içerisinde karşılaşılan güçlüklerden dolayı veri madenciliği sürecinden bilgi

yığınları arasında yolun kaybedilmemesi için standart bir yol haritasına ihtiyaç duyulmuştur.

Bazı danışmanlık şirketleri temelde benzer adımları içeren kendine özgü yöntemler geliştirmiştir. Bu yöntemlerden biri de 1996 yılında Daimler Chrysler, SPSS, AG, NCR ve OHRA gibi lider veri madenciliği kullanıcıları ve tedarikçilerinden oluşan bir konsoryum tarafından geliştirilmiş olan CRISP-DM (Cross-Industry Standart Process for Data Mining)'dir. Türkçe karşılığı “Endüstri Bağımsız Veri Madenciliği Süreci Standardı” olan CRISP-DM otomotiv, uzay endüstrisi, telekomünikasyon, eğitim ve sağlık gibi birçok sektörde kullanım alanı bulmuştur (Kocabaş, 2010). CRISP-DM süreci başarılı veri madenciliği için gerekli adımların anlaşılması ve kullanıcılara yardım etmesi iyi bir başlangıç olmaktadır (Edelstein, 1999).

CRISP-DM; veri madenciliği sürecinin hızlı, verimli ve düşük maliyetli gerçekleştirilebilmesi için geliştirilmiş ve altı adımdan oluşmaktadır (Brandão ve ark., 2015) Şekil 2.2 ve 2.3'te yer alan bu altı adımdan bahsedecek olursak (Larose, 2005).

2.3.1 İş hedeflerinin belirlenmesi (Business understanding)

Veri madenciliği sürecinin zor kısımlarından biri olan ilk aşamada amacın açıkça belirlenmesi ilk ve en önemli adımdır. Amacı doğru belirlenmemiş bir süreç yanlışlıklar üzerine inşa edilecek ve süreç sonunda ulaşılan yer amaca hizmet etmeyecektir. Çalışma sonucunda doğru cevaplanmış yanlış sorular elde edilmek istenmiyorsa, çalışmanın cevap aranan soru ile uyumlu olması gerekmektedir. İş tanımlama adımının diğer bir önemli unsuru ise personel, yazılım, bilişim kaynakları ve veri kümesini tanımlamaktır (Larose, 2005).

2.3.2 Veriyi anlama (Data understanding)

Veriyi toplamakla başlayan bu adım veri seti içerisinde hangi değişkenlerin olduğunu belirlenmesi, veri kalitesinin yeterli olup olmadığı ve verilerin neleri ifade ettiğinin anlaşılması ile devam eder. Şayet analiz yapacak kişi verilere hakim değilse uzman desteği almalı ve verileri kavramalıdır. Çünkü yanlış bir model oluşturulması muhtemeldir (Larose, 2005).

2.3.3 Veri hazırlığı (Data preparation)

Veri hazırlığı adımı, ham verilerden veri madenciliği sürecinde kullanılacak final veri kümesini oluşturmak amacıyla yapılan tüm işlemleri kapsamaktadır. Tablo oluşturma, değişken seçimi, veri temizliği ve dönüşümü bu aşamada yapılmaktadır. En önemli aşamalardan olan veri hazırlığı; veri madencisinin toplam zaman ve enerjisinin %50 - %90'ını harcamasına neden olmaktadır (Edelstein, 1999). Örnek olarak “Medeni Hal” alanında bulunan “Evli”, “evli”, “Evlenmiş” ya da “EVLİ” gibi kayıtların standartlaştırılarak “Evli” ye dönüştürülmesi bu adımda yapılır. Başka bir örnek verilecek olursa veri madenciliğinde kullanılacak olan değişkenlerin türlerinin (nominal, ordinal, flag vb.) belirlenmesi, kayıp değerlerin belirlenmesi ya da modellemeye katılması istenmeyen değişkenlerin filtrelenmesi, veriler üzerinde dönüşümlerin yapılması da bu adımda yapılmaktadır (Larose, 2005; Erşahin, 2008).

2.3.4 Modelleme (Modelling)

Akpınar’a (2000) göre tanımlanan problem için en uygun modelin bulunabilmesi, mümkün olduğunca çok sayıda modelin kurularak denenmesi ile mümkündür. Madenciliğe uygun hale getirilen veriye sorunu çözmeye yönelik veri madenciliği algoritmaları uygulanır. Aynı veri madenciliği problemi için farklı teknikler geliştirilmiştir (Larose, 2005). Yapılacak tahmin için veri madenciliği yöntemi belirlenmelidir, örneğin sınıflandırma yöntemi gibi, daha sonrasında ise model belirlenmelidir. Karar ağaçları gibi ancak bunların belirlenmesinde en önemli kriter veri setinin türüdür. Elimizdeki bağımlı ve bağımsız değişkenlerin kategorik ya da sürekli olması durumuna göre kullanabileceğimiz algoritma türleri değişkenlik göstermektedir (Edelstein, 1999).

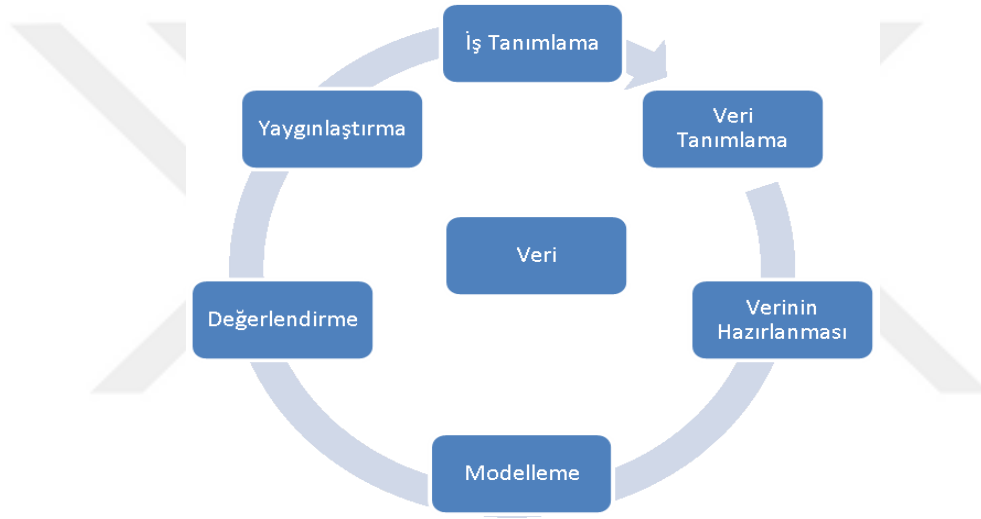
2.3.5 Değerlendirme (Evaluation)

Bu aşamada analistin elinde mevcut olan modelin iş amacına ne kadar uyup uymadığını değerlendirmesi gerekir, son aşama olan uygulama kısmına geçmeden önce bu adıma kadar yapılan bütün süreç gözden geçirilmelidir. Burada amaç bilgi keşfi

sürecinde gözden kaça bir şeyin olup olmadığının kontrol edilmesidir. Ayrıca eğitim verileri ile oluşturulan modelin ne kadar başarılı olduğunun test verileri ile sınanması bu aşamada gerçekleştirilir (Erşahin, 2008).

2.3.6 Uygulama/ dağıtım (Deployment)

Modelleme sonucunda elde edilen bilginin düzenlenmesi ve bu bilginin proje hedefleri ile birleştirilmesi gerekmektedir. İhtiyaç durumuna göre uygulama adımı raporlama veya başka sistemlerin içine entegre edilecek şekilde olabilir (Erşahin, 2008).



Şekil 2.2. CRISP-DM döngüsü (Larose, 2005).

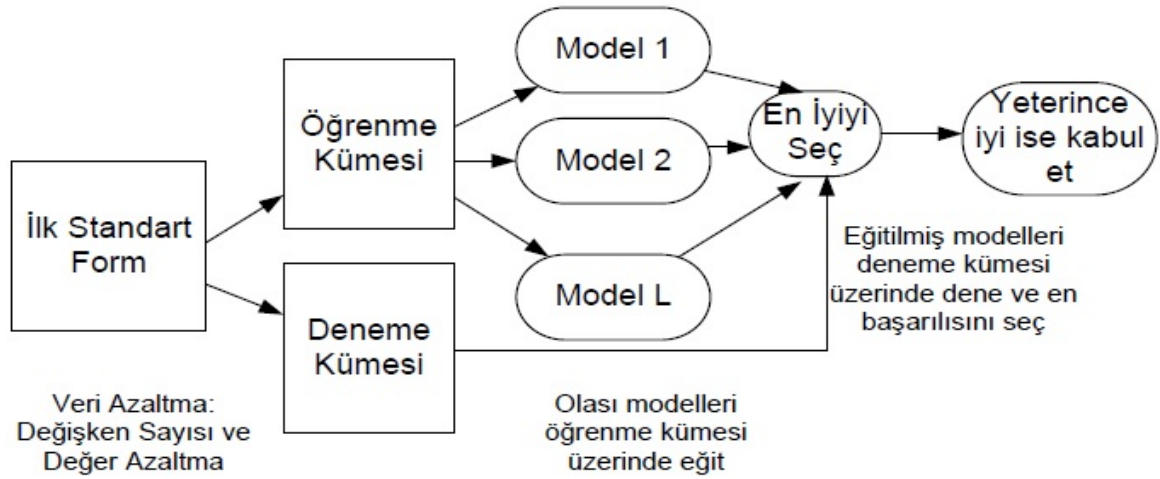
İş Tanımlama	Veriyi Tanımlama	Veriyi Hazırlama	Modelleme	Değerlendirme	Yaygınlaştırma
- İş Hedefini belirleme - Mevcut durumu değerlendirme - Veri Madenciliği Amacı belirleme	- ilk verilerin toplanması - verinin tanımlanması - verinin keşfedilmesi - veri alitesinin teyit edilmesi	- Veri seçimi - veri temizliği - Veri oluşturma - veri entegrasyonu	- Modelleme tekniğinin seçilmesi - Test tasarımının oluşturulması - Model inşa edilmesi - Modelin değerlendirilmesi	- Sonuçların değerlendirilmesi - sürecin gözden geçirilmesi - sonraki adımların belirlenmesi	- Planın konuşlandırılması - Planın gözetilmesi, bakımı ve sürdürülmesi - Final raporunun oluşturulması - Projenin gözden geçirilmesi

Şekil 2.3. CRISP-DM alt başlıkları (SPSS, 2000).

2.4 Metodoloji

Alpaydın'a (2000) göre veritabanındaki veriler öğrenme ya da eğitim (learning, training) ve deneme ya da test olmak üzere ikiye ayrılır. Uygulamalarda kullanılabilecek birçok teknik olduğundan hangisinin yararlı olduğunu seçmek zordur. Bunun için öğrenme kümesi üzerinde L adet değişik teknik kullanılarak L adet model elde edilir. Daha sonra elde edilen L adet model deneme kümesi üzerinde işleme konulur en başarılı olan model, yani deneme kümesi üzerindeki tahmin başarısı en yüksek olan model seçilir. Veri madenciliği metodolojisi Şekil 2.4'te verilmiştir.

Elde edilen en iyi model şayet yeterince başarılıysa kullanılabilir, aksi durumda başa dönerek çalışma tekrarlanmalıdır. Tekrar sırasında başarısız olan örnekler incelenir ve başarının nasıl arttırılabileceği araştırılır. Örnek olarak standart forma yeni değişkenler ekleyerek programa verilen bilgi arttırılabileceği gibi mevcut bilgiler dönüştürülebilir veya amaç daha farklı bir şekilde tanımlanabilir (Alpaydın, 2000).



Şekil 2.4. Veri madenciliği çalışmasında kullanılan metodoloji (Alpaydın, 2000).

3. VERİ MADENCİLİĞİ MODEL VE ALGORİTMALARI

İstatistik, bilgisayar bilimi ve veritabanı yönetimi gibi birçok disiplinin birleşmesinden oluşan veri madenciliğinde kullanılabilecek çok sayıda model ve algoritmalar vardır. Uygulamanın başarısı için seçilecek olan algoritmanın doğru seçilmesi çok önemlidir. Verinin yapısı seçilecek algoritma üzerinde etkili olduğu unutulmamalıdır.

İstatistikte de olduğu gibi veri madenciliğinde kullanılan modeller, tahmin edici (Predictive) ve tanımlayıcı (Descriptive) olmak üzere iki ana başlık (Şekil 3.1) altında incelenmektedir (Zong ve Zhou, 1999).

Özekes'e (2003) göre, tahmin edici modeller; sonuçları belli olan veri setinden yola çıkarak bir model geliştirilmesi ve kurulan model aracılığıyla sonuçları bilinmeyen veriler için sonuç değerlerinin tahmin edilmesi işlemidir. Tanımlayıcı modelleme işleminde ise karar verme işlemini gerçekleştirebilmek için mevcut verilerdeki örüntülerin tanımlanması sağlanmaktadır.

Veri madenciliği modelleri gördükleri işlevlere göre üç ana başlık altında toplanmaktadır (Özkan, 2008);

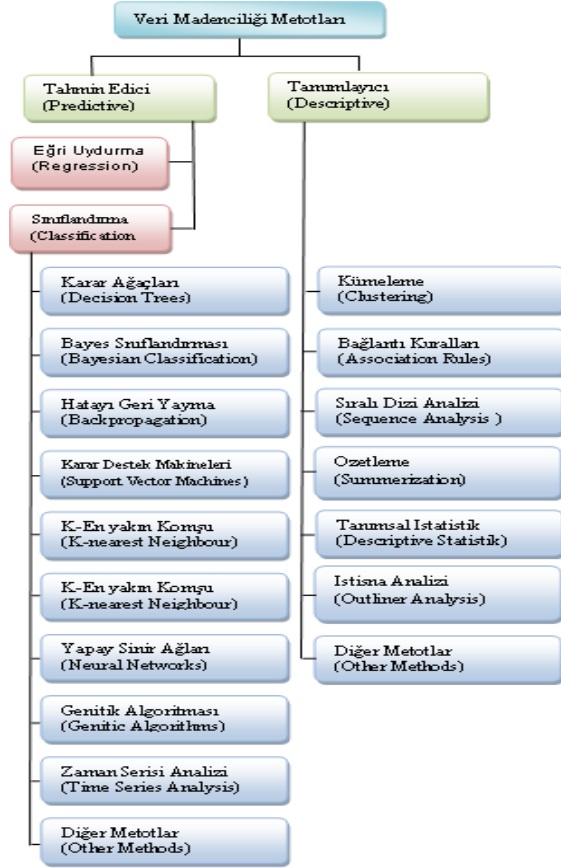
- a) Sınıflandırma (Classification)
- b) Kümeleme (Clustering)
- c) Birliktelik Kuralları (Association Rules)

Bu modellerin her biri için geliştirilmiş teknik ve algoritmalar vardır (Silahtaroglu, 2013). Sınıflandırma modeli tahmin edici, kümeleme ve birliktelik kuralları tanımlayıcı modellerdir (Özekes, 2003).

3.1 Sınıflandırma (Classification)

Sınıflandırma en çok bilinen ve yaygın olarak kullanılan veri madenciliği yöntemlerinden biridir. Resim, örüntü tanıma, hastalık tanıları, dolandırıcılık tespiti, kalite kontrol ve pazarlama sınıflandırma konularının bolca kullanıldığı alanlardır. Sınıflandırma tahminleyici bir modeldir; havanın bir sonraki gün seyri ya da bir kutudaki renkli topların tahmin edilmesi aslında sınıflandırma işlemidir (Silahtaroglu, 2013).

Sınıflandırma, her gözlemin karakteristiğinin önceden tanımlanmış gruplardan hangisine ait olduğunu bulmayı amaçlar, yani verinin içerdiği ortak özelliklere göre ayrıştırır.



Şekil 3.1. Tanımlayıcı ve tahmin edici modeller (Kaya ve Köymen, 2008).

Sınıflandırma iki adımdan oluşan bir süreçtir. İlk adım, veri tabanında bulunan değişkenlere uygun bir modelin ortaya konmasıdır. Model veri tabanında ki nitelikler kullanılarak gerçekleştirilir. Sınıflandırma modelinin işleminin gerçekleşmesi için verilerin bir kısmı eğitim verileri olarak kullanılır. Rasgele seçilen bu verilere algoritma uygulanır ve sınıflama modeli elde edilir. İkinci adımda ise elde edilen model sınıflandırma için kullanılır. (Han ve Kamber, 2006)

Sınıflandırma modellerinde kullanılan başlıca teknikler;

- Karar Ağaçları (Decision Tree)
- Yapay Sinir Ağları (Artificial Neural Networks)
- Genetik Algoritmalar (Genetic Algorithms)

- K-En Yakın Komşu (K-Nearest Neighbor)
- Regresyon (Regression)
- Bayesyen Sınıflandırma (Bayesian Classification)

Sınıflandırma modellerinin uygulandığı başlıca alanlar (Mitra ve Acharya, 2003);

- İmzaya duyarlı belge işlemlerinde ve bankalarda
(Uyuyor, uymuyor)
- Parmak izi belirlenmesi gibi parmak izi uygulamalarında
(Karşılıyor, karşılamıyor)
- Kredi kartı değerlendirmelerinde
(İyi, kötü)
- Banka konumunu değerlendirerek müşteri kalitesi ve iş olanakları
(İyi, vasat, zayıf)
- Görüntü kümesinden kimlik tespit etme
(Dost, düşman)
- Hastalıklara ilaçların tedavideki etkinliği
(İyi, orta, zayıf)

3.2 Kümeleme (Clustering)

Kümeleme, veritabanını farklı gruplara ayırır. Kümelemenin amacı, kendi içinde aynı özellikler taşıyan ancak birbirinden önemli farklar içeren farklı gruplar oluşturmaktır. Yani kümeleme işlemi sonunda elde edilen kümeler içerisinde yer alan nesneler arasındaki benzerlik maksimum; kümelerin ise birbirleri arasındaki farklılık maksimum olur. Veriler kümelere ayrılırken, benzerlik ve uzaklık kavramlarından yararlanılarak oluşturulur. Kümelemenin sınıflamadan farkı, başlangıçta oluşacak küme hakkında bir fikrin olmaması ve verilerin hangi özelliklere göre kümeleneceğinin bilinmemesidir. Sonuçların anlaşılabilmesi için küme hakkında bilgili olan birisinin kümeleri yorumlaması gerekmektedir. Küme oluştururken değişken seçimine dikkat edilmelidir. Bazı değişkenler üzerinde dönüşüm yapılarak yeni bir değişken oluşturulmuşsa, dönüşümde kullanılan değişkenler kümenin dışında tutulmalıdır, çünkü kullanıcı bu değişkenleri zaten gruplamada kullanmıştır (Özkan, 2008).

Kümeleme ile bölümlmeyi (segmentation) karıştırmamak gerekir (ancak bazı kaynaklarda kümeleme ile bölünme aynı anlamda da kullanılabilmektedir). Bölümleme, ortak özellik taşıyan grupların belirlenmesinin genel adıdır. Kümeleme ise daha önceden ortak özellikleri tanımlanmamış verilerin gruplara ayrılmasını sağlayan bir bölünme yöntemidir, sınıflamada ise gruplara atmak için önceden tanımlama yapılması gerekir (Edelstein, 1999).

Literatürde birçok kümeleme algoritmasının adı geçmektedir. Algoritmalar birbirlerinden kümelemenin oluşturuluş şekline ayrıldıkları gibi kullanılan veri türüne, yapılacak olan çalışmanın amacına göre de farklılıklar gösterirler. Kümeleme algoritmaları genel olarak hiyerarşik ve hiyerarşik olmayan olarak ikiye ayrılmaktadır (Silahtaroglu, 2013);

- Hiyerarşik Yöntemler;
 - Toplaşım Kümeleme Algoritmaları,
 - Bölünür Kümeleme Algoritmaları.
- Hiyerarşik Olmayan Yöntemler;
 - Yer Değiştirmeli Algoritmalar,
 - Olasılıksal Algoritmalar,
 - K-Medoid Yöntemler,
 - K-Means Yöntemler,
 - Yoğunluğa Dayalı Algoritmalar,
 - Yoğunluğa Dayalı Bağlantılı Kümeleme,
 - Yoğunluk Fonksiyonlu Kümeleme.

3.3 Birliktelik Kuralları (Association Rules)

Birliktelik kuralları, büyük hacimli veri setleri arasından ilişki bağlantılarını ve ilgi çekecek ilişkileri bulunması amacı ile kullanılır. Büyük veritabanlarında müşterilerin davranışlarını yakalamaya çalışır. Birliktelik kuralları bir ürün satın alındığında, o ürünle beraber başka hangi ürünlerin alındığının belirlenmesini araştırır. Kullanıldığı en yaygın örnek pazar sepeti analizidir (market basket analysis). Sepet analizi aracılığıyla müşteriler tarafından alışverişte satın alınan hizmet, mal veya ürün gruplarının belirlenmesi amaçlanmaktadır. Analiz sonucunda elde edilen bilgiler

kullanılarak raf düzenlemelerini belirleyerek; satış oranını arttırabilir, etkili satış stratejileri geliştirilebilir, promosyonların daha etkili olması sağlanabilir (Timor ve Şimşek, 2008; Gürsoy, 2011).

Birliktelik kuralları aynı anda gerçekleşen olayların tanımlanmasında kullanılır. Örneğin;

- Bir giyim mağazasında ceket ve kemer alan müşterilerin %89'u pantolon satın almaktadır.
- Bir restoranda havyar siparişi veren bir kişi %95 olasılıkla votkada isteyecektir.
- Tatile gitmek üzere bir otelde yer ayırtan, ayrıca kredi kartı ile uçakta rezervasyon yaptıran çocuklu bir aile; %85 olasılıkla gideceği yerde arabada kiralayacaktır (Gürsoy, 2011).
- 40 yaş üzeri premium kredi kartı sahipleri önerildiğinde %80 oranında kredi kartlarına sigorta yaptırmaktadır (Alkan, 2007).

Birliktelik kuralları uygulanmasında iki kıstas bulunmaktadır. Bu kıstaslar güven (confidence) ve destek (support) ölçütleridir. Büyük hacimli veri setleri içerisinde gereksiz bilgilerin ayıklanarak anlam bilgilerin keşfedilmesinde bu iki kriter kullanılmaktadır. B olayı olduğu anda A olayının da gerçekleşme olasılığı “Güven” olarak ifade edilir. Kesişen olayların ortaya çıkma sayısının toplam olay sayısı içerisinde oranı ise “Destek” olarak ifade edilir. İki değerinde olabildiğince yüksek olması iki ürünün satın alınmasındaki bağıntının önemli olduğu anlamına gelmektedir (Timor ve Şimşek, 2008).

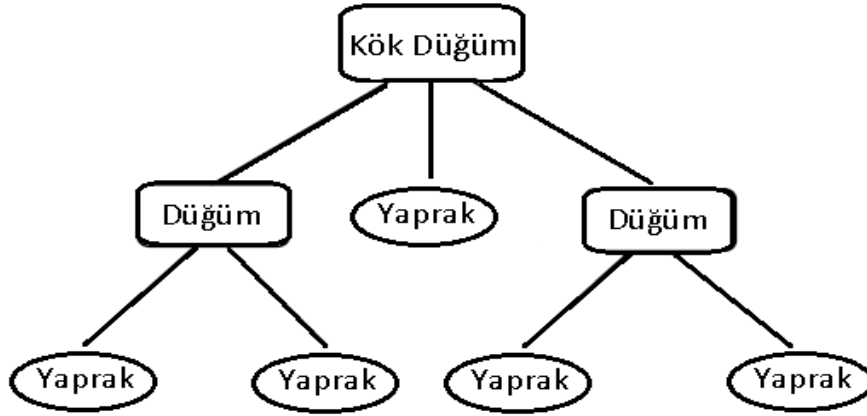
Birliktelik kuralları kullanım alanları içinde kategorik değişken olan her yer olarak tanımlanabilir. Appriori algoritması birliktelik kurallarında en bilinen ve en yaygın kullanılan algoritmadır.

4. KARAR AĞAÇLARI

Karar ağaçları sınıflandırma problemlerinde en çok kullanılan yöntemlerdendir. Diğer yöntemlerle kıyaslandığında karar ağaçlarının yapılandırılması, anlaşılması ve yorumlanması daha kolaydır. Bunların yanında karar ağaçlarının diğer bir avantajı ise başarılı modeller üretmesidir. Karar ağaçları yönteminde sınıflandırma yapmak için elimizdeki verilerden ağaç oluşturulur ve veri setindeki kayıtlar bu ağaca uygulanır, çıkan sonuca göre kayıtların sınıflandırma işlemi gerçekleşir. Yani veritabanından elde edilen karar ağaçlarına göre hangi sınıfa ait olduğunu bilmediğimiz bir veri elimize geldiğinde oluşturulmuş olan kural dizisine göre sınıfı tahmin edilir (Silahtaroğlu, 2013).

Sınıflarını bildiğimiz eğitim veri setinden tümevarım yöntemiyle ağaç şekilli karar yapılarının oluşturulması işlemine karar ağaçları denir (Dondurmacı, 2011). Bu ağaçlar, geniş hacimli veri setlerini daha küçük alt gruplara karar verme kuralları ile bölünerek oluşturur. Doğru ayrışma işlemleri ile elde edilen grupların üyeleri birbirleri ile daha benzer olmaktadır. Bir karar ağacı modeli büyük heterojen grupları belirlenmiş hedef değişkene göre yani kökten yaprağa doğru kural seti oluşturarak homojen alt gruplara ayırır (Berry ve Linoff, 2004). Karar ağaçları, hangi faktörlerin etken olduğunu belirlemek ve bu faktörlerin geçmiş ilişkilerini ortaya koymakta yardımcı olan bir yöntemdir (Kayri, 2014).

Karar ağaçları adından anlaşılacağı üzere ağaç şeklinde gösterilebilmektedir. Bu ağaçlar şekil 4,1'de görülebileceği üzere karar düğümlerinden, dal ve yapraklardan oluşmaktadır. Karar ağaçları ile sınıflandırma süreci kök düğüm ile başlar. Niteliklerin değerine göre düğümler alt dallara ayrılır, bu süreç yaprak elde edilinceye kadar devam eder. Burada düğümler; test işlemine tabi tutulan nitelikleri gösterir ancak bilinmelidir ki tüm değişkenlerin ya da niteliklerin karar ağacında yer alması beklenmez. Dallar; teste tabi tutulan düğümlerin sonuçlarını gösterir. Elde edilen dallar ile tahmin edilecek sınıfa giden yol belirlenir. Dalın sonucunda sınıflama tamamlanmıyorsa tekrar düğüm oluşturulur. Yapraklar ise veri setindeki bir karar sınıfını temsil etmektedir.



Şekil 4.1. Karar ağacı örneği (Silahtaroglu, 2013).

Karar ağaçlarının oluşturulması için önce girdi verisine, yani örneklere gereksinim vardır. Örnekler kategorik ya da nümerik değişkenlerden oluşabilir. Bunun yanı sıra hedef değişkene yani sınıflandırma için kullanılacak verilere ihtiyaç vardır.

4.1 Avantaj ve Dezavantajlarıyla Karar Ağaçları

Karar ağacı kökenli sınıflandırma yöntemlerinin bazı avantajlarından bahsedecek olursak (Murthy, 1998 ; Mitra 2003);

- Kolay anlaşılabilir ve yorumlanabilir olmasından dolayı daha kullanışlıdır,
- Çok büyük boyutlardaki veri setine kolayca uygulanabilir,
- Ağaç modelleri çıkarımsal modellerin aksine araştırmacıdır,
- Karar ağaçları non-parametriklerdir. Diğer tekniklerde olduğu gibi istatistiksel varsayımlarla uğraşmaz,
- Hiyerarşik ayrışması mevcut özelliklerinin ilişki yönünü ve önem sırasını ortaya koyar,
- Ağaç yapısından dolayı karışık bağlantılar ortaya çıkarılabilir,
- Sürekli ve kategorik niteliklerle çalışabilir,
- En çok kullanılan veri madenciliği modeli olduğu için bu alan da yayımlanmış birçok doküman ve bilgisayar programı bulunması,

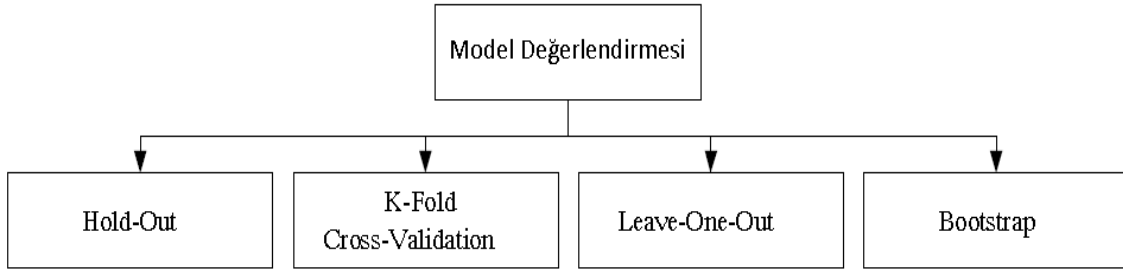
Karar ağaçlarının bazı dezavantajları ise (Myatt, 2007; Öğüdücü, 2008);

- Hesaplama maliyeti: özellikle büyük veri setlerinde çok sayıda sürekli değişkenin olması durumunda karar ağacı oluşturmak diğer sınıflandırma yöntemlerine nazaran daha maliyetli olabilir,
- En uygun ağacı bulmak bazen zor olabilir. Yararlı karar ağacını oluşturmaya çalışılırken birden karmaşık ve büyük ağaçlar üretilebilir,
- Girdi sayısı az olduğunda ağaç yeterli bilgiyi yakalamayabilir,
- Bazı algoritmalar sadece kategorik verilerle çalışır ve sürekli değişkenlerin sınıflamasında başarılı olmayabilir,
- Öğrenme kümesinin hacmine ve değişkenlerin sayısına bağlı olarak zaman ihtiyacı ve ağaç karmaşıklıkları değişmektedir.

4.2 Sınıflandırma İşleminin Doğruluğu

Sınıflandırma sürecinde karar ağacının eğitilmesinde kullanılacak olan verilerin seçilmesi gerekmektedir. Veri madenciliği süreci aşamalarında bahsetmiş olduğumuz veri hazırlığı aşaması sonucunda elde edilmiş olan analize tabi tutulacak temizlenmiş verilere bu aşamada ihtiyaç duyulmaktadır. Ancak yöntem kısmında bahsedildiği üzere bütün veriler ağacın eğitilmesinde kullanılmaz. Veriler ikiye bölünür; biri öğrenme ya da eğitim (raining, training) diğeri ise deneme ya da test (testing)'dir. Bu ikiye bölme işlemi oluşturulan ağacın doğruluğunu ölçmek için gerçekleştirilir. Bazı durumlarda veri seti bölünebilecek kadar büyük olmayabilir böyle durumlarda doğruluğu test etmek için geliştirilmiş yöntemler vardır. Yapılan işlemlerin sonunda doğruluk oranı tatmin edici düzeyde ise karar ağacı verilerin sınıflandırılmasında kullanılabilir. Aksi takdirde oluşturulan karar ağacı kuralları tahminleme işleminde kullanılamaz (Dondurmacı, 2011).

Sınıflandırma sürecinde eğitim verileri üzerinde kullanılan algoritmanın doğruluğunun belirlenmesi (model değerlendirilmesi) için geliştirilen birçok yöntem bulunmaktadır. Bu kapsamda en yaygın kullanılan yöntemler şekil 4.2'de belirtildiği gibi; *Holdout*, *K-fold Cross-Validation*, *Bootstrap* ve *leave-one-out* olarak sıralanabilir (Souza ve ark., 2002). Bu yöntemler sırasıyla aşağıda tanımlanmıştır:



Şekil 4.2. Model değerlendirme (Sınıflama Doğruluğu), (Souza ve ark., 2002).

4.2.1 Holdout

Veri seti eğitim ve test kümesi olmak üzere iki ayrı parçaya ayrılır. Eğitim verilerinden elde edilen sınıflandırma modelinin performansı, test verilerinde ölçülür. Holdout yönteminde verilerin genellikle $2/3$ 'ü eğitim, geriye kalan $1/3$ 'ü ise test için ayrılır. Sınıflandırmanın doğruluğu ise test kümesin üzerinden hesaplanır. Holdout yönteminin iki eksikliği bulunmaktadır; birinci olarak bu yöntem çok fazla veri olmadığı durumlarda test için verileri belirli oranlarda ayırmak mümkün olmayabilir. Diğer eksikliği ise isabetli veri ayrımı yapılamadığında yanıltıcı hata oranının bulunabilmesi ihtimalidir. Yani şöyle bir ikilem ortaya çıkmaktadır iyi bir sınıflandırma yapabilmek için mümkün olduğunca veriyi daha çok öğrenme verileri üzerinde uyguluyoruz ancak iyi bir hata oranı elde için ise verileri daha çok test verilerinde kullanmak isteriz. Bu ikilemden dolayı diğer yöntemler geliştirilmiştir (Witten ve ark., 2011).

4.2.2 K-fold cross validation

Çapraz geçerlilikten türetilen “K-katlı çapraz doğrulama” olarak adlandırılmaktadır. Çapraz geçerlilik tüm verileri kullanır. İlk veri seti üzerinden bir model oluşturulur ikinci model üzerinde bu model uygulanır ve hata oranı tespit edilir. Daha sonra ikinci veri seti üzerinden model oluşturulur daha sonra ise birinci model üzerinden hata oranı tespit edilir ve bu iki hata oranının ortalaması alınarak hata oranı elde edilir (Babadağ, 2003).

K-katlı çapraz doğrulamada ise veri seti k adet alt kümeye ayrılır. Uygulamalarda genel olarak $k=10$ alınır ve alt kümeler hemen hemen birbirlerine eşittir. Alt kümelerin belirlenmesinin ardından test kümesinin yeri değiştirilmek suretiyle hata oranları sırayla elde edilir. Verilerin 10 gruba bölündüğü varsayıldığında ilk grup test için kalan dokuz grup hep birlikte model oluşturulmada kullanılır. Model verinin %90'ı tarafından oluşturulur, bu işlem her seferinde test grubunun değiştirilmesiyle 10 kez tekrar edilir. Elde edilen 10 adet bağımsız hata değerinin ortalaması, kurulan modelin hata oranı olmaktadır. Uygulamada ve teoride oluşturulması gereken grup sayısının 10 olması yönünde bir görüş oluşmuştur. 10'lu çapraz geçerlilik standart bir yöntem olmaya başlasa da, yine de oluşturulacak grup sayısı analist tarafından belirlenir (Babadağ, 2003)

Veri tabanlarının birkaç bin ya da daha az satırdan meydana gelmesi durumunda kullanılması uygundur. Bu yöntemin sakıncası ise maliyetin k 'ya bağlı olarak artmasıdır.

4.2.3 Leave-one-out

Leave-one-out yöntemi, K-katlı çapraz geçerlilik yönteminin özel halidir. Bu yöntemdeki K sayısı veri kümesindeki örnek sayısına yani N 'ye eşittir. Model $N-1$ adet örnek eğitim verisi olarak kullanılır, dışarıda kalan 1 örnek ise test için kullanılır. Bu işlemi her örnek bir kere sınama için kullanılacak şekilde gerçekleştirilir (Özbudak 2009). Model başarısı ise yani sınıflamanın doğruluk oranı denemeler sonucunda elde edilen hataların ortalaması alınarak bulunur (Vercellis, 2009). İki sebepten dolayı çok kullanışlıdır. Birincisi; her durumunda mümkün olduğunca çok miktarda veri eğitim için kullanılır buda büyük olasılıkla doğru bir sınıflama yapma şansını yakalar. İkincisi yöntem deterministiktir: Rastgele örnekleme yoktur, 10 defa tekrarlama veya hepsini tekrarlama gibi bir durum da yoktur. Aynı sonuç her seferinde gözlemlenir (Witten ve ark., 2011).

Ancak örnek sayısı kadar yani N kere test yapılması gerektiği için çok büyük veri setlerinde kullanılması mümkün olmayabilir. Ayrıca çok fazla deneme olabileceği için yüksek hesaplama maliyeti dezavantajları arasında sayılabilir (Witten ve ark., 2011).

4.2.4 Bootstrap

En son olarak bahsedeceğimiz model hata tahmin yöntemi olan bootstrap, yerine koyarak istatistiksel örnekleme işlemine dayanır (Witten ve ark., 2011).

Örnekleme, anakütle parametrelerinin tahmin edilmesinde kullanılan istatistiksel yöntemdir. Parametrik istatistiksel yöntemlerin uygulanmadığı durumlarda test edilen hipotezdeki istatistiğin anlamlılığını ortaya koymak için kullanılan tekrarlı örnekleme yöntemleri içinde en iyi bilinen yöntemlerden biri Bootstrap yöntemidir (Dondurmacı, 2011).

Genellikle çok küçük veri setlerinde uygulanır, model tüm veri setinden oluşturulur. Daha sonra bootstrap olarak adlandırılan çok sayıda örneklem çekilir. Her gözlem örnekleme seçildikten sonra yerine konur ve tüm örneklem seçilene kadar işlem tekrar edilir. Bir kayıt birden fazla veri setinde yer alabilir. Model bu veri setinden oluşturulur ve hata oranı hesaplanır. En az 200 olmak üzere bazen de 1000'i geçecek şekilde birçok bootstrap örnekleme yaratılır (Kavurgacı ve ark., 2011) Model için tüm veri setinden oluşan son hata tahmin değeri bütün bootstrap örneklemlerinin hata tahmin değerinin ortalaması alınarak hesaplanır (Babadağ, 2003).

4.3 Sınıflandırma Performanslarının Değerlendirilmesi

Veri madenciliği yöntemlerinde kullanılmak istendiğinde pek çok sınıflandırma algoritması var olduğundan, hangi algoritmanın seçilmesi için geliştirilmiş birkaç kriter bulunmaktadır. Bunlar karışıklık (confusion) tablolarından hesaplanan; doğruluk (accuracy) hata oranı (error rate), anma (recall), kesinlik (precision) ve F-ölçütü (F-score) ile ROC eğrileri olarak gösterilebilir (Öğüdücü, 2008; Dalkılıç ve Dalkılıç 2015).

4.3.1. Karışıklık Tabloları

Modelin başarısı, doğru ve yanlış sınıflandırma işleminin gerçekleştirilmesi ile alakalı bir durumdur (Coşkun, 2010). Şekil 4.3'de yer alan karışıklık matrisinden doğruluk (accuracy) hata oranı (error rate), anma (recall), kesinlik (precision) ve F-ölçütü gibi değerler hesaplanmaktadır.

Çizelge 4.1. Karışıklık matrisi (Witten ve ark, 2011)

		Tahmin Edilen	
		Pozitif	Negatif
Gerçek Durum	Pozitif	TP(Doğru Olumlular)	FN (Yanlış Olumsuzluklar)
	Negatif	FP (Yanlış Olumlular)	TN (Doğru Olumsuzluklar)

$$Doğruluk = \frac{TP + TN}{Hepsi} \quad (4.1)$$

$$Hata Oranı = \frac{FN + FP}{Hepsi} \quad (4.2)$$

Doğruluk ve hata oranları yardımı ile modelin başarı durumu incelenir (Şeker, 2013). Doğruluk en basit değerlendirme ölçütüdür (Öğüdücü, 2008).

$$Kesinlik = \frac{TP}{TP + FP} \quad (4.3)$$

$$Anma = \frac{TP}{TP + FN} \quad (4.4)$$

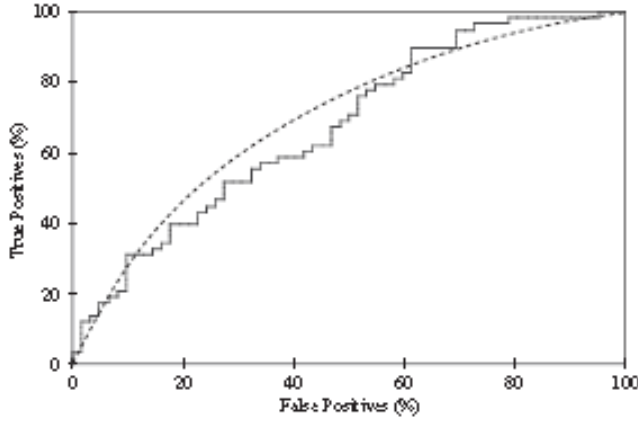
Modellerin karşılaştırılması durumunda “kesinlik” ve “anma” değerleri yüksek olan model daha iyi bir sınıflandırıcıdır denilebilir (Öğüdücü, 2008). “kesinlik” ile “anma” arasında ters orantı mevcuttur.

$$F - ölçütü = \frac{2 * kesinlik * anma}{kesinlik + anma} \quad (4.5)$$

F ölçütü “anma” ve “kesinlik” değerlerinin harmonik ortalaması ile bulunur (Dalkılıç ve Dalkılıç, 2015).

4.3.2. ROC Eğrileri

ROC (Receiver Operating Characteric) eğrileri farklı sınıflandırma tekniklerinin karşılaştırılması için kullanılır. Şekil 4.4’de yer alan ROC eğrisi yatay eksenle yanlışlıkla pozitif (FP) düşey eksenle ise doğrulukla pozitif (TP) değerlerini gösterir.



Şekil 4.3. ROC eğrisi (Witten ve ark., 2011).

ROC eğrisinin çiziminde “doğru pozitif oranı” (TPR-x eksen) ve “yanlış pozitif oranı” değerlerine ihtiyaç vardır. Bunlar;

$$TPR = \frac{TP}{TP + FN} \quad (4.6)$$

$$FPR = \frac{FP}{TN + FP} \quad (4.7)$$

Eğri üzerindeki her bir nokta bir sınıflandırıcının oluşturduğu alana karşılık düşer. TP ve FP değerleri ile çizilen ROC eğrisi grafikte yer alan değerler için;

(0,0): Bütün değerler negatif sınıflandırıldığını

(1,1): Bütün örnekler pozitif sınıflandırıldığını,

(0,1): Bütün negatif ve pozitif değerlerin doğru sınıflandırıldığını (en ideal durum),

(1,0): bütün negatif ve pozitif değerlerin hatalı sınıflandırıldığını ifade eder.

Bu durumlardan da anlaşılacağı üzere ROC eğrisi altındaki alanı 1’e daha yakın olan modelin sınıflandırma işleminde kullanılması tercih edilir (Dondurmacı, 2011).

4.4 Karar Ağacı Algoritmaları

Uygulamalı istatistik biliminde birçok karar ağacı algoritması “makine öğrenme” (machine learning) başlığı altında geliştirilmiştir. Örneklerden oluşan veri setinden yararlanarak karar ağacının oluşmasını sağlayan birçok öğrenme yöntemi bulunmaktadır (Özkan, 2008).

Karar ağaçları oluşturulurken hangi algoritmanın kullanıldığı çok önemlidir. Kullanılan algoritma çeşidine bağlı olarak oluşturulan ağaç yapısı farklılık göstermektedir. Bundan dolayı farklı ağaç yapıları sınıflandırma işlemini gerçekleştirirken farklı kurallar üretir. Karar ağaçlarına dayalı olarak geliştirilen algoritmalar kök düğümün ve diğer düğümlerin seçimlerinde izledikleri yol açısından birbirlerinden ayrılırlar, ancak bazı noktalarda ise birbirlerine benzemektedirler ya da benzerlik gösterirler (Silhataroğlu, 2013).

Karar ağaçları hedef değişkeni tahmin etmede pek çok test gerçekleştirerek en iyi sırayı kullanır. Karar ağaçlarında kullanılan temel algoritma adımları birbirlerine benzemektedir. Başlangıçta kök düğüm (root) oluşturulur. Eğer tüm örnekler aynı sınıfa ait ise düğüm yaprağa dönüşür, değilse test yapılarak bölünme gerçekleştirilir dallar (branch) oluşturulur. Dallar diğer testlerin gerçekleşmesine neden olur ve test işlemi yapraklarda (leaf) sonlandırılır (Bounsaythip ve Runsala, 2001). Örneklerin her bölümlemesinde aynı algoritmalar kullanılır. Genellikle kategorik değişkenler kullanılır eğer veriler sürekli ise kategorik hale dönüştürülür. Bölümlemenin sonlandırılması aşağıdaki adımlardan biri meydana geldiğinde gerçekleşir;

- Bir düğümde bulunan bütün örnekler aynı sınıfa ait ise,
- Örnekleri bölecek nitelik kalmadığında,
- Başka örneklem kalmadığında.

Karar ağaçlarında kökten başlayarak hedef yaprağa doğru giden bu yol, hedef değişkeni sınıflandıran kural olarak adlandırılmaktadır. Kurallar “eğer-sonra” (if-then) yapısındadır (Albayrak ve Yılmaz, 2009). Bu kurallar sayesinde karar ağaçları büyük ve karmaşık olsa bile kolay anlaşılabilir.

Karar ağaçları için geliştirilen algoritmalara kabaca aşağıda sıralanmıştır. Daha sonra ID3, C4.5 ve C5.0, C&RT, CHAID ve QUEST algoritmaları detaylı açıklanmıştır;

- **AID (1970)** (Automatic Interaction Detector): Karar ağaçları için ilk algoritma ve yazılım Morgan ve Sonquist tarafından 1970’lerin başlarında geliştirilmiştir (Emel ve Taşkın, 2005).

- **CHAID (1980)** (Chi-Squared Automatic Interaction Detector): G. V. Kass tarafından 1980'de geliştirilmiştir. Bölme işlemi olarak ki-kare testlerini kullanır. Dalların sayısı iki ile sınıf kategorisi arasında değişir (Silahtaroglu, 2013).
- **C&RT (1984)** (Classification and Regression Trees): Leo Breinman, Charles J. Stone, Jerry Friedman ve R. Olshen tarafından 1984 yılında geliştirilmiştir. Hem sayısal hem de kategorik verilerle çalışabilir. Dallanma kriteri olarak Gini indeksinden yararlanır ve ikili ağaçlar üretir (Özkan, 2008).
- **ID3 (1986)** (Iterative Dichotomiser 3): J. Ross Quinlan tarafından 1986 yılında geliştirilmiştir. ID3 makine öğrenmesi ve bilişim teorisine dayanmaktadır. Entropi kavramından yararlanarak bilgi kazancından yararlanarak önemli gördüğü değişkeni tespit eder (Silahtaroglu, 2013).
- **Exhaustive CHAID (1991)**: Ayrıntılı CHAID Biggs D., Ville B. ve Sue E. tarafından 1991 yılında geliştirilmiştir. CHAID algoritmasının modifiye edilmiş versiyonudur. CHAID algoritmasına benzemekle birlikte hesaplama işlemleri fazla zaman almaktadır (Köktürk, 2012).
- **MARS (1991)** (Multivariate Adaptive Regression Splines): 1991 yılında Friedman tarafından geliştirilmiştir. Lineer regresyon analizi kökenli bir algoritmadır (Miner ve ark., 2009).
- **C4.5 ve C5.0 (1993)** Ross Quinlan tarafından 1993 yılında ortaya atılmıştır. C4.5 algoritması ID3 algoritmasının geliştirilmiş hali iken C5.0 algoritması da C4.5 algoritmasının geliştirilmiş halidir. Dallanma kriteri bilgi kazancından yararlanır ve büyük hacimli veri setlerinde kullanılmaktadır. Budama işlemi ise her yapraktaki hata oranına dayanır. ID3 algoritmasından ayrılan bir yönü ise karar ağacı oluşturulurken kayıp veriler hesaba katılmaz (Çalış ve ark., 2014).
- **SLIQ (1996) (Supervised learning in Quest)**: IBM Almaden Araştırma Merkezinde Mehta M., Agrawal R. Ve Rissanen J. tarafından geliştirilmiştir. Nümerik ve

kategorik verilerin sınıflanmasında kullanılabilir. Nümerik veriler ile işlem yaparken maliyeti azaltmak için ağaç oluşturulurken önceden-sıralama tekniği kullanır. Önce genişlik ilkesiyle çalıştığı için aynı anda birçok yaprağı oluşturur. Dallara ayırmada gini endeksini kullanır. Bu algoritmanın öne çıkan özelliği hızlı olması ve iyi sonuçlar veren karar ağaçları üretmesidir (Silahtaroglu, 2013).

- **SPRINT (1996)** (Scalable PaRallelzizable INduction of Decision Trees): Shafer J., Agrawal R. ve Mehta M. tarafından 1996 yılında geliştirilmiştir. (Shafer ve ark., 1996). Büyük veri kümeleri için idealdir. Tüm değişkenler için ayrı bir değişken listesi hazırlar. Her tabloda kullanılacak olan sınıf ve sıra numarası bulunur. Değişken sayısı kadar tablo oluşturulur ve sınıflandırma ağacı uygulanır. Dallandırma kriteri olarak Gini indeksini kullanır (Silahtaroglu, 2013).
- **QUEST (1997)** (Quick, Unbiased, Efficient Statistical Tree): Shih ve Loh tarafından 1997 yılında geliştirilmiştir. İkili ağaçlar üreten algoritmadır. Bunun sebebi, budama ve doğrudan durma kuralı gibi yöntemlerin ikili ağaçlarda kullanılmasıdır. Bu algoritma ile ağaç oluşturulurken, değişken seçimi ve bölünmeyi eş zamanlı olarak yapar. QUEST, ağacın dallanması esnasındaki önyargılı seçimin daha genel duruma getirilmesi ve hesaplama maliyetinin azaltılması amacıyla kullanılmaktadır (Çalış ve ark., 2014).

4.4.1 ID3 algoritması

İlk olarak J. Ross Quinlan tarafından Sydney Üniversitesinde geliştirilen ID3 algoritması, diğer değişkenler arasından, sınıflamada kullanılacak en ayırt edici değişkeni bulurken entropi kavramından yararlanır (Silahtaroglu, 2013). Entropi;

- Eldeki bilginin sayısallaştırılması,
- Beklentisizliğin maksimumlaştırılması,
- Belirsizlik ve rastgeleliğin ölçülmesi için kullanılır.

Entropi, 0-1 arasında bir değer alır. Elimizdeki bütün veriler tek bir sınıfa ait olduğunda entropi sıfır olur. Örneğin herkes aynı futbol takımını tuttuğu bir durumda

rastgele birine hangi takımı tuttuğunu sorduğumuzda alacağımız cevap belli olacaktır yani entropi sıfır değerini alır. Bütün olasılıklar eşit olduğu zaman entropi maksimum değerine ulaşır. Matematiksel olarak aşağıdaki gibi ifade edilir.

$$H(T) = -\sum p_i \log_2(p_i) \quad (4.8)$$

Karar ağacının oluşturulacağı veri setinin tamamının entropisi hesaplanır; ancak dallanma gerçekleştiği zaman bölümlere ayrılan her bir bölüm için entropinin yeniden hesaplanması gerekir. Buradaki $H(T)$ hedef niteliği temsil eder, p_i ise düğüm içinde i sınıfında olma olasılığıdır.

ID3 algoritması, veritabanında doğru dallandırma yapmak için bilgi kazancı (information gain) kullanır. Veritabanında dallandırmalar oluşukça doğru sınıflandırmalar için gereken bilgi azalacaktır. Kazanımı hesaplamak için: verilerin ham halinin entropisi ile her bir nitelik için hesaplanan entropilerin ağırlık toplamı arsındaki fark alınır. Bu fark hangi alt bölüm için büyük ise o alt bölüme doğru dallandırma işlemi gerçekleştirilir. Bilgi kazancının formülü aşağıdaki gibi ifade edilir;

$$Kazanç(X, T) = H(T) - H(X, T) \quad (4.9)$$

$$H(X, T) = \sum_{i=1}^n \frac{|T_i|}{|T|} \quad (4.10)$$

Buradaki amaç Kazanç (X, T) değerini maksimum yapan değeri bulmaktır. Bir örnek daha açıklayıcı olacaktır.

Söz konusu örnekler ID3 algoritmasını ortaya atan Quinlan'ın makalesinde kullandığı ve karar ağaçlarının anlatıldığı birçok çalışmada bahsedilen “hava durumu” verilerden oluşmaktadır (Quinlan, 1986).

Çizelge 4.1'deki veri setimizde; “Hava”, “Sıcaklık”, “Nem”, “Rüzgar” ve “Oyun Oynama” olmak üzere beş adet nitelik bulunmaktadır. Bu eğitim verisinde “Oyun Oynama” niteliği hedef (bağımlı ya da çıkış da denilebilir) sınıf değerleri içermektedir. Diğerleri ise bağımsız ya da girdi verilerini oluşturmaktadır. Aşağıdaki tabloda veri setimizdeki değişkenlerin alabileceği değerler yer almaktadır. Bu eğitim veri setinden

elde etmek istediğimiz amacımız karar ağacı oluşturmak ve daha sonra sınıfını bilmediğimiz bir değişken geldiğinde karar ağacında elde edeceğimiz kural yapısına göre hangi sınıfa ait olduğunu ortaya koymak.

Çizelge 4.2. ID3 algoritması uygulanacak olan hava durumu verileri

Sıra	Hava	Sıcaklık	Nem	Rüzgar	Oyun Oynama
1	Güneşli	Sıcak	Yüksek	Yok	Hayır
2	Güneşli	Sıcak	Yüksek	Var	Hayır
3	Bulutlu	Sıcak	Yüksek	Yok	Evet
4	Yağmurlu	Ilık	Yüksek	Yok	Evet
5	Yağmurlu	Soğuk	Normal	Yok	Evet
6	Yağmurlu	Soğuk	Normal	Var	Hayır
7	Bulutlu	Soğuk	Normal	Var	Evet
8	Güneşli	Ilık	Yüksek	Yok	Hayır
9	Güneşli	Soğuk	Normal	Yok	Evet
10	Yağmurlu	Ilık	Normal	Yok	Evet
11	Güneşli	Ilık	Normal	Var	Evet
12	Bulutlu	Ilık	Yüksek	Var	Evet
13	Bulutlu	Sıcak	Normal	Yok	Evet
14	Yağmurlu	Ilık	Yüksek	Var	Hayır

Çizelge 4.3. Niteliklerin alabileceği değerler

Nitelik	Değerleri
Hava	Güneşli, Bulutlu, Yağmurlu
Sıcaklık	Sıcak, Ilık, Soğuk
Nem	Normal, Yüksek
Rüzgâr	Var, Yok
Oyun Oynama	Evet, Hayır

“Oyun Oyna” hedef sınıf değerleri incelendiğinde; 9 adet “evet”, 5 adet “hayır” değeri olduğu gözlenir.

Oyun Oyna = {hayır, hayır, evet, evet, evet, hayır, evet, hayır, evet, evet, evet, evet, hayır}

Başlangıç olarak hedef sınıfa ait entropi değeri $H(T) = -\sum p_i \log_2(p_i)$ formülünden aşağıdaki gibi elde edilir;

$$H(Oyun) = -\left(\frac{5}{14} \log_2 \frac{5}{14} + \frac{9}{14} \log_2 \frac{9}{14}\right) = 0.940$$

Adım 1: Kök düğümün seçilmesi

İlk olarak karar ağacının oluşturulmasında hangi niteliğin kök düğüm olmasına karar vermemiz gerekir. Bunun için $H(Oyun)=0.940$ olarak bulmuş olunan entropi değeri kullanılarak; “Hava”, “Sıcaklık”, “Nem” ve “Rüzgâr” niteliklerinin her biri için bilgi kazançlarının bulunması gerekir.

a) Hava değeri için bilgi kazancı:

Hava değeri için 5 adet “güneşli”, 5 adet “yağmurlu ve 4 adet “bulutlu” şeklinde tekrarlanmış olarak üç farklı değer bulunmaktadır. Hava niteliği için entropi değeri;

$$H(Hava, Oyun) = \frac{5}{14} H(Hava_{güneşli}) + \frac{4}{14} H(Hava_{bulutlu}) + \frac{5}{14} H(Hava_{yağmurlu}) \text{ şeklindedir.}$$

Güneşli, bulutlu ve yağmurlu hava durumları için ayrı ayrı entropi değerleri;

$$H(Hava_{güneşli}) = -\left(\frac{3}{5} \log_2 \frac{3}{5} + \frac{2}{5} \log_2 \frac{2}{5}\right) = 0.971$$

$$H(Hava_{yağmur}) = -\left(\frac{2}{5} \log_2 \frac{2}{5} + \frac{3}{5} \log_2 \frac{3}{5}\right) = 0.971$$

$$H(Hava_{bulutlu}) = -\left(\frac{4}{4} \log_2 \frac{4}{4}\right) = 0$$

Hava durumlarına göre bulduğumuz entropi değerlerini $H(Hava, Oyun)$ entropi değerini bulmak için formülde yerine konulursa;

$$H(Hava, Oyun) = \frac{5}{14}(0.971) + \frac{4}{14}(0) + \frac{5}{14}(0.971)$$

= 0.693 elde edilir.

Son olarak sıra hava değeri için bulunmaya çalışılan bilgi kazancı değerine gelir.

$$Kazanç(Hava, Oyun) = H(Oyun) - H(Hava, Oyun)$$

$$= 0.940 - 0.693$$

$$= 0.247$$

Hava niteliği için bilgi kazancı 0.247 olarak elde edilir ve daha sonra bu işlemi sırası ile bütün nitelikler için sırayla yaparız.

b) Sıcaklık değeri için bilgi kazancı:

Sıcaklık değeri için 4 adet “soğuk”, 6 adet “ılık” ve 4 adet “sıcak” şeklinde tekrarlanmış olarak üç farklı değer bulunmaktadır. Sıcaklık niteliği için entropi değeri;

$$H(\text{Sıcaklık}, \text{Oyun}) = \frac{4}{14} H(\text{Sıcaklık}_{\text{soğuk}}) + \frac{6}{14} H(\text{Sıcaklık}_{\text{ılık}}) + \frac{4}{14} H(\text{Sıcaklık}_{\text{sıcak}})$$

şeklindedir.

Soğuk, ılık ve sıcak durumları için ayrı ayrı entropi değerleri;

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

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

$$H(\text{Sıcaklık}_{\text{sıcak}}) = -\left(\frac{2}{4} \log_2 \frac{2}{4} + \frac{2}{4} \log_2 \frac{2}{4}\right) = 1$$

Sıcaklık durumlarına göre bulunan entropi değerlerini $H(\text{Sıcaklık}, \text{Oyun})$ entropi değerini bulmak için formülde yerine konulursa;

$$\begin{aligned} H(\text{Sıcaklık}, \text{Oyun}) &= \frac{4}{14}(0.811) + \frac{6}{14}(0.918) + \frac{4}{14}(1.00) \\ &= 0.911 \end{aligned}$$

elde edilir.

Son olarak sıra sıcaklık değeri için bulunmaya çalışılan bilgi kazancı değerine gelir.

$$Kazanç(Sıcaklık, Oyun) = H(Oyun) - H(Sıcaklık, Oyun)$$

$$= 0.940 - 0.911$$

$$= 0.029$$

Sıcaklık niteliği için bilgi kazancı 0.029 olarak elde edilir.

c) Nem değeri için bilgi kazancı:

Nem değeri için 7 adet “yüksek” ve 7 adet “normal” şeklinde tekrarlanmış olarak iki farklı değer bulunmaktadır. Nem niteliği için entropi değeri;

$$H(Nem, Oyun) = \frac{7}{14} H(Nem_{yüksek}) + \frac{7}{14} H(Nem_{normal}) \text{ şeklindedir.}$$

Yüksek ve normal nem durumları için ayrı ayrı entropi değerleri;

$$H(Nem_{yüksek}) = -\left(\frac{4}{7} \log_2 \frac{4}{7} + \frac{3}{7} \log_2 \frac{3}{7}\right) = 0.985$$

$$H(Nem_{düşük}) = -\left(\frac{1}{7} \log_2 \frac{1}{7} + \frac{6}{7} \log_2 \frac{6}{7}\right) = 0.592$$

Nem durumlarına göre bulduğumuz entropi değerlerini $H(Nem, Oyun)$ entropi değerini bulmak için formülde yerine konulursa;

$$\begin{aligned} H(Nem, Oyun) &= \frac{7}{14} (0.985) + \frac{7}{14} (0.592) \\ &= 0.798 \end{aligned}$$

Son olarak sıra nem değeri için bulunmaya çalışılan bilgi kazancı değerine gelmiştir.

$$Kazanç(Nem, Oyun) = H(Oyun) - H(Nem, Oyun)$$

$$= 0.940 - 0.789$$

$$= 0.151$$

Nem niteliği için bilgi kazancı 0.151 olarak elde edilir.

d) Rüzgâr değeri için bilgi kazancı:

Rüzgâr değeri için 8 adet “yok” ve 6 adet “var” şeklinde tekrarlanmış olarak iki farklı değer bulunmaktadır. Rüzgâr niteliği için entropi değeri;

$$H(\text{Rüzgar}, \text{Oyun}) = \frac{8}{14} H(\text{Rüzgar}_{\text{yok}}) + \frac{6}{14} H(\text{Rüzgar}_{\text{var}}) \text{ şeklindedir.}$$

Yok ve var rüzgar durumları için ayrı ayrı entropi değerleri;

$$H(\text{Rüzgar}_{\text{yok}}) = -\left(\frac{2}{8} \log_2 \frac{2}{8} + \frac{6}{8} \log_2 \frac{6}{8}\right) = 0.811$$

$$H(\text{Rüzgar}_{\text{var}}) = -\left(\frac{3}{6} \log_2 \frac{3}{6} + \frac{3}{6} \log_2 \frac{3}{6}\right) = 1$$

Rüzgâr durumlarına göre bulduğumuz entropi değerlerini $H(\text{Rüzgâr}, \text{Oyun})$ entropi değerini bulmak için formülde yerine koyarsak;

$$\begin{aligned} H(\text{Rüzgar}, \text{Oyun}) &= \frac{8}{14} (0.811) + \frac{6}{14} (1.00) \\ &= 0.892 \end{aligned}$$

Son olarak sıra rüzgâr değeri için bulunmaya çalışılan bilgi kazancı değerine gelir.

$$\text{Kazanç}(\text{Rüzgar}, \text{Oyun}) = H(\text{Oyun}) - H(\text{Rüzgar}, \text{Oyun})$$

$$= 0.940 - 0.892$$

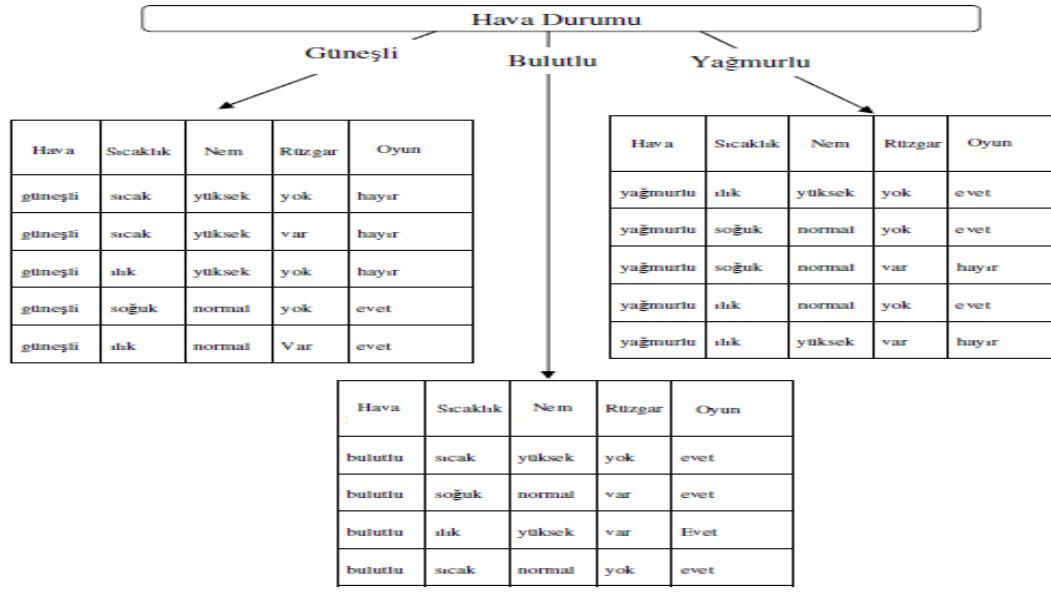
$$= 0.048$$

Rüzgâr niteliği için bilgi kazancı 0.048 olarak elde edilir.

Çizelge 4.4. Kök düğüm seçiminde bilgi kazancı

Bilgi Kazancı	Değerleri
Kazanç (Hava)	0.247
Kazanç (Sıcaklık)	0.029
Kazanç (Nem)	0.151
Kazanç (Rüzgar)	0.048

Yapılan bilgi kazancı hesaplamalar sonucunda Çizelge 4.3’den görüleceği üzere “Hava” niteliği 0.247 ile en fazla kazanç sağlayan nitelik olduğu hesaplanmıştır. Bu durumda kök düğüm olarak “Hava” niteliği seçilir daha sonra hava düğümünün “Güneşli”, “Bulutlu” ve “Yağmurlu” değerleri için dallandırma işlemi gerçekleştirilir. Böylece veri seti Şekil 4.3’de görüldüğü üzere hava durumuna göre üçe bölünür.



Şekil 4.4. ID3 ile kök düğüm seçildikten sonra veri setinin dallanması.

Adım 2: Güneşli hava durumuna göre dallandırma

Hava niteliğinin “Güneşli” değeri için karşılık gelen “Isı”, “Nem” ve “Rüzgar” değerleri için önce entropi daha sonra ise Bilgi Kazancı değerleri hesaplanır. Bu işlem sonucunda elde edilen bilgi kazancı Çizelge 4.5’de verilmiştir.

Çizelge 4.5. Güneşli hava durumu değerleri

Hava	Sıcaklık	Nem	Rüzgâr	Oyun Oynama
Güneşli	Sıcak	Yüksek	Yok	Hayır
Güneşli	Sıcak	Yüksek	Var	Hayır
Güneşli	Ilık	Yüksek	Yok	Hayır
Güneşli	Soğuk	Normal	Yok	Evet
Güneşli	Ilık	Normal	Var	Evet

Çizelge 4.6. Güneş hava durumuna göre bilgi kazancı

Bilgi Kazancı	Değerleri
Kazanç (Sıcaklık)	0.570
Kazanç (Nem)	0.970
Kazanç (Rüzgar)	0.019

Güneş dalından sonra tablodaki bilgi kazancı değerlerine bakılınca en büyük kazancın “Nem” değerinde olduğu gözlenir. Dolayısıyla dallandırma işlemi “Nem” değerlerine göre yapılacaktır. “Nem” niteliği incelendiğinde “yüksek” değerine karşılık sadece “hayır” yaprak değeri, “normal” değerine karşılık ise sadece “evet” yaprak değeri karşılık geldiği için bu noktadan itibaren dallandırma durdurulur. Böylece “güneşli” dalından yaprağa ulaşıldığı için diğer dallandırma işlemine geçilir.

Adım 3: Bulutlu hava durumuna göre dallandırma

Çizelge 4.6’dan görüldüğü üzere “Bulutlu” değerlerine karşılık gelen bütün değerler “evet” olduğu için herhangi bir analize gerek kalmadığı için dallandırma işlemi olmaz ve bu değere karşılık gelen yaprak sınıfı belirlenmiş olur.

Çizelge 4.7. Bulutlu hava durumu değerleri

Hava	Sıcaklık	Nem	Rüzgar	Oyun Oynama
Bulutlu	Sıcak	Yüksek	Yok	Evet
Bulutlu	Soğuk	Normal	Var	Evet
Bulutlu	Ilık	Yüksek	Var	Evet
Bulutlu	Sıcak	Normal	Yok	Evet

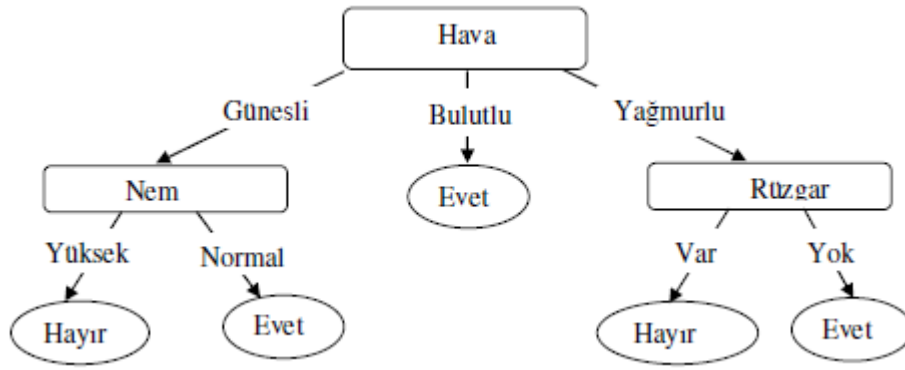
Adım 4: Yağmurlu hava durumuna göre dallandırma

Hava durumunun “Yağmurlu” değerinin dallandırma işlemi gerçekleştirilmek için diğer nitelikler için Bilgi Kazancı değerleri hesaplanır. Bu işlem sonucunda Rüzgar niteliği düğüm olarak seçilir. Rüzgar niteliğinin “yok” değerlerine karşılık sadece “evet” yaprak değeri, “var” değerlerine karşılık ise sadece “hayır” yaprak değeri denk geldiği için dallandırma işlemi sonlandırılır. Bu işlem ile karar ağacı oluşturma süreci bütün

veriler için sonlandırılmış olur ve böylece Şekil 4.6 daki gibi karar ağacı oluşturulmuş olur.

Çizelge 4.8. Yağmurlu hava durumu değerleri

Hava	Sıcaklık	Nem	Rüzgar	Oyun Oynama
Yağmurlu	Ilık	Yüksek	Yok	Evet
Yağmurlu	Soğuk	Normal	Yok	Evet
Yağmurlu	Soğuk	Normal	Var	Hayır
Yağmurlu	Ilık	Normal	Yok	Evet
Yağmurlu	Ilık	Yüksek	Var	Hayır



Şekil 4.5. ID3 algoritması ile oluşturulan karar ağacı.

Karar ağaçları oluşturulduktan sonra, Eğer-sonra (if-then) yapısı şeklinde sınıflandırma kurallarının çıkarılabileceğini belirtmiştik. ID3 Algoritmasına göre Hava Durumu örneğinde elde edilmiş olan sınıflama kuralları aşağıdaki gibi ifade edilebilir;

- **If** Hava = "güneşli" **and** Nem = "yüksek" **Then** Oyun = "Hayır"
- **If** Hava = "güneşli" **and** Nem = "normal" **Then** Oyun = "Evet"
- **If** Hava = "bulutlu" **Then** Oyun = "Evet"
- **If** Hava = "yağmur" **and** Rüzgar = "var" **Then** Oyun = "Hayır"
- **If** Hava = "güneşli" **and** Rüzgar = "var" **Then** Oyun = "Evet"

Algoritma mantığının kavratılması amacıyla bu algoritma örnek ve adımlarıyla detaylı açıklanmıştır. Diğer algoritmalar ID3 algoritması gibi örneklerle

açıklanmayacaktır. Ayrıca C4.5 ve C5.0 algoritmaları ID3 algoritmasının detaylandırılmış hali oldukları için bu algoritmalar arasındaki farklılıklar belirtilmiştir.

4.4.2 C4.5 algoritması

C4.5 Algoritması ilk olarak 1993 yılında Quinlan tarafından ortaya atılmıştır. ID3 Algoritmasının detaylandırılmış halidir. ID3 algoritması sadece kategorik verilerle uğraşırken C4.5 algoritması sayısal değişkenlere de uyarlanabilir (Özkan, 2008). Ayrıca C4.5 algoritması karar ağaçları oluştururken kayıp verilerle uğraşmaz. Kazanım oranını hesaplarken verileri eksik olmayan kayıtları kullanarak karar ağacı oluşturulur. Bu algoritma kayıp verileri diğer veri ve değişkenleri kullanarak kazanç oranı hesaplamada kullanır. Bu sayede daha duyarlı ve daha anlamlı kurallar üreten ağaç elde edilebilir (Silahtaroglu, 2013). C4.5 algoritmasını ID3 algoritmasından ayıran bir diğer farklılık ise ağacın budanması işlemidir. Budama işlemi için iki ayrı yöntem kullanılmaktadır, bunlar;

Alt ağaç yerleştirilmesi: ağaç içerisindeki alt ağaçlar yapraklara dönüştürülür. Ancak, bu değişimin yapılabilmesi için konulacak yaprak hatasının, alt ağacın hatasından düşük olması gerekir.

Alt ağaç yukarı çekilmesi: bir alt ağacın bu ağacı en çok kullanan ağacın yerini almasıdır. Ancak hata oranının bu işlem sonucunda düşmesi gerekir (Silahtaroglu, 2013)

4.4.3 C5.0 algoritması

C5.0 algoritması Clementine, Modeler ve RuleQuest vb. yaygın olarak kullanılan veri madenciliği yazılımlarında kullanılan C4.5 algoritmasının geliştirilmiş ticari bir versiyonudur (Terlemez, 2008) ve büyük veritabanların da kullanılması hedeflenmiştir (Yakut, 2012).

C4.5 algoritmasına göre temel farklılığı; hafıza kullanımında %90 oranında daha az bellek kullanması, hız konusunda ise 5,7 ila 240 kez arasında daha hızlı çalıştığı ve daha kesin kurallar ürettiği gösterilebilir (Terlemez, 2008). Bazı veri kümelerinde

bulunan hata oranının C4.5 ile bulunan yarısından daha az olduğu belirtilmiştir. Ayrıca C5.0 algoritması daha küçük karar ağaçları oluşturabilir, verinin ayrıştırılması ve kirli verilerin iyileştirilmesinde daha etkilidir (Kocabaş, 2010).

4.4.4 C&RT algoritması

Literatürde CART ya da C&RT olarak kısaltılan algoritmanın tam açılımı Classification and Regression Trees Türkçe karşılığı ise Sınıflandırma ve Regresyon Ağaçları olarak tanımlanır. C&RT algoritması ilk olarak Leo ve arkadaşları tarafından 1984 yılında yazıldı (Kuzey, 2012) C&RT algoritması kategorik veya sayısal değişkenlerle çalışabilme özelliğine sahiptir. Cevap değişken kategorik ise sınıflandırma ağacı (Classification Tree), sürekli ise regresyon ağacı (Regression Tree) olarak adlandırılır, uygulamalarda genel olarak C&RT şeklinde ifade edilmektedir (Yakut, 2012).

Bu algoritma ikili karar ağaçları oluşturur. Her aşamada düğümlerin kendinden daha homojen alt dallara ayrılması sağlanmaktadır. Ayrılma kriteri için geliştirilmiş yöntemler bulunmaktadır. Bunlar arasında gini, twoin, sıralı twoin, simetrik gini, en küçük kareler sapması (Least Squared Deviation) gibi yöntemler kullanılmaktadır. Bu yöntemlerin kullanımı sınıf hedeflerin sürekli ya da kategorik olmasına göre belirlenmektedir. En yaygın olarak kullanılan ayırma kriteri Gini ve Twoin kuralıdır, bu kurallar detaylı açıklanmıştır (Köktürk, 2012)

Veri seti üzerinde kök düğümünden başlanarak dallandırmalar yapılır ve durdurma kuralı olmadan yapraklara kadar ulaşılır böylece ağaç en yüksek büyüklüğüne ulaşır. C&RT algoritması çok karmaşık bir ağaç oluşturduktan sonra, çapraz geçerlilik (cross validation) ve test (testing) kümesi sınaması işlemi sonuçlarını dikkate alarak en uygun ağaca ulaşır. Daha sonra karar ağacı, test veri kümesi üzerindeki budanmış versiyonlarının başarılarını temel alarak geriye doğru budama işlemini gerçekleştirir. Karar ağacı, yeni ve daha önceden görülmemiş veriyi, çapraz geçerlilik sınaması ile seçer (Koyuncugil, 2006).

C&RT algoritmasında, vekil değişkenler kullanılmaktadır yani kayıp olan gözlem değeri için vekil belirlemekte ve kayıp değerleri bu vekil değişken ile doldurmaktadır (Koyuncugil, 2006).

4.4.4.1 Twoing algoritması

Nitelikler içerdği değere bağlı olarak sol ve sağ tarafa olmak üzere iki ayrı aday bölünme yapar. Aday bölünmelerin her biri için aşağıdaki değerler hesaplanır.

t : dallanmanın yapılacağı düğüm

P_{sol} : veri setindeki kaydın solda olma olasılığı

$P_{sağ}$: veri setindeki kaydın sağda olma olasılığı

$P(j|t_{sol})$: j sınıf değerinin sol taraftaki bölünmede olma olasılığı

$P(j|t_{sağ})$: j sınıf değerinin sağ taraftaki bölünmede olma olasılığı

Elde edilen bu değerlere bağlı olarak t düğümündeki s aday bölünmelerinin uygunluk ölçüsü aşağıdaki gibi hesaplanır;

$$\phi(s|t) = 2P_{sol}P_{sağ} \sum_{j=1}^n |P(j|t_{sol}) - P(j|t_{sağ})| \quad (4.11)$$

Bu işlem sonucunda bulunan en büyük uygunluk ölçüsüne göre kök düğümde hangi niteliğin olacağına karar verilir ve dallandırma işlemi yapılır. Aynı işlem tüm yapraklara ulaşıncaya kadar devam ettirilir (Silahtaroglu, 2013).

Çizelge 4.8'deki veri setinde yer alan bir firmaya ait 11 adet müşteri bilgisinden “Gelir”, “Eğitim”, “Sektör” ve “Memnun” nitelikleri bulunmaktadır. Burada “Memnun” niteliği hedef sınıf değeri içermektedir. Twoing algoritması ile karar ağacı oluşturmak için sırası ile tüm değişkenler eşitlik 4.11'e göre hesaplanır.

Çizelge 4.9. C&RT algoritması uygulanacak veriler

Müşteri	Gelir	Eğitim	Sektör	Memnun
1	Normal	Orta	Bilişim	Evet
2	Büyük	İlk	Bilişim	Evet
3	Küçük	İlk	İnşaat	Evet
4	Büyük	Orta	İnşaat	Evet
5	Küçük	Orta	İnşaat	Evet
6	Büyük	Lise	İnşaat	Evet
7	Küçük	Lise	İnşaat	Evet
8	Büyük	Orta	Bilişim	Hayır
9	Küçük	Orta	Bilişim	Hayır
10	Büyük	Lise	Bilişim	Hayır
11	Küçük	Lise	Bilişim	Hayır

Adım 1: Kök düğümün seçilmesi

Algoritmanın uygulanması için niteliklerin her birisi iki ayrı dizi oluşturur. Örneğin “Gelir” düzeyi “Normal” olarak alınırsa bu sol tarafın elemanı olacaktır, sağ tarafı ise “Gelir” düzeyi “Büyük” ve “Küçük” olan elemanlar oluşturacaktır. Sol tarafta bulunanlar t_{sol} , sağ tarafta bulunanlar ise $t_{sağ}$ olarak değerlendirilir. Çizelge 4.9’da aday bölünmeler gösterilmiştir.

Çizelge 4.10. C&RT algoritması için aday bölünmeler

Aday Bölünme	t_{sol}	$t_{sağ}$
1	Gelir=Normal	Gelir = {Büyük, Küçük}
2	Gelir=Büyük	Gelir = {Normal, Küçük}
3	Gelir=Küçük	Gelir = {Büyük, Normal}
4	Eğitim=İlk	Eğitim = {Orta, Lise}
5	Eğitim=Orta	Eğitim = {İlk, Lise}
6	Eğitim=Lise	Eğitim = {İlk, Orta}
7	Sektör=Bilişim	Sektör = {İnşaat}
8	Sektör=İnşaat	Sektör = {Bilişim}

Örnek olarak Çizelge 4.9'daki değerlere göre ağacın sol ve sağ tarafı hesaplanacak olursa; sol taraf için “Gelir” sütununda “Normal” değerlerini alan satırlar, sağ taraf için “Gelir” sütununda “Büyük, Küçük” değerlerini alan satırlar incelenecektir.

Sol taraf için Çizelge 4.8'den görüleceği üzere sadece birinci satırda “Normal” değeri bulunmaktadır, bu satıra karşılık gelen hedef değişken “Evet” değerini almıştır, bu değerlerin “Gelir” durumuna karşılık gelen “Normal” değerini elde etme olasılığı;

$$P_{sol} = \frac{1}{11} = 0.09$$

“Normal” gelir düzeyindeki bireylerin hedef değişken içerisinde “Evet” ve “Hayır” değerini almalarına göre koşullu olasılık değerleri aşağıdaki gibidir;

$$P(\text{Evet} | t_{sol}) = \frac{1}{1} = 1$$

$$P(\text{Hayır} | t_{sol}) = \frac{0}{1} = 0$$

Sağ taraf için “Gelir” durumu “Küçük” ve “Büyük” olan değerler için olasılık değerleri hesaplanır;

$$P_{sağ} = \frac{10}{11} = 0.91$$

“Küçük” ve “Büyük” gelir düzeyindeki bireylerin hedef değişken içerisinde “Evet” ve “Hayır” değerini almalarına göre koşullu olasılık değerleri aşağıdaki gibidir;

$$P(\text{Evet} | t_{sağ}) = \frac{6}{10} = 0.6$$

$$P(\text{Hayır} | t_{sağ}) = \frac{4}{10} = 0.4$$

Bütün satırlar için aynı hesaplamalar yapılır ve daha sonra bu değerler sırası ile eşitlik 4.11'e göre hesaplanır ve Çizelge 4.10'da yer alan $\phi(s|t)$ değerleri bulunur. İlk satır için hesaplama yapılırsa;

$$\phi(s | t) = 2(0.09)(0.91)[|1 - 0.6| + |0 - 0.4|]$$

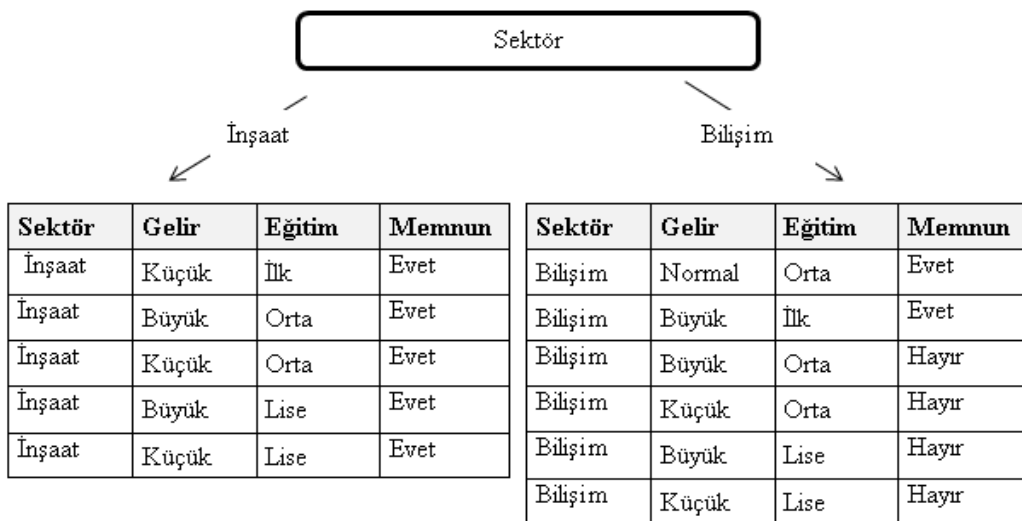
$$\phi(s | t) = 0.13$$

Çizelge 4.9'daki bütün adaylar için algoritma uygulanır ve çizelge 4.10'daki $\phi(s|t)$ değeri hesaplanır.

Çizelge 4.11. Kök düğüm seçimi için $\phi(s|t)$ değeri

Aday Bölünme	P_{sol}	$P_{sağ}$	$2 P_{sol} P_{sağ}$	$\sum P(j t_{sol}) - P(j t_{sağ}) $	$\phi(s t)$
1	0.09	0.91	0.17	0.79	0.13
2	0.45	0.55	0.50	0.14	0.07
3	0.45	0.55	0.50	0.14	0.07
4	0.18	0.82	0.30	0.87	0.26
5	0.45	0.55	0.50	0.14	0.07
6	0.36	0.64	0.46	0.43	0.20
7	0.55	0.45	0.50	1.32	0.66
8	0.45	0.55	0.50	1.32	0.66

Çizelge 4.10'da maksimum $\phi(s|t)$ değerinin alan 7. ve 8. satır üzerinde yer alan 0.66 değeridir. Her ikisi de bölünme için seçilebilir ancak, 8. satır için çizelge 4.9'da "Sektör=İnşaat" değerlerine Çizelge 4.8'deki hedef değışkenden karşılık gelen "Evet" değerleri için dallandırma sonlanır, dolayısıyla kök düğüm seçilmesi gerçekleşir (Şekil 4.7).



Şekil 4.6. C&RT ile kök düğüm seçildikten sonra veri setinin dallanması.

Adım 2: İnşaat sektörüne göre dallandırma

İnşaat sektöründe çalışanlar için, Çizelge 4.11’den görüleceği karşılık hedef değışkende sadece “Evet” değerlerin olduđu için dallandırma işlemi sonlandırılır.

Çizelge 4.12. İnşaat sektörü verileri

Sektör	Gelir	Eğitim	Memnun
İnşaat	Küçük	İlk	Evet
İnşaat	Büyük	Orta	Evet
İnşaat	Küçük	Orta	Evet
İnşaat	Büyük	Lise	Evet
İnşaat	Küçük	Lise	Evet

Adım 3: Bilişim sektörüne göre dallandırma

Bilişim sektöründe çalışanlar için, Çizelge 4.12’de yer alan adaylar için tekrar bir hesaplama gerçekleştirilir.

Çizelge 4.13. Bilişim sektörü

Sektör	Gelir	Eğitim	Memnun
Bilişim	Normal	Orta	Evet
Bilişim	Büyük	İlk	Evet
Bilişim	Büyük	Orta	Hayır
Bilişim	Küçük	Orta	Hayır
Bilişim	Büyük	Lise	Hayır
Bilişim	Küçük	Lise	Hayır

Bilişim sektörü için aday bölünme durumu Çizelge 4.13’de yer almaktadır ve Çizelge 4.14’deki ϕ (s|t) değeri hesaplanır.

Çizelge 4.14. C&RT algoritması için aday bölünmeler

Aday Bölünme	t_{Sol}	$t_{Sağ}$
1	Gelir=Normal	Gelir = {Büyük, Küçük}
2	Gelir=Büyük	Gelir = {Normal, Küçük}
3	Gelir=Küçük	Gelir = {Büyük, Normal}
4	Eğitim=İlk	Eğitim = {Orta, Lise}
5	Eğitim=Orta	Eğitim = {İlk, Lise}
6	Eğitim=Lise	Eğitim = {İlk, Orta}

Çizelge 4.15. Bilişim niteliği için $\phi(s|t)$ değeri

Aday Bölünme	P_{sol}	$P_{sağ}$	$2 P_{sol} P_{sağ}$	$\sum P(j t_{sol}) - P(j t_{sağ}) $	$\phi(s t)$
1	0.17	0.83	0.28	1.57	0.44
2	0.50	0.50	0.50	0.00	0.00
3	0.33	0.67	0.44	1.00	0.44
4	0.17	0.83	0.28	1.57	0.44
5	0.50	0.50	0.50	0.00	0.00
6	0.33	0.67	0.44	1.00	0.44

Çizelge 4.14’de maksimum $\phi(s|t)$ değeri 1., 3., 4. ve 6. satır üzerinde yer alan 0.44 değeridir. En büyük değer olarak herhangi biri seçilebilir ancak Çizelge 4.13’de birinci satırda yer alan gelir düzeyi “Normal” değeri tek olduğu için sol taraf için dallanma durur. Daha sonra gelir düzeyi “Büyük” ve “Küçük” değerleri için dallanma gerçekleşir.

Adım 3: Gelire göre dallandırma

Gelir düzeyi için, Çizelge 4.15’de yer alan adaylar için tekrar bir hesaplama gerçekleştirilir.

Çizelge 4.16. Gelir düzeyi verileri

Sektör	Gelir	Eğitim	Memnun
Bilişim	Büyük	İlk	Evet
Bilişim	Büyük	Orta	Hayır
Bilişim	Küçük	Orta	Hayır
Bilişim	Büyük	Lise	Hayır
Bilişim	Küçük	Lise	Hayır

Gelir düzeyi için aday bölünme durumu Çizelge 4.16’da yer almaktadır ve Çizelge 4.17’deki $\phi(s|t)$ değeri hesaplanır.

Çizelge 4.17. C&RT algoritması için aday bölünmeler

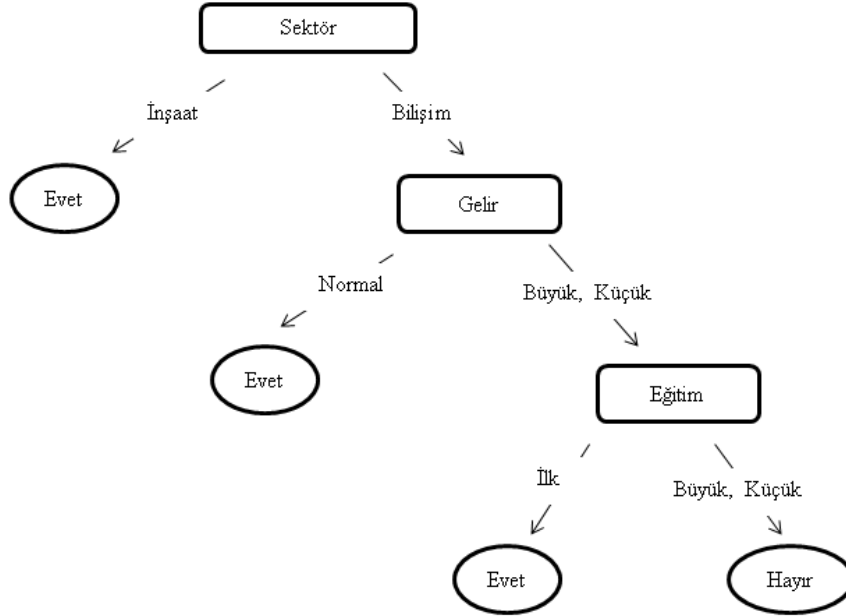
Aday Bölünme	t_{sol}	$t_{sağ}$
1	Gelir=Büyük	Gelir = Küçük
2	Gelir=Küçük	Gelir = Büyük
3	Eğitim=İlk	Eğitim={Orta, Lise}
4	Eğitim=Orta	Eğitim={İlk, Lise}
5	Eğitim=Lise	Eğitim={İlk, Orta}

Çizelge 4.18. Gelir niteliği için $\phi(s|t)$ değeri

Aday Bölünme	P_{sol}	$P_{sağ}$	$2 P_{sol} P_{sağ}$	$\sum P(j t_{sol}) - P(j t_{sağ}) $	$\phi(s t)$
1	0.60	0.40	0.48	0.67	0.32
2	0.40	0.60	0.48	0.67	0.32
3	0.20	0.80	0.32	2.00	0.64
4	0.40	0.60	0.48	0.67	0.32
5	0.40	0.60	0.48	0.67	0.32

Çizelge 4.17’de maksimum $\phi(s|t)$ değeri 3. satırda yer alan 0.64 değeri olduğu gözlenmiştir. Bu değer Çizelge 4.16’da incelendiğinde “Eğitim” niteliğine “İlk” değeri karşılık gelmektedir. Çizelge 4.15’de buna karşılık “Evet” değişkeni gelmektedir. Ayrıca dalın sağ tarafında “Eğitim” niteliğine karşılık gelen “Orta” ve “Yüksek”

değerleri için Şekil 4.15’de karşılık gelen “Hayır” değerleri olduğu için dallandırma işlemi sona erdirilir ve Şekil 4.8’deki gibi karar ağacı oluşturulmuş olur.



Şekil 4.7. C&RT algoritması ile oluşturulan örnek karar ağacı.

4.4.4.2 Gini algoritması

Twoın algoritmasındaki gibi nitelik değerlerinin sol ve sağda olması gibi ikili bölünmeye dayanmaktadır ancak Gni algoritması dallandırma işlemini yaparken Gini ölçütünden yararlanır. Bunun için öncelikle $gini_{sol}$ ve $gini_{sağ}$ değerleri hesaplanır;

$$gini_{sol} = 1 - \sum_{i=1}^k \left(\frac{L_i}{|T_{sol}|} \right)^2 \quad (4.12)$$

$$gini_{sağ} = 1 - \sum_{i=1}^k \left(\frac{L_i}{|T_{sağ}|} \right)^2 \quad (4.13)$$

k : sınıf sayısı

T : düğümdeki örnekler

L_i : sol tarafta i kategorisindeki örneklerin sayısı

R_i : sağ tarafta i kategorisindeki örneklerin sayısı

$|T_{sol}|$: Sol taraftaki örneklerin sayısı

$|T_{sağ}|$: Sağ taraftaki örneklerin sayısı

Daha sonra her j niteliği için, n kümesindeki satır sayısı olmak üzere Gini ölçütü aşağıdaki gibi elde edilir;

$$gini_j = \frac{1}{n} \left(|T_{sol}| gini_{sol} + |T_{sağ}| gini_{sağ} \right) \quad (4.14)$$

Böylece her nitelik için $gini_j$ ölçütü hesaplanır ve aralarından en küçük olanı seçilir bu sayede kök düğüm belirlenmiş olur. Bölünme bu niteliğe göre gerçekleşir ve yapraklara ulaşıncaya kadar bu işlem devam ettirilir (Özkan, 2008).

4.4.5 CHAID algoritması

G.V. Kass tarafından 1980 yılında geliştirilen CHAID (Chi-squared Automatic Interaction Detector) algoritmasının dilimizdeki karşılığı “Otomatik ki-kare Etkileşim Keşfedici”dir. İstatistik tabanlı olan bu algoritma adından anlaşılacağı üzere ki-kare test yöntemini kullanarak karar ağacı oluşturur. Ancak verileri daha alt homojen alt gruplara ayırırken kullanılacak olan istatistiksel test hedef değişkenin ölçüm düzeyine göre değişmektedir. Hedef değişken ya da bağımlı değişken de denilebilir sürekli ise F testi, kategorik ise ki-kare testi kullanılmaktadır. Buradan da anlaşılacağı üzere CHAID algoritması C&RT algoritması gibi hem sayısal hem de kategorik verilerle çalışabilmektedir (Rokack ve Maimom, 2008)

CHAID algoritması ile karar ağaçları oluşturulurken ikiden fazla dallandırma yapabilir, yani nitelik kategorisine göre dal üretebilir. Alt gruplarda yer alma ihtimali olan bütün değerleri ağaç yapısında göstermektedir (Koyuncugil, 2006)

CHAID algoritması adımları aşağıda yer almaktadır;

1. Veri setindeki her bir bağımsız değişkenin kategorileri ile bağımlı değişkenin kategorileri arasında çapraz tablo oluşturulur. Bunların arasında en az öneme sahip

kategori çiftleri bulunur (en büyük p değerine sahip olan). Burada bağımlı değişkenin türüne göre p değeri hesaplanacaktır;

- a. Bağımlı değişken sürekli ise F testi ile
- b. Bağımlı değişken kategorik ise Ki-kare testi ile hesaplanır.

2. En büyük p değerine sahip bağımsız kategori çifti için, daha önceden belirlenmiş alfa düzeyi ile p değeri kıyaslanır;
 - a. p değeri alfadan büyük ise bu iki kategori birleştirilir ve bir önceki adıma geri dönülür,
 - b. p değeri alfadan küçük ise sıradaki adıma geçilir.
3. Bağımlı ve bağımsız kategori kümeleri için uygun Bonferroni düzeltmesi kullanılarak, düzeltilmiş p değeri hesaplanır.
4. En önemli bağımsız değişken, en küçük düzeltilmiş p değerine sahip olan arasından seçilir ve bu değer önceden tanımlanmış alfa düzeyi ile kıyaslanır;
 - a. p değeri alfa değerinden küçük yada eşit ise düğüm belirlenmiş olan bağımsız kategori kümesi temel alınarak bölünür,
 - b. p değeri alfa değerinden büyük ise bölünme olmaz, bu düğüm uç düğüm olur.
5. Bu işlem durma kuralları gerçekleşinceye kadar devam eder.

CHAID algoritmasını bir örnekle açıklayacak olursak;

60 kişilik bir örneklem grubunun “Sağlık”, “Gelir” ve “Umut” bağımsız değişkenleri ile “Mutluluk” hedef değişkeni arasındaki ilişkiyi CHAID algoritması ile oluşturulan karar ağacı ile göstermeye çalışalım. Çizelge 4.18’de bağımsız ve bağımlı değişkenlerin kategorileri yer almaktadır.

Çizelge 4.19. CHAID nitelikleri

Değişkenler	Kategorileri
Mutluluk	Mutlu, Orta, Mutsuz
Sağlık	Sağlıklı, Sağlıksız
Gelir	İyi, Kötü
Umut	Umutlu, Umutsuz

Kök Düğümün Seçilmesi

Sırası ile bağımsız değişkenlerin kategorileri ile bağımlı değişkenin kategorileri arasında çapraz tablolar oluşturulur ve ki-kare değerleri hesaplanır ve en büyük ki-kare (ya da en küçük p değeri) değerine sahip değişken kök düğüm olarak seçilir.

a) Sağlık değeri için çapraz tablo oluşturulması:

Sağlık niteliğinin “Sağlıklı” ve “Sağlıksız” değerleri ile bağımlı değişkenin “Mutlu”, “Orta” ve “Mutsuz” değerleri arasında çapraz tablo oluşturulur ve gözlenen frekans değerleri tablo içeresine yerleştirilir, beklenen değerlerde eşitlik 4.15’e göre hesaplanarak parantez içeresinde gösterilmiştir.

$$B_{ij} = \frac{(T_j)(T_i)}{n} \quad (4.15)$$

Burada; T_i = i.satır toplamı, T_j = j.satır toplamı n ise genel toplamı ifade etmektedir.

Daha sonra eşitlik 4.16’ay göre değişkenlerin ki-kare değerleri hesaplanır.

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(G_{ij} - B_{ij})^2}{B_{ij}} \quad (4.16)$$

G_{ij} = i. satır ve j. sütundaki gözlenen değeri,

B_{ij} = i. satır ve j. sütundaki beklenen değeri ifade etmektedir.

Çizelge 4.20. Sağlık ve mutluluk değerleri için çapraz tablo

Sağlık	Mutluluk			Toplam
	Mutlu	Orta	Mutsuz	
Sağlıklı	16 (11.7)	12 (11.2)	4 (9.1)	32
Sağlıksız	6 (10.3)	9 (9.8)	13 (7.9)	28
Toplam	22	21	17	60

“Sağlık“ bağımsız değişkeni ile “Mutluluk” bağımlı değişkeni için eşitlik 4.16’ya göre ki-kare değeri hesaplanır.

$$\begin{aligned}
&= \frac{(16-11.7)^2}{11.7} + \frac{(12-11.2)^2}{11.2} + \frac{(4-9.1)^2}{9.1} + \frac{(6-10.3)^2}{10.3} + \frac{(9-9.8)^2}{9.8} + \frac{(13-7.9)^2}{7.9} \\
&= (1.55) + (0.06) + (2.83) + (1.77) + (0.07) + (3.24) \\
&= 9.51
\end{aligned}$$

b) Gelir düzeyleri için çapraz tablo oluşturulması:

Gelir niteliğinin “İyi” ve “Kötü” değerleri ile bağımlı değişkenin “Mutlu”, “Orta” ve “Mutsuz” değerleri arasında çapraz tablo oluşturulur.

Çizelge 4.21. Gelir ve mutluluk değerleri için çapraz tablo

Gelir	Mutluluk			Toplam
	Mutlu	Orta	Mutsuz	
İyi	20 (16.0)	12 (14.7)	8 (9.3)	40
Kötü	4 (8.0)	10 (7.3)	6 (4.7)	20
Toplam	24	22	14	60

“Gelir” bağımsız değişkeni ile “Mutluluk” bağımlı değişkeni için eşitlik 4.9’a göre ki-kare değeri hesaplanır.

$$\begin{aligned}
&= \frac{(20-16.0)^2}{16.0} + \frac{(12-14.7)^2}{14.7} + \frac{(8-9.3)^2}{9.3} + \frac{(4-8.0)^2}{8.0} + \frac{(10-7.3)^2}{7.3} + \frac{(6-4.7)^2}{4.7} \\
&= (1.00) + (0.48) + (0.19) + (2.00) + (0.97) + (0.38) \\
&= 5,03
\end{aligned}$$

c) Umut değerleri için çapraz tablo oluşturulması:

Umut niteliğinin “Umutlu” ve “Umutsuz” değerleri ile bağımlı değişkenin “Mutlu”, “Orta” ve “Mutsuz” değerleri arasında çapraz tablo oluşturulur.

Çizelge 4.22. Umut ve mutluluk değerleri için çapraz tablo

Umut	Mutluluk			Toplam
	Mutlu	Orta	Mutsuz	
Umutlu	14 (13.0)	17 (18.5)	6 (5.6)	37
Umutsuz	7 (8.1)	13 (11.5)	3 (3.5)	23
Toplam	21	30	9	60

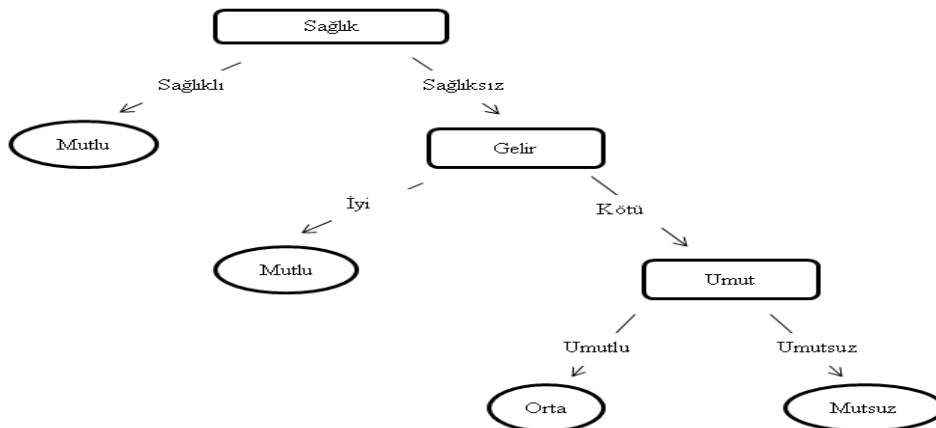
“Umut” bağımsız değişkeni ile “Mutluluk” bağımlı değişkeni için eşitlik 4.16’a göre ki-kare değeri hesaplanır.

$$\begin{aligned}
 &= \frac{(14-13.0)^2}{13.0} + \frac{(17-18.5)^2}{18.5} + \frac{(6-5.6)^2}{5.6} + \frac{(7-8.1)^2}{8.1} + \frac{(13-11.5)^2}{11.5} + \frac{(3-3.5)^2}{3.5} \\
 &= (0.09) + (0.12) + (0.04) + (0.14) + (0.20) + (0.06) \\
 &= 0.63
 \end{aligned}$$

Yapılan hesaplamalar neticesinde üç ayrı nitelik için hesaplanan ki-kare değerleri;

- “Sağlık” değişkeni için ki-kare değeri 9.03 (p=0.01)
- “Gelir” değişkeni için ki-kare değeri 5.03 (p=0.08)
- “Umut” değişkeni için ki-kare değeri 0.63 (p=0.73) olarak hesaplanmıştır.

Bu üç nitelikten kök düğüm için en fazla ki-kare değeri 9.03 (ya da en küçük p değeri) ile “Sağlık” değeri ortaya çıkmıştır. Bu durumda kök düğümüne “Sağlık” değişkeni yerleştirilir ve “Sağlıklı” ile “Sağlıksız” değişkenleri için dallandırma işlemi gerçekleştirir. Daha sonra bu işlem sırası ile devam ettirilir ve şekil 4.9’deki gibi karar ağacı oluşturulur.



Şekil 4.8 CHAID algoritması ile oluşturulan örnek karar ağacı.

4.4.6 QUEST algoritması

Wei-Yin Loh ve Yu-Shan Shih tarafından 1997 yılında geliştirilen QUEST (Quick Unbiased Efficient Statistical Tree) algoritmasını “Hızlı, Yansız, Etkili İstatistiksel Ağaç” olarak ifade edebiliriz. FACT algoritmasından esinlenerek geliştirilmiştir (Shih ve Lon, 1997). C&RT algoritmasındaki gibi ikili karar ağaçları oluşturan bir yapısı vardır. Budama ve doğrudan durma kuralı gibi tekniklerin kullanılabilmesinden dolayı ikili ağaç yapısını kullanmaktadır. QUEST algoritması, ağaç oluştururken değişken seçimi ve dallandırma işlemini aynı anda yapan CHAID ve C&RT algoritmasından farklı olarak hepsi ile tek tek ilgilenir. QUEST algoritması, ağacın bölünmesi sırasındaki önyargılı seçimin daha genel davranması ve hesaplama maliyetinin azaltılması amacıyla geliştirilmiştir (Çalış ve ark, 2014). Bağımsız değişkenler sürekli ya da kategorik olabilir ancak bağımlı değişken yani hedef değişken kategorik olmalıdır (Köktürk, 2012) Bağımsız değişken seçimi ve bölme işlemlerinde ilgilenilen değişken kategorik ise ki-kare testi kullanılır. Eğer ilgilenilen değişken sürekli ise F testi ile hesaplama yapılır (Rokack ve Maimom, 2008)

QUEST algoritmasının adımları aşağıda yer almaktadır;

Bağımsız değişken seçimi;

1. Her bir belirleyici bağımsız değişken (X) için, eğer bağımsız değişken kategorik alan ise, nominal ya da sıralı, bağımsız değişken ve bağımlı değişken arasındaki bağımsızlığın bir pearson ki-kare testinin p değeri hesaplanır. Eğer bağımsız değişken sürekli ise bu durumda F testi ile p değeri bulunur.
2. En küçük p değeri önceden belirlenen bonferroni düzeltilmiş alpha değeri (bonferroni-adjusted alpha level) (α_B) ile kıyaslanır;
 - a. Hesaplanan en küçük p değeri α_B ’den küçük ise, karşılık gelen ilgili değişken düğümü bölmek için seçilir.
 - b. Hesaplanan en küçük p değeri α_B ’den büyük ise, sürekli olan her bir X için bir p değeri hesaplanır, eşit olmayan varyanslar için Levene’s testi kullanılır.

- c. Levene's testinden elde edilen küçük p değeri yeni bir Bonferroni-düzeltilmiş alpha düzeyi α_L ile kıyaslanır.
- d. Eğer p değeri α_L 'den küçük ise düğümü bölmek için Levene's testinden en küçük p değerli olan ilgili alan seçilir.
- e. Eğer p değeri α_L 'den büyük ise düğüm bölünmez.

Bölünme Noktası Seçim Adımları;

1. Eğer bağımlı değişken (Y) sadece iki kategoriye sahip ise, bir sonraki adıma geçer. değilse Y'nin kategorilerini iki üst sınıfa aşağıdaki gibi ayırır.
 - a. Y'nin her bir kategorisi için X'in ortalaması hesaplanır.
 - b. Eğer tüm ortalamalar aynı ise, en büyük ağırlıklı frekans kategori bir üst sınıf olarak seçilir ve tüm diğer kategoriler öteki üst sınıf oluşturmak için birleştirilir.
 - c. Eğer ortalamalar tamamen aynı değil ise, başlangıç küme merkezleri iki en uç sınıf ortalaması olarak ayarlanması ile, k- ortalamalar kümesinde k=2 olarak alınarak, iki-ortalamalar kümesi algoritması Y'nin iki üst sınıfını elde etmek için uygulanır.
2. Bölme noktasının belirlenmesinde Kuadratik Diskriminant Analizi (QDA) uygulanır. QDA genel olarak iki kesme noktası üretir, birinci üst sınıfın örnek ortalamasına en yakın olanı seçilir.

QUEST ilk önce, kategorik alanı sürekli alana dönüştürür. Diskriminant koordinatlarını kategorilere göre atayarak dönüştürme işlemini gerçekleştirir daha sonra yukarıdaki adımlara göre bölme işlemini gerçekleştirir (Kuzey, 2012)

Bu bölümde detaylı olarak açıklanan CHAID, C&RT, QUEST ve C5.0 karar ağacı algoritmaları Çizelge 4.22'de bazı kriterler bakımından karşılaştırılmıştır. Ayrıca bu tez çalışmasında uygulama C&RT ve CHAID algoritmaları ile yapıldığı için bu iki algoritma kıyaslandığında;

Shmueli'ye (2007) göre, CHAID mevcut durumu analiz etmede daha kullanışlı iken C&RT ise gelecek ile tahminleme yapmak için daha uygundur. Diğer bir deyişle, amaç bir dizi açıklayıcı değişkenler ile hedef değişken arasındaki ilişkinin tanımlanması

olduğunda CHAID daha kullanışlı iken; yeni sınıfların yüksek doğruluk oranı ile tahmin edilmesi için model oluşturulmasında C&RT'nin kullanılması daha uygundur.

Çizelge 4.23. Karar ağacı algoritmalarının karşılaştırılması.

	CHAID	C&RT	QUEST	C5.0
Dallanma kriteri	Kategorik ki-kare Sürekli ANOVA	Gini Twin	Kategorik ki-kare Sürekli ANOVA	Bilgi kazancı (Entropi)
Bölünme Yapısı	2'den fazla	2'li bölünme	2'li bölünme	2'den fazla
Hedef Değişken	Sürekli & Kategorik	Sürekli & Kategorik	Kategorik	Kategorik
Tahmin Edici Değişken	Sürekli & Kategorik	Sürekli & Kategorik	Sürekli & Kategorik	Sürekli & Kategorik
Kayıp Değişkenler	Kayıp gözlemler için kategori oluşturur ağaçta kullanır.	Vekil değişken ile kayıp gözlemleri değerlendirir	Vekil değişken ile kayıp gözlemleri değerlendirir	Kayıp gözlemleri bir noda düşürür.

5. UYGULAMA

Bu bölümde veri madenciliği yöntemlerinden olan ve detaylıca bahsedilen karar ağacı algoritmaları ile uygulama yapılmıştır. Uygulamada kullanılan veri seti Türkiye İstatistik Kurumu (TÜİK) tarafından yayımlanan Yaşam Memnuniyeti Araştırmalarından (YMA) elde edilen B grubu mikro verilerden oluşmaktadır. Uygulamanın yapılması amacı ile kullanılan program IBM SPSS Modeler 14.2'dir. Uygulamanın yapılışından önce IBM SPSS Modeler ve Yaşam Memnuniyeti Araştırması hakkında kısa bilgiler verilmiştir.

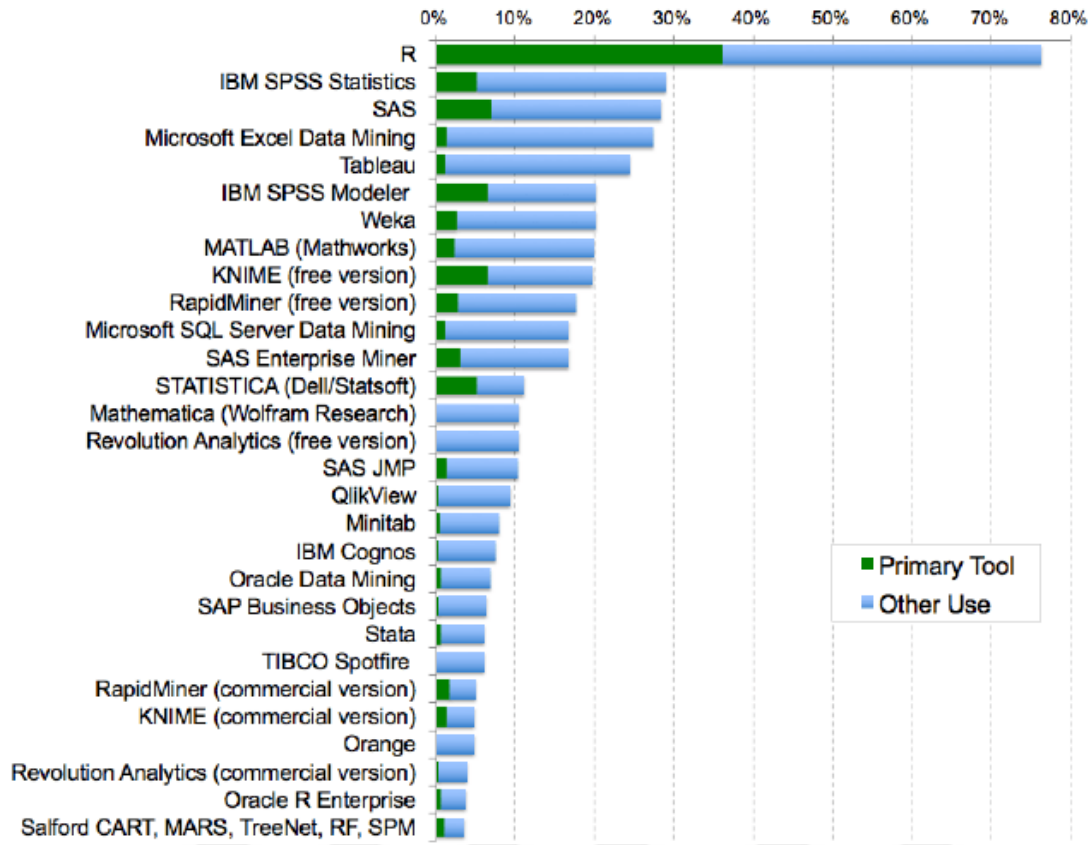
5.1 IBM SPSS MODELER 14.2

Veri madenciliği için geliştirilmiş birçok program/yazılım bulunmaktadır. Bu çalışmada; yaygın olarak kullanılan, programlama bilgisine gerek olmadan, kolay anlaşılabilir ara yüze sahip olan, hızlı bir şekilde modellemeleri yapabilen ve görsel olarak incelenmesinin rahat olması sebebiyle IBM SPSS Modeler programı kullanılmıştır.

Daha önceleri SPSS Clementine olarak piyasada var olan program 2009 yılında IBM firmasının SPSS'i satın almasından sonra birçok değişikliğe giderek programın yeni versiyonlarını piyasaya sürülmektedir (Soner, 2015).

IBM SPSS Modeler Bölüm 2.3'te bahsedildiği üzere veri madenciliği süreci olarak nitelendirilen CRISP-DM metodolojisi üzerine kurulmuştur.

Rexer Analytics tarafından yürütülen ve 1.220 Veri Bilimcinin (Data Scientist) katıldığı bir araştırmada kullanılan veri madenciliği programına ait grafik Şekil 5.1'de verilmiştir. Grafikten de anlaşılacağı üzere açık kaynak kodlu olarak kullanılan R programı en çok tercih edilen programdır. IBM SPSS Modeler yaklaşık %20'lik kullanım oranıyla 6. sırada gelmektedir (Muenchen, 2015).



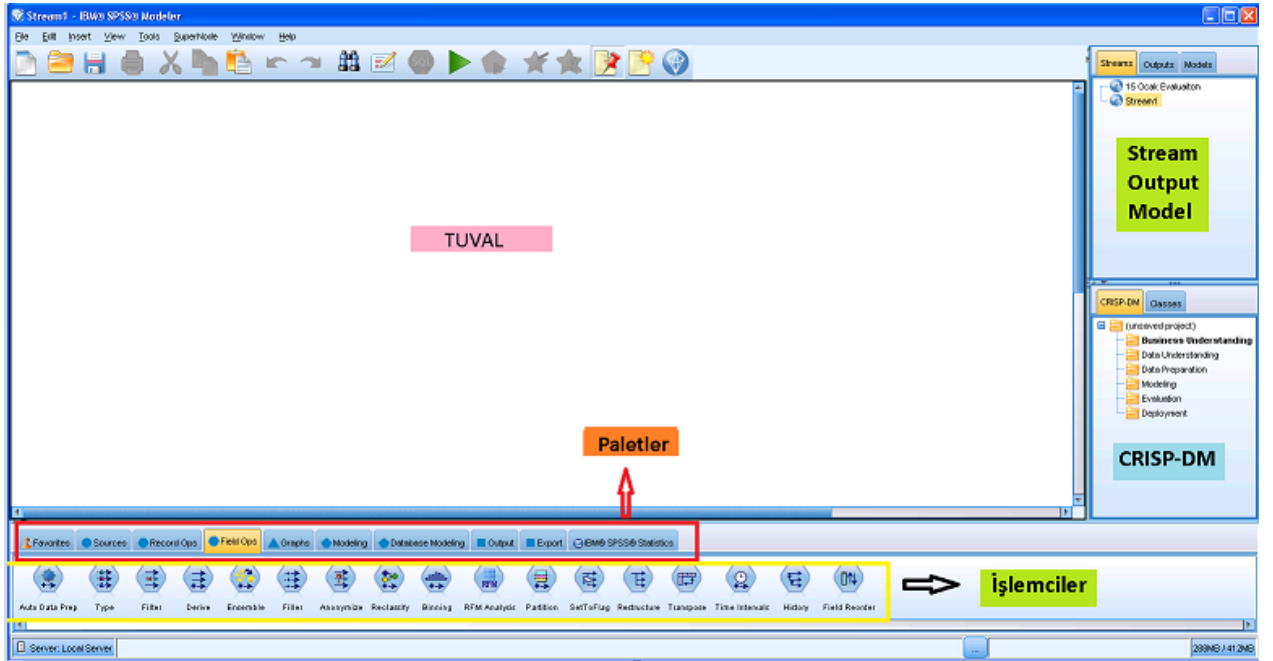
Şekil 5.1. Tercih edilen veri madenciliği programları (Muenchen, 2015).

5.1.1 IBM SPSS Modeler kullanımı

IBM SPSS Modeler çalıştırıldığında Şekil 5.2'deki arayüz ekrana gelir. Modeler veri madenciliği işlemlerinin görsel programlama ile yapılmasına izin verir. Boş alan tuval olarak adlandırılır ve burada işlemciler (nodes) birbirlerine bağlanarak akış diyagramı (stream) oluşturulur ve çalışmalar bu tuval üzerinden yürütülür.

IBM SPSS Modeler penceresinin alt bölümünde paletler bulunur ve her palet veri akışında kullanılabilecek, yaptıkları görevlere göre gruplanmış işlemcileri barındırır. Bu paletlerin tanımlarına ve altında yer alan bazı önemli işlemcilere kısaca bakacak olursak (TUİK, 2011; SPSS,2012);

- **Favorites:** Sık olarak kullanılan işlemciler bulunur.



Şekil 5.2 IBM SPSS modeller arayüzü.

- **Sources:** Verilerin bulundukları kaynaktan okunmasını sağlayan işlemcileri içerir. SAS, excel dosyası gibi.



Var File: Değişkenleri herhangi bir ayraçla ayrılmış olan .csv uzantılı metin dosyalarını okunması için kullanılır. Nokta, virgül, noktalı virgül vb.



Databases: ODBC protokolünü destekleyen veritabanlarından verilerin okunmasını sağlar. Bu işlemci ile veritabanı tabloları ya da SQL sorguları ile verilere erişim sağlanabilir. Ancak verilerin databases işlemcisi ile okunmasında önce bilgisayara ODBC sürücülerini kurulmuş olmalı ve veri kaynağı ilgili işletim sisteminde tanımlanmış olmalı.



Excel: Excel dosyalarının okunmasını sağlar.

- **Record Ops:** Seçme, birleştirme, sıralama gibi işlemlerin yapılmasını sağlayan işlemcileri içerir.



Merge: İki farklı veri kaynağı anahtar değişken ile birleştirmek için kullanılır. Dört farklı birleştirme seçeneği bulunmaktadır.

- **Inner join:** Sadece eşleşen kayıtları gösterir.
- **Full outer join:** Her iki tabloda olan tüm kayıtları gösterir.
- **Partial-outer join:** İlk tabloda olan tüm kayıtlar gelir.
- **Anti-join:** Eşleşmeyen kayıtlar üzerinde birleştirme işlemleri yapılır.



Aggregate: İstenilen değişkenlere göre gruplama işlemi yapılabilir ve oluşturulan grupların temel istatistik ölçümleri alınabilir.



Sort: Kayıtlar istenen değişkene/değişkenlere göre artan ya da azalan sırada sıralanabilir.



Append: Veri kümeleri alt alta birleştirmek için kullanılır.

- **Field Ops:** Değişkenler üzerinde değişken tipi belirleme, filtreleme, yeni değişken türetme gibi işlemlerin yapılmasını sağlar.



Type: En önemli işlemcilerden birisi olup değişkenlerin tiplerinin belirlenmesinde kullanılır. Modellerde tanımlı olan değişken tipleri;

- **Continuous:** Sürekli değişkenlerin tanımlanmasında kullanılır.
- **Categorical:** Kategorisi sayısı belirsiz olan metin tipindeki değişkenlerin tanımlanmasında kullanılır.
- **Flag:** Kategori sayısı iki olan değişkenler için kullanılır (Evet, Hayır)

- **Nominal:** Üç ya da daha fazla kategori sayısı olan ve aralarında sıra ilişkisi olmayan değişkenler için kullanılır (Evli, Bekar, Boşanmış).
- **Ordinal:** Üç ya da daha fazla kategori sayısı olan ve aralarında sıra ilişkisi (Hiyerarşi) olan değişkenlerin tanımlanmasında kullanılır (Küçük, Orta, Büyük).
- **Typless:** Yukarıdaki veri tiplerine uymayan ve çok sayıda değer içeren veriler için kullanılır.

Type işlemci bağlandıktan sonra verilerin tipleri belirlenirken aynı zamanda verilerin analizde yer alıp almayacağıda burada belirlenebilmektedir. Tipler belirlendikten sonra “Role” kısmında verinin modelde nasıl davranacağı seçilir. Bunlar;

- **Input (Bağımsız değişken):** Modelleme çalışmasında değişkenler girdi ya da tahmin edici olarak belirlenir.
- **Target (Bağımlı değişken):** Çalışmada değişkenlerin çıktı, ya da tahmin edici olarak belirlenir.
- **Both:** Değişkenin hem girdi hem çıktı olarak kullanılmasını sağlar. Ancak sadece “Apriori” algoritmasında kullanılır.
- **None:** modelleme çalışmasında kullanılmayacak olan değişkenlerin belirlenmesi sağlar.
- **Partition:** Değişkenlere ait veileri rastgele alt örneklemelere ayırır ve bunların testing, training, validation olarak kullanılmasını sağlar.
- **Split:** Sadece nominal, ordinal ve flag tipindeki değişkenler için, bir modelin ilgili değişkenin tüm kategorileri için ayrı yürütülmesini sağlar.
- **Frequency:** C&RT, CHAID, QUEST ve Lineer modelleme işlemcilerinde her kayıt için frekans ağırlık faktörü kullanılması sağlar. Diğer modelleme çalışmalarında bu değişken tipindeki veiler kullanılmamaktadır.



Filter: İstenmeyen değişkenlerin analiz dışı bırakılmasını sağlar.



Derive: Yeni değişken üretilmesini sağlar.



Ensemble: Modelleme sonucu elde edilen tüm modeller ensemble işlemcisi ile birleştirilerek en doğru tahmin elde edilmeye çalışılır.



Reclassify: Kategorik değişkenlerin yeniden düzenlenmesine yardımcı olur. Örneğin veri setinde 1 ve 2 olan değerler 1=Erkek ve 2=Kadın şeklinde düzenlenebilir.



Partition: Veri setini training ve testing olarak bölmemize yarayan (validation olarak ta bölme imkanı verir) işlemci, modeli training ile geliştirir testing ile test eder. Varsayılan olarak %50 testing ve %50 training gelir ancak uygulamada isteğe göre değiştirilir.

- **Graphs:** verileri görel olarak incelememizi sağlayan grafik türlerini içerir.



Graphboard: Değişkenin ölçüm tipine göre otomatik olarak hangi alternatif grafiklerin çizilebileceğinin gösterir.



Web: İki yada daha fazla kategorik değişken arasındaki bağlantının gücünü gösterir. İncelemeye alınan değişkenlerin kategorileri birer nokta ile ifade edilir ve hepsi düz bağlantı çizgileriyle birbirlerine bağlanır.



Evalation: Farklı modellerin performansını karşılaştırılamsına olanak sağlar.



Distribution: Değişkenlerin dağılımı elde edilir.

- **Modelling:** Verilerin modellemesinde kullanılacak olan algoritmaları içerir.



C&RT: İkili dallar üreten C&RT algoritmasına ait işlemcidir, bağımlı ve bağımsız değişkenler sayısal ya da kategorik olabilir.



QUEST: İkili dallar üretir, bağımsız değişkenler sürekli olabilir ancak bağımlı değişken kategorik olmak zorundadır.



CHAID: İki'den fazla bölünme gerçekleştirilebilir, bağımlı ve bağımsız değişkenler sayısal ya da kategorik olabilir.



C5.0: İki'den fazla bölünme gerçekleştirilebilir, bağımlı değişken kategorik olmalıdır.



Feature Selection: Çok sayıda değişken arasından hangilerinin bağımlı değişkenin tahmini üzerinde önemli etkiye sahip olduğunu belirleyerek analiz işleminde gereksiz değişkenlerin elenmesi için kullanılır.



Auto Classifier: Sınıflama ölçme düzeyine sahip bağımlı değişken için birden çok modeli tahmin eder ve karşılaştırır. Bağımsız değişkenler sürekli ya da kategorik olabilir. Bu işlemci karar ağacı yukarıda bahsedilen algoritmalarını destekler.



Auto Numeric: Bağımlı değişken sürekli olduğunda birden fazla modeli karşılaştırmak için kullanılır.



Auto Cluster: Benzer özelliklere sahip olan kayıt gruplarını tanımlar, kümeleme modellerini tahmin eder ve bu modelleri birbirleri ile karşılaştırır.

- **Output:** Modelleme işlemi sonuçlarını görebilmemizi sağlayan ve verileri gözlemleyebileceğimiz işlemcileri içerir.



Table: Yapılan işlemlerin tablo şeklinde görüntülenmesini sağlar.



Matrix: İki kategorik değişken arasındaki ilişkinin incelenmesini sağlar ve çapraz tablo şeklinde gösterir.



Analysis: Bu işlemci üretilen modelin hem testing hem de trainig alt kümelerinde gerçek değerleri ne kadar iyi tahmin ettiğini değerlendirmektedir. Aynı zamanda farklı modellerin elimizdeki model ile ne kadar örtüştüğünü göstermektedir.



Data Audit: Değişkenlere ait istatistiksel ölçümleri verir (min-max, ortalama, standart sapma, varyans, dağılımlar vb.). Kayıp ve aykırı değerlerin tespitinde de kullanılmaktadır.



Statistics: Korelasyon katsayısı ile sayısal değerler arasındaki ilişkinin incelenmesini sağlar.



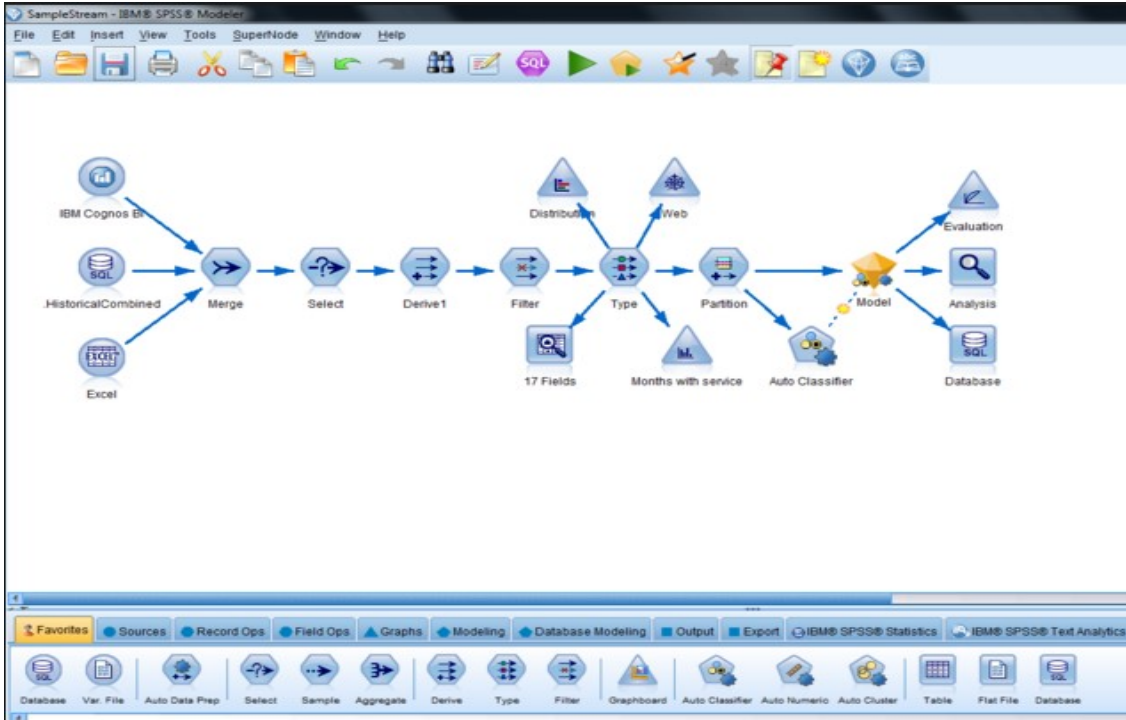
Means: Bağımsız gruplar ya da bağımlı değişken çiftleri arasında farklılık olup olmadığı incelenir.

- **Export:** Çıktı tablolarının farklı formatta kaydedilmesini, görüntülenmesini ve dışa aktarılmasını sağlar. Excel, SAS vb.

IBM SPSS Modeler'ın ana penceresinde sağ üst köşede Modeler'a ait nesnelerinin yönetildiği "Streams", "Outputs" ve "Models" adında üç tane yönetici sekmesi bulunmaktadır. Oluşturulan akış şemaları, çıktılar ve modeller bu bölümlerde gösterilmektedir.

Modeler'ın ana ekran sağ alt köşesinde ise proje penceresi yer almaktadır. Burada yürütülen projeye ait dokümanların organizasyonu sağlanır. CRISP-DM sekmesinde çalışan projeye ait dokümanlar bu süreç çerçevesinde gruplanabilir. Bu sekmedeki klasörlerin altına sadece Modeler'a ait nesneler konulmayabilir. Word'de yazılmış bir not veya sunum dosyaları da bu sekmelere eklenebilir.

Şekil 5.3'te IBM SPSS Modeler'da veri madenciliği süresince yapılan stream (iş akış şeması) örneği gösterilmiştir.



Şekil 5.3. IBM SPSS modeller’da stream örneği.

5.2 Yaşam Memnuniyeti Araştırması

Mutluluk ve yaşam memnuniyeti kavramları, dünyada çok eski zamanlardan bu yana üzerinde düşünülen ve felsefe alanında türlü boyutlarıyla irdelenen bir konudur. Mutluluk kavramı, TDK’nın web sitesinde “bütün özlemlere eksiksiz ve sürekli olarak ulaşmaktan duyulan kıvanç durumu” olarak tanımlanmaktadır (TDK, 2015). Derin ve soyut yapısından dolayı mutluluk farklı yorumlamalara sahiptir. Yıllar boyunca mutluluk insanların maddi veya manevi değerleri ile birlikte bireyin içsel beğenisi ile toplumun beğenisini üzerine çekebilecek donanımlara sahip olmasına ve bu değerlerin birleşimi ile sürdürdüğü hayatından memnun olması durumuna göre birçok yönüyle ele alınmıştır. Artık mutluluk kavramı, bireyin kendini iç değerlendirmeleri sonucunda bir çıktı olarak kendisini mutlu hissedip hissetmemesi olarak değerlendirilmektedir (TÜİK, 2013). Yaşam memnuniyeti kavramı ise, bireyin sürdürmekte olduğum yaşamdan ne kadar zevk, haz ya da hoşnutluk aldığını bir bütün olarak ele alması sonucunda elde ettiği toplam olumluluk derecesini ifade etmektedir (Güler ve Emeç, 2006).

Saha çalışmalarından elde edilen veriler aracılığı ile mutluluk düzeyinin ölçümü, dünyada 1940’ların ortasından itibaren başlamıştır. “World Data Base of Happiness”

veri tabanı mutluluk kavramı alanında yapılan çalışmalar hakkında zengin içeriğe sahiptir. Bunların yanında yaşam memnuniyeti konusunda “Journal of Happiness Studies” gibi süreli yayınlar bulunmaktadır. Türkiye’de resmi istatistik olarak mutluluk konusunda yapılan ilk araştırma, Türkiye istatistik Kurumu (TÜİK) tarafından “Yaşam Memnuniyeti Araştırması” (YMA) adıyla 2003 yılında gerçekleştirilmiştir.

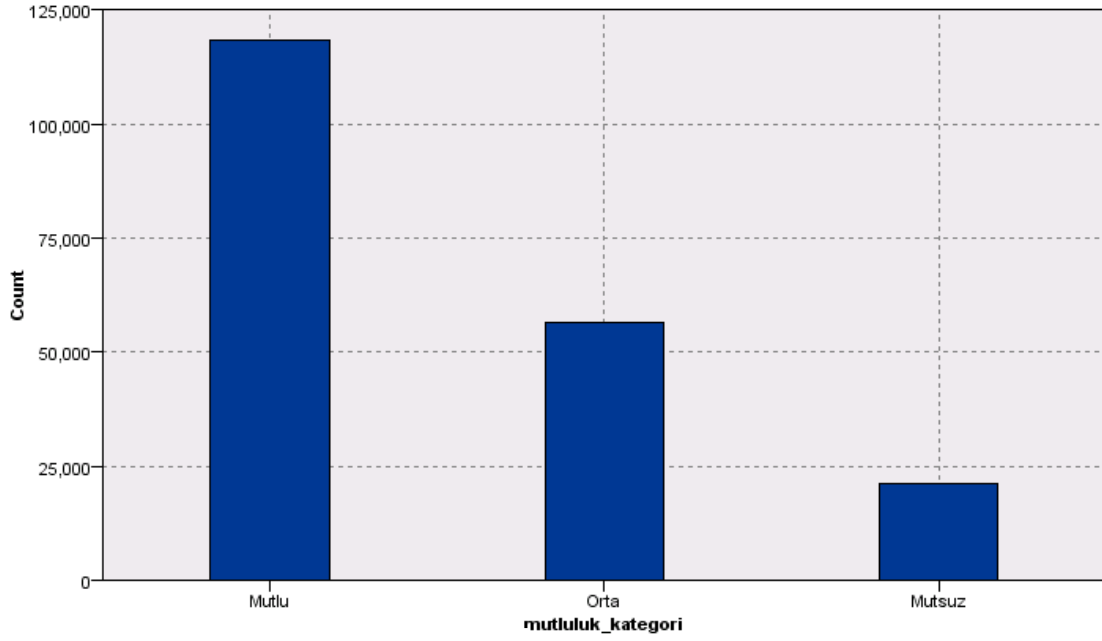
TÜİK’ in toplumsal içerikli araştırması olan Yaşam Memnuniyeti Araştırması, bunun yanında öznel öğeler içeren ilk araştırmadır. Araştırmada bireylerin mutluluk, umut gibi temel yaşam alanlarındaki genel memnuniyeti ile bu alanlardaki kamu hizmetlerinden memnuniyeti ölçülmüştür (TÜİK, 2013).

2013 yılında ilk defa il düzeyinde tahmin üretmek amacıyla örneklem büyüklüğü hesaplanmıştır. 2013 yılında Türkiye genelinde örneğe çıkan 125 720 haneden 103 312 hanedeki 18 ve daha yukarı yaştaki 196 203 birey ile yüz yüze görüşme yöntemiyle veri toplanmış ve il bazında sunulmuştur.

5.3 Uygulamanın Gerçekleştirilmesi

2013 yılında gerçekleştirilen yaşam memnuniyeti araştırmasından elde edilen mikro veriler, IBM SPSS Modeler programı ile karar ağacı algoritmaları uygulanarak incelenmiştir. Veri tanımlama işlemleri yapıldıktan sonra Auto classifier işlemci veri setine bağlanmış ve elimizdeki veri yapısına uygulanabilecek karar ağacı algoritmaları belirlenmiştir. Yapılan bu incelemeler sonucunda CHAID ve C&RT algoritması için karar ağacı oluşturulmuş ve bu algoritmalar şekiller ile gösterilmiş olup yorumlanmıştır.

Uygulamada anket soru formunda yer alan “Yaşamınızı bir bütün olarak düşündüğünüzde ne kadar mutlusunuz?” sorusu hedef değişkeni yani bağımlı değişken olarak belirlenmiştir. Bu soruya verilen “Çok mutlu, mutlu, orta, mutsuz ve çok mutsuz” cevapları hedef sınıfları oluşturmaktadır. Mutluluk düzeyi reclassify işlemcisi ile; “çok mutlu ve mutlu” olanlar “mutlu”; “Çok mutsuz ve mutsuz” olanlar “mutsuz” olacak şekilde tekrardan kodlanmıştır (Şekil 5.5). Şekil 5.4’de genel mutluluk düzeyine ait grafik verilmiştir.



Şekil 5.4. Genel mutluluk düzeyi.

mutluluk_kategori

Preview

Settings Annotations

Mode: ☒ Single ☐ Multiple

Reclassify into: ☒ New field ☐ Existing field

Reclassify field:

New field name:

Reclassify values:

Get Copy Clear new Auto...

Original value	New value
1	Mutlu
2	Mutlu
3	Orta
4	Mutsuz
5	Mutsuz

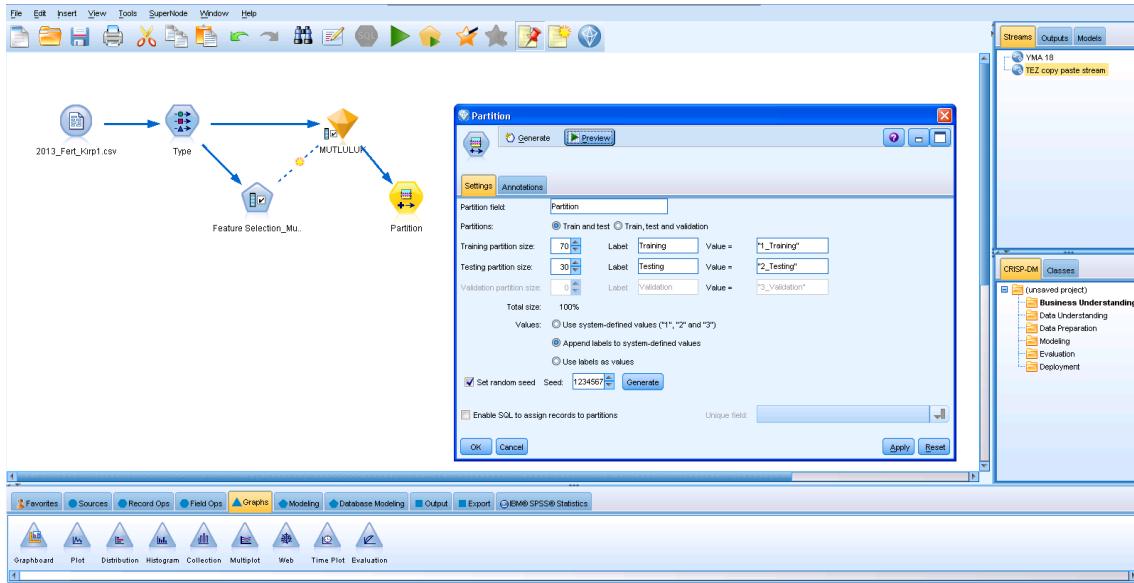
For unspecified values use: ☒ Original value ☐ Default value

OK Cancel Apply Reset

Şekil 5.5. Reclassify işlemcisi ile dönüştürme.

5.3.1 Veri setinin bölünmesi (training ve testing)

Veri setinin doğruluğunu test etmek için modelleme işleminden önce veriler eğitim ve test olmak üzere iki parçaya bölünür, bölünme “Field.Ops” paleti altında yer alan “partition” işlemcisi ile gerçekleştirilir. Varsayılan olarak %50 eğitim ve %50 test tanımlanmıştır. Burada %70 eğitim (training) ve %30 test (testing) seçerek analiz işlemine devam edilmiştir (Şekil 5.6).

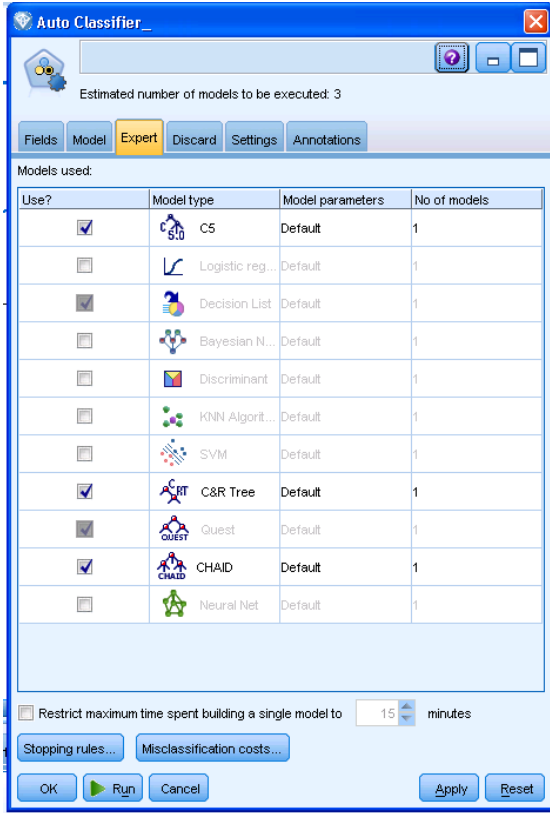


Şekil 5.6. Verilerin partiton işlemcisi ile bölünmesi.

5.3.2 Otomatik sınıflama işlemcisi ile kullanılacak modelin belirlenmesi

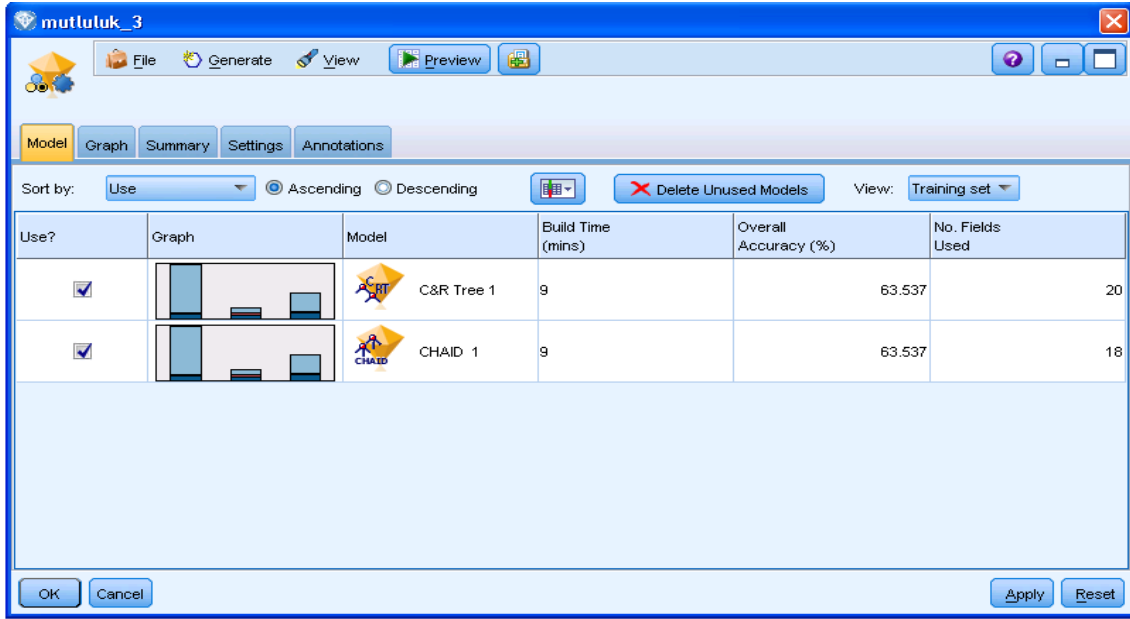
Otomatik modelleme işlemcisi farklı algoritmaları karşılaştırır ve modelleme için elimizde ki veri yapısına uygun olan en iyi algoritmanın belirlenmesini sağlar. Bağımlı değişkenin sürekli ya da kategorik olmasına göre kullanılacak otomatik modelleme işlemcileri bulunmaktadır. Elimizdeki veri kategorik ve sınıflama işlemi yapılacağı için otomatik sınıflama işlemcisi (Auto Classifier) kullanılacaktır (Şekil 5.7). Bu işlemci ile sınıflama ölçme düzeyinde bağımlı değişken için birden çok bağımlı değişken test eder ve karşılaştırır. Modelleme işleminde kullanılacak değişkenler kategorik ya da sürekli olabilir ancak bağımlı değişken kategorik olmak zorundadır.

Otomatik sınıflama işlemcisi tezde bahsettiğimiz CHAID, C&RT, QUEST ve C5.0 karar ağacı algoritmalarını desteklemektedir.



Şekil 5.7. Auto classifier işlemcisi.

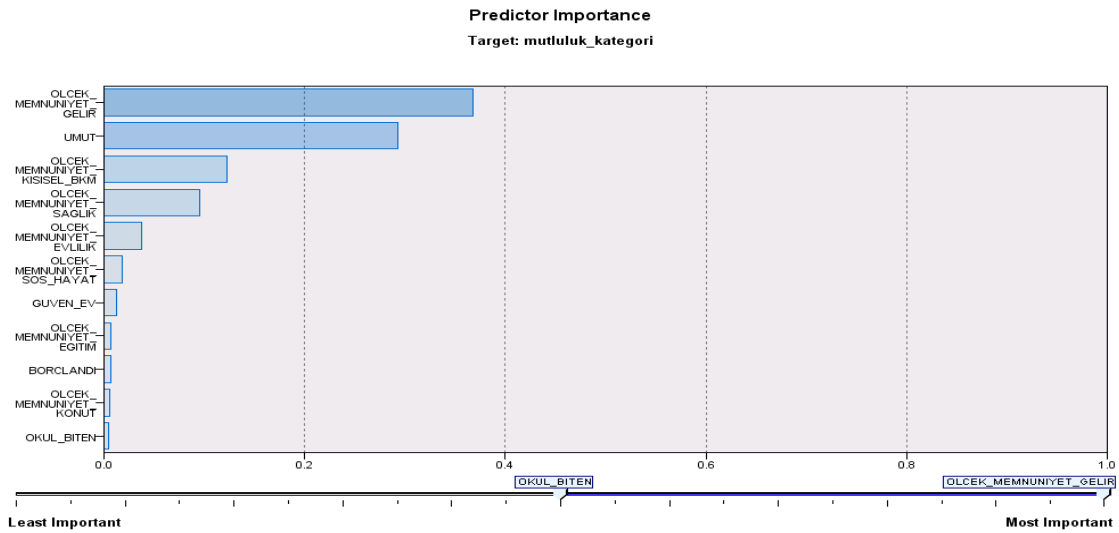
Auto Classifier işlemcisinde karar ağacı algoritmaları seçildikten sonra nodül çalıştırılır. Çıkan sonuca göre elimizdeki veri yapısına uygun olarak CHAID ve C&RT algoritmaları ile model oluşturulabileceği belirlenmiştir. Auto Classifier işlemcisine göre C5.0 ve QUEST algoritmaları ile ağaç oluşturulamayacağı belirlenmiştir. Ancak partition işlemcisinden sonra C5.0 ve QUEST algoritmaları bağlanmış ve ağaç oluşturulamayacağına dair hata uyarısı vermiştir. Bu bağlamda CHAID ve C&RT algoritmaları ile oluşturulan ağaç yapıları ve bu algoritmaların ağaç kurulmasında önemli gördüğü değişkenler aşağıda belirtilmiştir (Şekil 5.8).



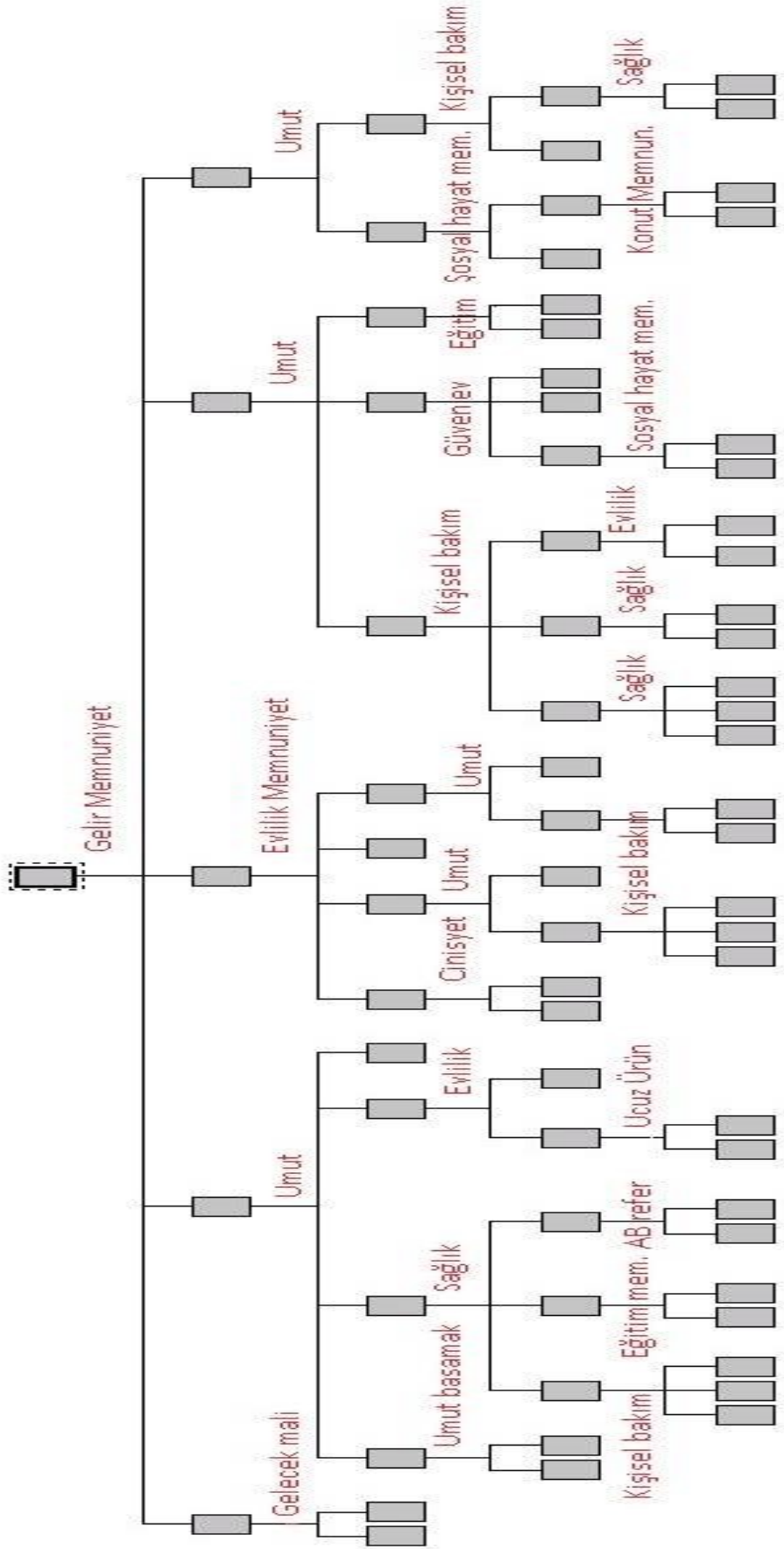
Şekil 5.8 Auto classifier işlemcisi sonuçları.

5.3.3 CHAID Algoritması

CHAID algoritmasının en önemli gördüğü değişken gelirinden memnun olma durumu olmuştur, bunu umut değişkeni takip etmiş ve diğer önemli gördüğü değişkenler Şekil 5.9 yer almaktadır.

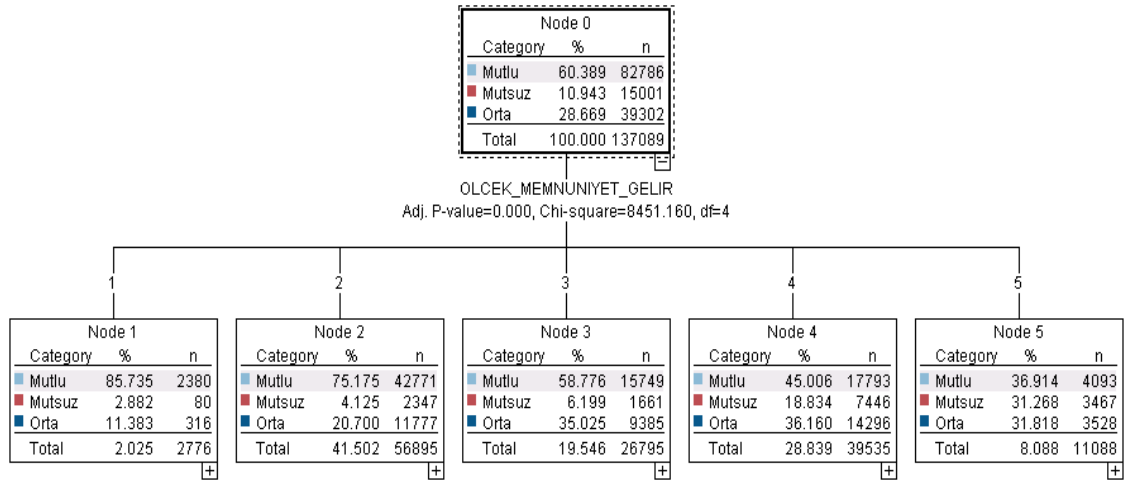


Şekil 5.9. CHAID algoritmasına göre önemli değişkenler.



Şekil 5.10. CAHID algoritması ile oluşturulan karar ağacının genel görünümü.

Şekil 5.10’da CHAID algoritması ile oluşturulan karar ağacı gösterilmiştir. Toplam 196.204 kayıttan 137.089 kayıt ile karar ağacı oluşturulmuştur. Geri kalan kayıtlar ile oluşturulan model test edilecektir. Kök düğüm (node 0) incelendiğinde “Mutlu” bireylerin oranı %60.389, “Orta” düzeyde bireylerin oranı 28.669, “Mutsuz” bireylerin oranı %10.943 olarak bulunmuştur (Şekil 5.11).



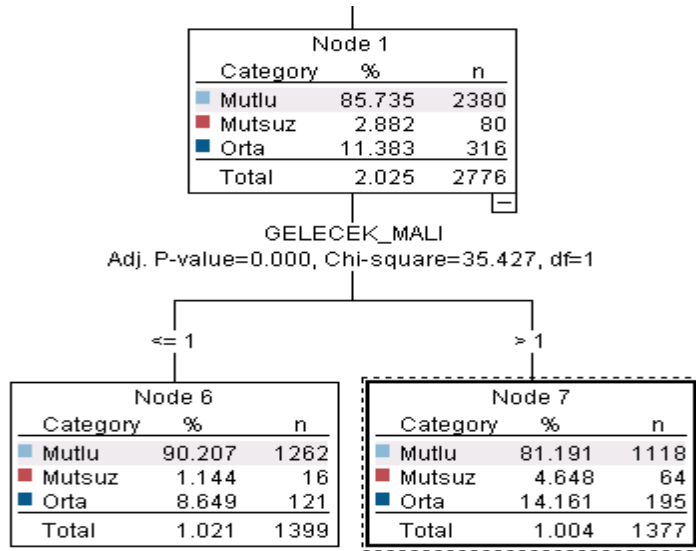
Şekil 5.11. CHAID kök düğüm (node0) bölünmesi.

Oluşturulan ağaca göre dallanma işleminde “Aylık hanehalkı gelirinizden memnun musunuz?” sorusuna verilen “Çok memnun, memnun, orta, memnun değil ve hiç memnun değil ” cevapları ile başlanmıştır. Buna göre 5 düzeydeki gelir memnuniyetine göre dallanma gerçekleşmiştir.

Node1’e göre Hanehalkı gelirinden “çok memnun” olan bireyler incelendiğinde mutlu olan bireylerin oranı %85.735 olduğu görülmektedir. Bundan sonra gelen “Önümüzdeki yıl hanenizin mali durumu nasıl olacak?” sorusuna “Daha iyi olacak” yanıtını verenler için mutluluk oranının %90.207 olduğu gözlenmiştir (Şekil 5.12).

Karar ağaçlarında eğer-sonra yapısında karar kuralları oluşturulabileceği belirtilmişti, burada düğüm 1 için aşağıdaki gibi kural oluşturulabilir.

If Gelir=Çok memnun **and** Gelecek mali durum= Daha iyi olacak **Then** Birey=Mutlu



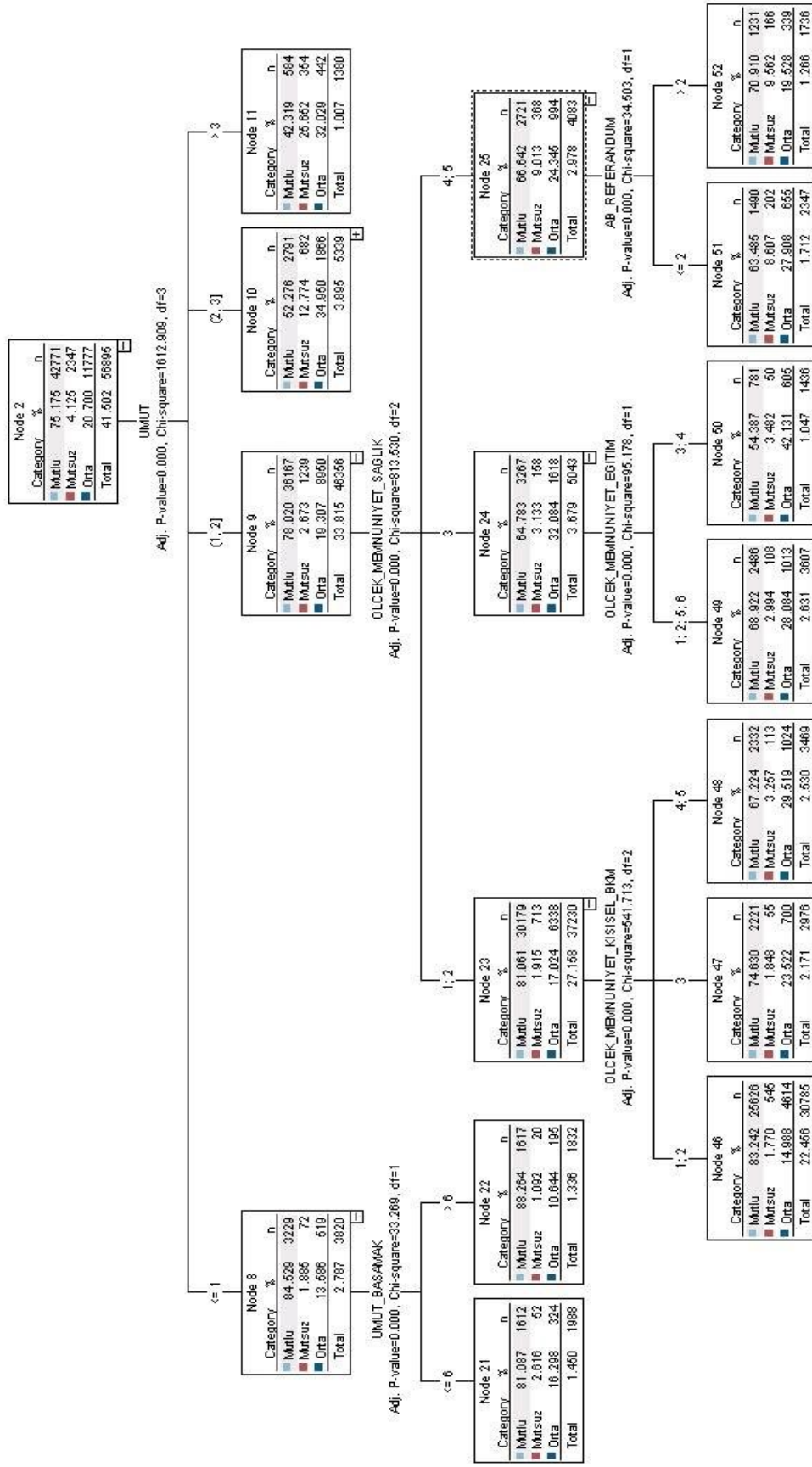
Şekil 5.12. CHAID node1 bölünmesi.

Node2’ye göre Hanehalkı gelirinden “memnun” olan bireyler incelendiğinde mutlu olan bireylerin oranı %75.175 olduğu görülmektedir. Bundan sonra ise “Kendi geleceğinizden ne kadar umutlusunuz?” sorusuna verilen “çok umutlu, umutlu, umutlu değil ve hiç umutlu değil” yanıtlarına göre dallandırma işlemi gerçekleştiği görülmektedir (Şekil 5.13).

Geleceğinden “çok umutlu” olanlar için mutluluk oranı %84.529 bulunmuştur. Burada “Türkiye’de yaşayan insanların refah düzeyini "0" basamağı en düşük "10" basamağı en yüksek" düzey olarak düşündüğünüzde, kendinizi hangi düzeyde görüyorsunuz?” sorusuna verilen cevaba göre bölünme gerçekleşmiştir ve 6 basamağına eşit ve düşük olanlar ile 6 basamağından yüksek olanlar algoritma tarafından bir grupta kategorize edilmiştir. 6 basamağından yüksek olan grup için mutluluk oranı %88.264’e yükselmiştir.

Geleceğinden “umutlu” olanlar için mutluluk oranı %78.020’ye çıktığı ancak “Sağlığınızdan memnun musunuz?” sorusuna verilen cevaba göre bir bölünme daha gerçekleşmiştir. Sağlığından “çok memnun ve memnun” olanlar için mutluluk oranı %81.061 bulunmuş ve “Kendinize ayırdığınız zamandan memnun musunuz?” sorusu için bir bölünme daha gerçekleşmiştir. Kendilerine ayırdıkları vakitten “çok memnun, memnun” olanlar için mutluluk oranı %83.242’ye yükselmiştir.

Geleceğinden “umutlu değilim” diyenler için mutluluk oranının %52.276’ya düştüğü “hiç umutlu değilim” diyenler için ise %42.319’a kadar düştüğü gözlenmiştir.



Şekil 5.13. CAHİD node2 bölünmesi.

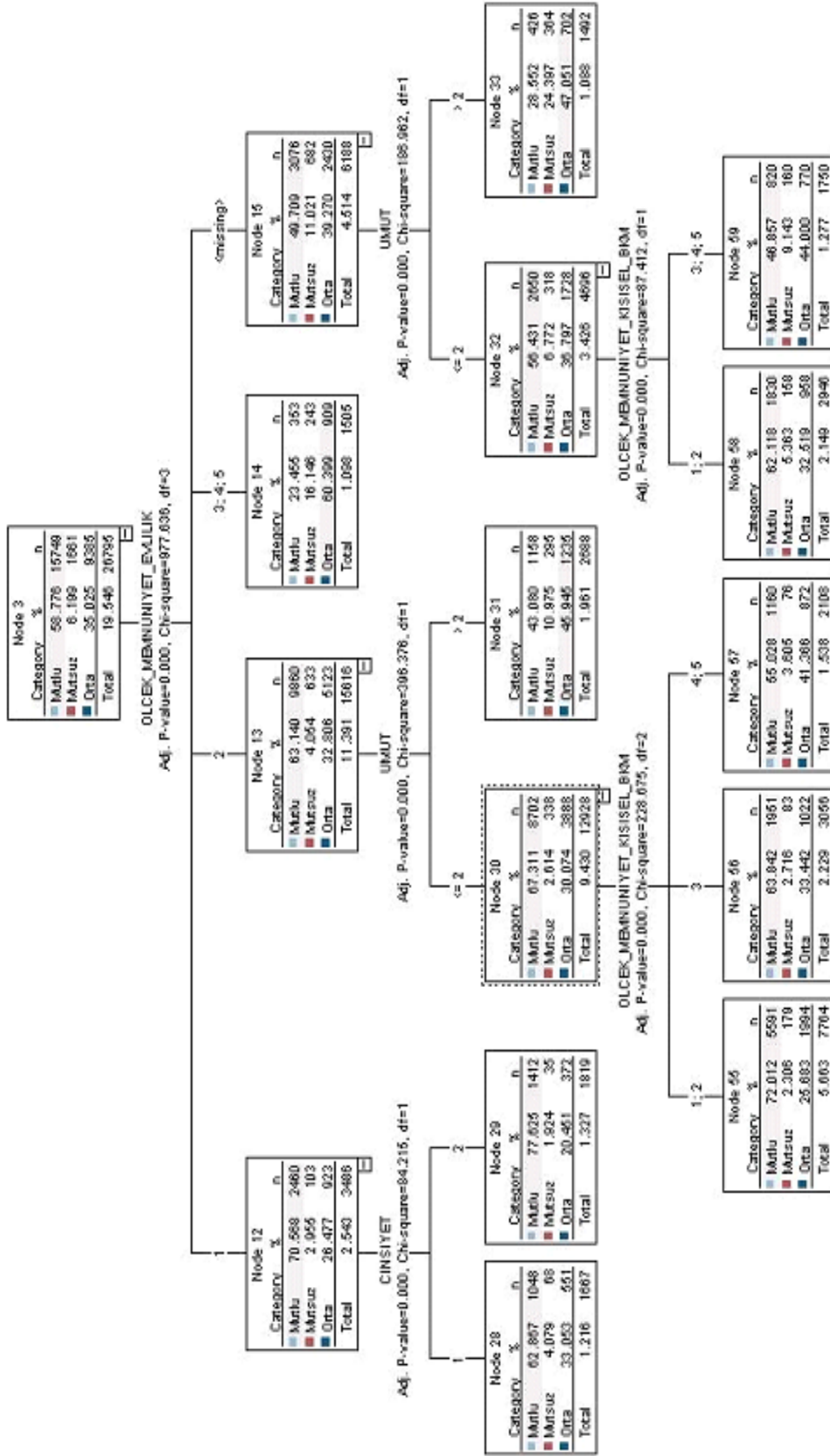
Node3'e göre hanehalkı gelirinden “orta” düzeyde memnun olan bireyler için mutluluk oranının %58.776 olduğu gözlenmiştir ve buradaki bölünmenin “Evliliğinizden memnun musunuz?” sorusuna verilen cevaplara göre gerçekleşeceği belirlenmiştir. Burada “çok memnun”, “memnun”, “orta, memnun değil ve hiç memnun değil” ile “missing” olmak üzere dört kategoriye göre bölünme gerçekleştiği gözlenmiştir. CHAID algoritmasının bir özelliği kayıp gözlemleri bir kümede toplayıp ondan sonra bu grubu ayrı analiz etmesidir. Burada evlilik memnuniyetleri hakkında bilgi alınamayan bireyleri bir grupta toplamış ve ona göre incelemiştir (Şekil 5.14).

Evliliğinde “çok memnun” olan bireyler için mutluluk oranı %70.568 çıkmış ancak burada cinsiyete göre bir bölünme gerçekleşmiştir ve erkekler için mutluluk oranı %62.867 iken kadınlar için mutluluk oranı %77.625 bulunmuştur.

Evliliğinden “memnun” olan bireyler için mutluluk oranı %63.140 bulunmuş ve “geleceğin den umutlu musunuz?” değişkeni için bölünme gerçekleşmiştir. “çok umutlu ve umutlu” bir kategori olmuş ve mutluluk oranı %67.311 olmuştur. Bu grup için “Kendinize ayırdığınız zamandan memnun musunuz?” sorusu için son bir bölünme daha gerçekleşmiş ve “çok memnun, memnun” olanlar için mutluluk oranı %72.012'ye yükselmiştir.

Evliliğinden “orta, memnun değil ve hiç memnun değil” olan grup için mutluluk oranı %23.455 bulunmuştur.

Evlilik memnuniyeti durumu bilinmeyenler bir grupta toplanmış ve mutluluk oranı %49.709 bulunmuştur. Evlilik memnuniyeti bilinmeyen grup için “Kendi geleceğinizden ne kadar umutlusunuz?” sorusu için bölünme gerçekleşmiş ve “çok umutlu, umutlu” olanlar için mutluluk oranı %56.431 bulunmuş daha sonra “Kendinize ayırdığınız zamandan memnun musunuz?” sorusu için son bölünme gerçekleşmiş ve “çok memnun ve memnun” olanlar için mutluluk oranı %62.118 olduğu gözlenmiştir.



Şekil 5.14. CAHİD node3 bölünmesi.

Node4'e göre hanehalkı gelirinden “memnun değil” diyenler için mutluluk oranının %45.006 olduğu gözlenmiştir (Şekil 5.15). Bundan sonra ise “Kendi geleceğinizden ne kadar umutlusunuz?” sorusuna verilen “Çok umutlu, umutlu, umutlu değil ve hiç umutlu değil” yanıtlarına göre dallandırma işlemi gerçekleştirilmiştir.

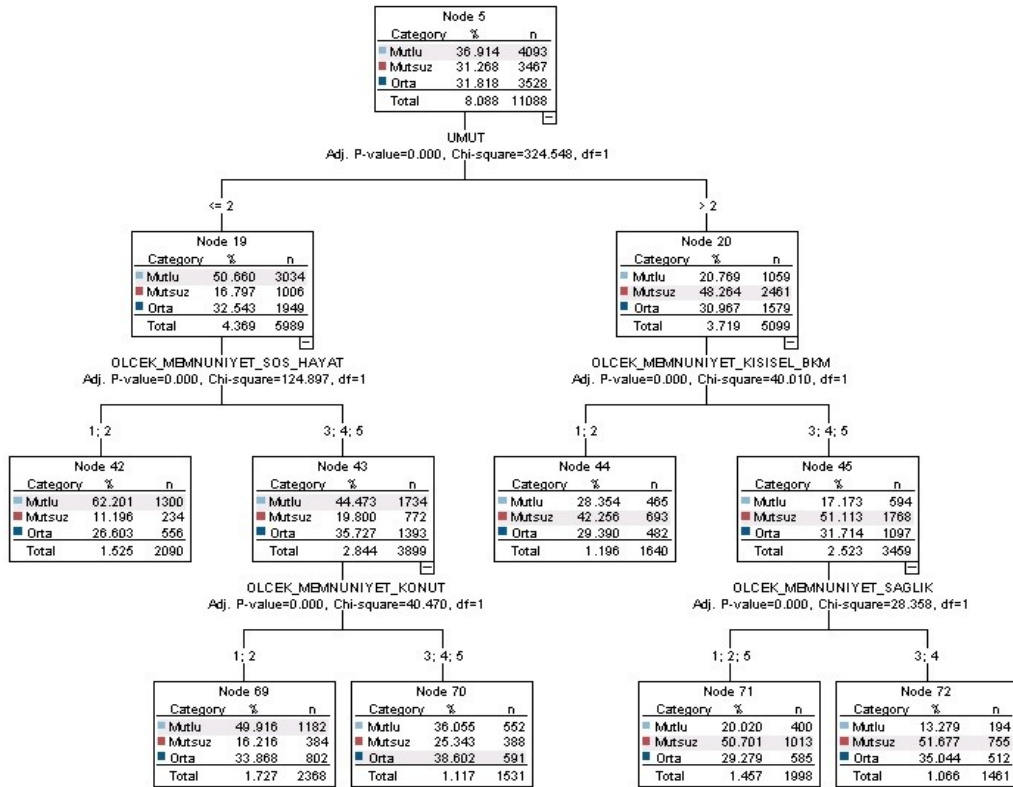
Geleceğinden “çok umutlu ve umutlu” olanlar bir grupta toplanmış ve mutluluk oranı %54.798'e yükselmiştir. Burada “Kendinize ayırdığınız zamandan memnun musunuz?” değişkeni için bölünme gerçekleşmiş ve “çok memnun ve memnun” olanlar bir kategoride toplanmıştır mutluluk oranı %61.286'ya yükselmiştir. Daha sonra “Sağlığınızdan memnun musunuz?” değişkeni için bölünme gerçekleşmiştir, “çok memnun ile memnun” olanlar bir grupta toplanmış ve mutluluk oranı %65.829'a yükselmiştir. “Kendinize ayırdığınız zamandan memnun musunuz?” sorusuna “orta” diyenler için sağlık memnuniyetine göre bölünme gerçekleşmiş, “memnun değil ve hiç memnun değil” diyenler için ise evlilik memnuniyetine göre bölünme gerçekleşmiştir.

Geleceğinden “umutlu değil” diyenler için mutluluk oranı %29.717'ye düşmüştür. “Evinizde yalnızken kendinizi ne kadar güvende hissediyorsunuz?” sorusuna verilen “çok güvenli, güvenli” ile “orta” ve “güvensiz ve çok güvensiz” grupları olmak üzere üç adet bölünme gerçekleşmiştir. “çok güvenli ve güvenli” hissedenler için mutluluk oranı %32.078 bulunmuş ve “kendinize ayırdığınız zamandan memnun musunuz?” sorusuna göre bir bölünme daha gerçekleşmiştir. Burada dikkat çeken durum kendilerinin evlerinde “güvensiz ve çok güvensiz” hissedenler için mutsuzluk oranı %39.873 olduğu görülmüştür.

Geleceğinden “hiç umutlu değil” diyenler için mutluluk oranı %21.076'ya düşmüştür. Burada “şimdiye kadar almış olduğunuz eğitimden memnun musunuz?” Sorusuna göre bir dallanma gerçekleşmiştir ve “orta, memnun değil ve hiç memnun değil” olanlar için mutluluk oranı %17.986 bulunmuş ve mutsuzluk oranı %44.549 olduğu görülmüştür.

Node5'e göre hanehalkı gelirinden “hiç memnun değil” diyenler için mutluluk oranının %36.403 olduğu gözlenmiştir. Bundan sonra ise “Kendi geleceğinizden ne kadar umutlusunuz?” sorusuna verilen “çok umutlu, umutlu” ve “umutlu değil ve hiç umutlu değil” gruplamasına göre iki adet dallandırma işlemi gerçekleştirildiği görülmektedir (Şekil 5.16).

Geleceğinden “umutlu değil ve hiç umutlu değil” grubunda bulunan bireylerde mutluluk oranı %20.769 ve mutsuzluk oranının %48.264 olarak gerçekleşmiştir. Burada “Kendinize ayırdığınız zamandan memnun musunuz?” değişkenine göre bölünme gerçekleşmiş ve “çok memnun ve memnun” ile “orta, memnun değil ve hiç memnun değil” olmak üzere iki dala ayrılmıştır. Memnun olmayan grubundaki bireyler için mutluluk oranı %17.173 ve mutsuzluk oranı %51.113 olduğu görülmüştür. Daha sonra Sağlık memnuniyetine göre “çok memnun, memnun, hiç memnun değil” ile “orta, memnun değil” gruplanmış şekilde son bir bölünme gerçekleşmiştir. Burada sağlık memnuniyeti “orta ve memnun değil” olan grup için mutluluk değeri %13.279 ve mutsuzluk değeri %51.677 ile veri setindeki en düşük yaşam memnuniyetine sahip grup olmuştur.



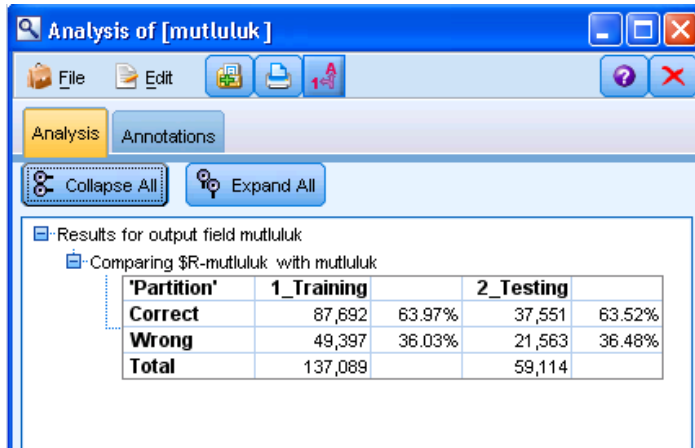
Şekil 5.16. CHAID node5 bölünmesi.

CHAID ile oluşturulan karar ağacı incelendiğinde en yüksek mutluluk düzeyi; gelirinden çok memnun olma ve önümüzdeki yıl mali durumunun iyileşeceğini düşünenler için mutluluk oranının %90.207 olduğu gözlenmiştir.

En düşük mutluluk oranı ile en yüksek mutsuzluk oranı ise; gelirinden “hiç memnun” olmayanlar, geleceğinden umutlu olmayanlar, kendine ayırdığı zamandan ve sağlığından memnun olmayanlar için mutluluk oranı %13.279 ve mutsuzluk oranı %51.677 olduğu gözlenmiştir.

5.3.4 CHAID algoritmasının başarısının test edilmesi

Partition işlemcisi ile %70 eğitim olarak ayrılan veri setinde CHAID algoritması ile karar ağacı oluşturulduktan sonra %30 test olarak ayrılan kısımda ağaç test edilmektedir. Algoritmaya analiz işlemcisi eklenerek inceleme gerçekleştirilir. Bu sonuca göre eğitim sonucu doğru sınıflama yüzdesi %63.97 ve modelin geliştirilmesinde hiç görmediği test kümesi üzerinde oluşturulan ağacın doğru sınıflama yüzdesi %63.52 olduğu gözlenmiştir. Eğitim ve test sınıflama yüzdeleri neredeyse aynı çıkmıştır (Şekil 5.17).



'Partition'	1_Training		2_Testing	
Correct	87,892	63.97%	37,551	63.52%
Wrong	49,397	36.03%	21,563	36.48%
Total	137,089		59,114	

Şekil 5.17. CHAID algoritmasının test edilmesi.

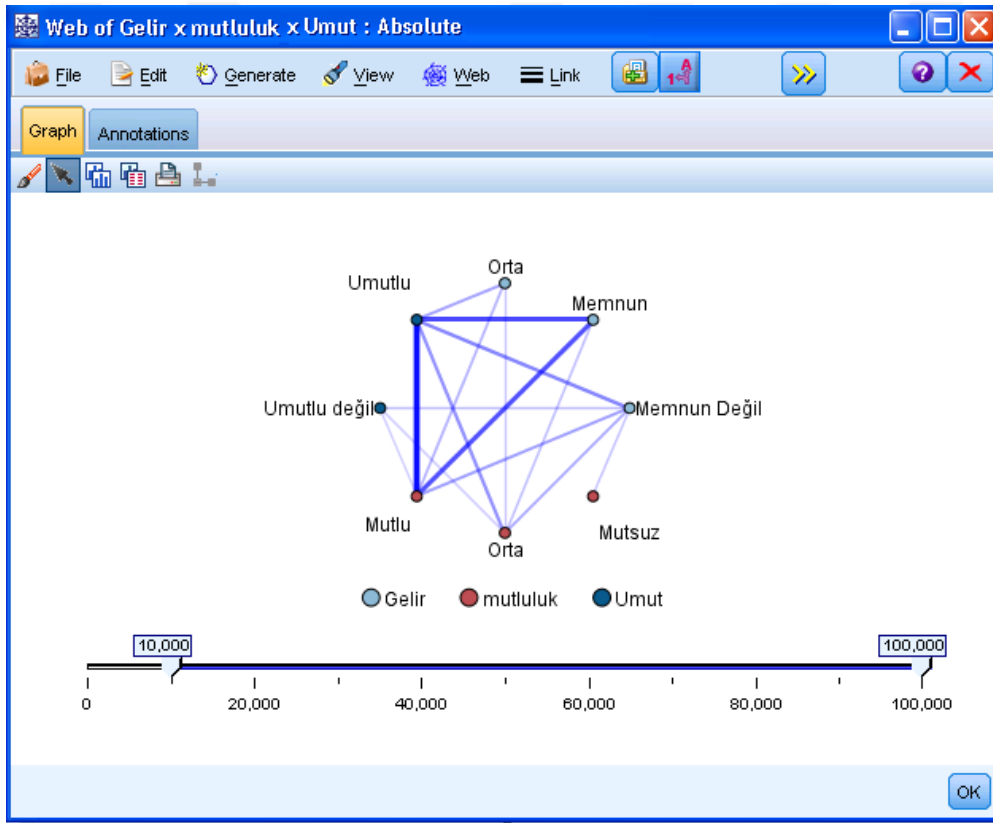
5.3.5 CHAID algoritmasındaki önemli değişkenler için web grafiği

İki yada daha fazla kategorik veri arasındaki ilişkinin gücünü görmek için kullanılan web işlemcisi ile CHAID ile oluşturulan karar ağacında mutluluk hedef

değişkenini tahmin etmede önemli olan değişkenlerden; gelir memnuniyeti ve umut değişkenleri web grafiği ile incelenmiştir.

Nokta ile gösterilen değişkenler arasındaki düz çizgilerle birbirlerine bağlanırlar ve çizgilerin kalın olması iki değişken arasında güçlü bir ilişki olduğunu gösterir. İnce olması zayıf bir ilişki olduğunu gösterir.

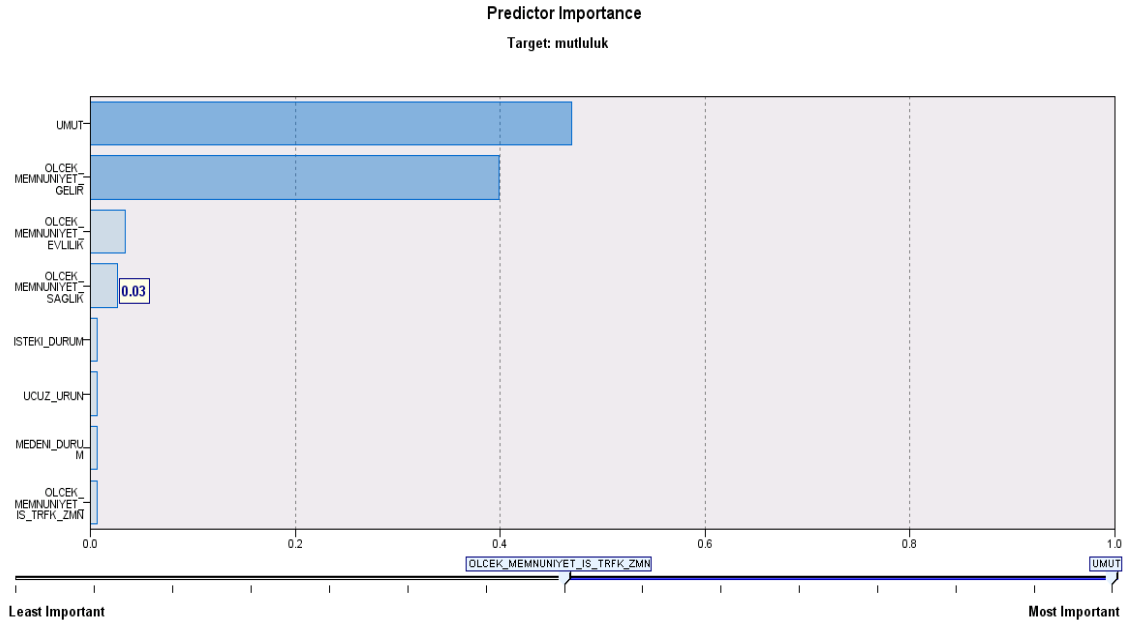
Web grafiği incelenirse en kuvvetli ilişkinin mutlu olmak ile geleceğinden umutlu olma arasında olduğu görülmektedir. Daha sonra ise gelirinden memnuniyet ile geleceğinden umutlu olma arasında kuvvetli bir ilişki olduğu görülmektedir. Üçüncü olarak ise gelirinden memnun olma ile mutluluk arasında bir ilişki olduğu gözlenmektedir (Şekil 5.18).



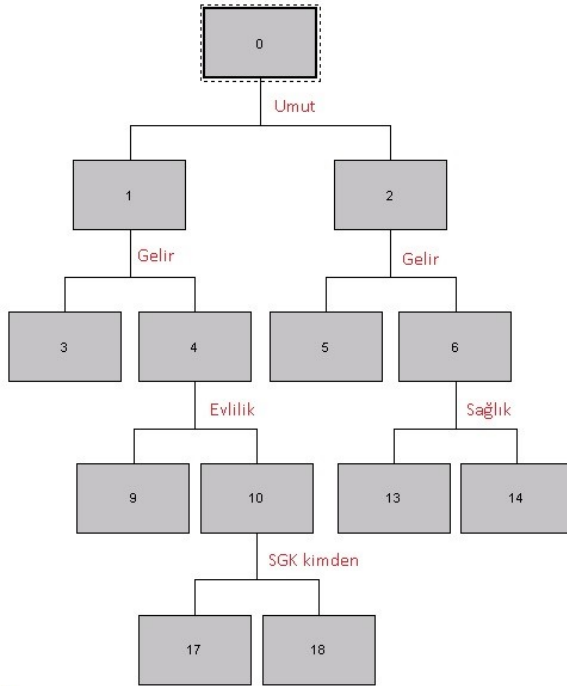
Şekil 5.18. CHAID algoritmasındaki önemli değişkenler için web grafiği.

5.3.6 C&RT Algoritması

C&RT algoritmasının en önemli gördüğü değişken geleceğinden umutlu olup olmama olmuştur bunu gelir ve evlilik memnuniyeti değişkenleri takip etmiştir (Şekil 5.19). Şekil 5.20’de ise ağacın genel görünümü yer almaktadır.

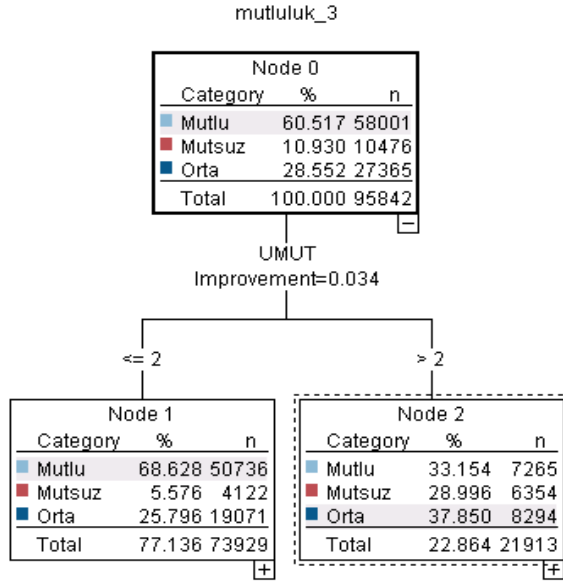


Şekil 5.19. C&RT algoritmasına göre önemli değişkenler.



Şekil 5.20. C&RT algoritması ile oluşturulan karar ağacının genel görünümü.

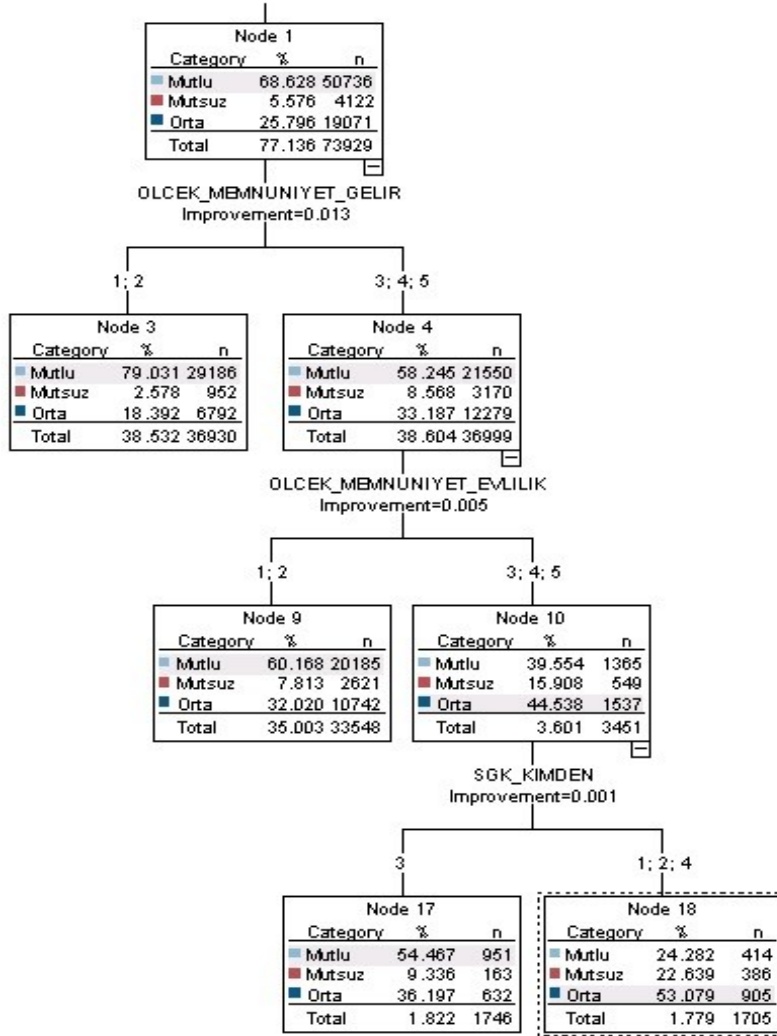
C&RT algoritması incelendiğinde kök düğümde “Kendi geleceğinizden ne kadar umutlusunuz?” değişkeni yer almıştır. Geleceğinden “çok umutlu, umutlu” olanlar bir grup “umutlu değil, hiç umutlu değil” olanlar bir gruba bölünmüştür (Şekil 5.21).



Şekil 5.21. C&RT node0 bölünmesi.

Node1’e göre “Kendi geleceğinizden ne kadar umutlusunuz?” sorusuna verilen “çok umutlu, umutlu” cevaplarına göre bölünme gerçekleşmiş ve mutluluk oranı %68.628 olduğu görülmüştür. Daha sonra “Aylık hanehalkı gelirinizden memnunuz?” sorusuna göre bölünme olmuş ve “çok memnun, memnun” cevabını verenler bir grup oluşturmuş ve mutluluk oranı %79.031 olduğu görülmüştür. “Orta, memnun değil ve hiç memnun değil” diyenler için mutluluk oranı %58.245 ve gelirinden memnun olmayanlar için bir bölünme daha gerçekleşmiştir. Bu bölünme “Evliliğinizden memnun musunuz?” sorusuna göre olmuş ve “çok memnun, memnun” cevabını verenler bir grup oluşturmuş ve mutluluk oranı %60.168 olduğu gözlenmiştir. “Orta, memnun değil ve hiç memnun değil” diyenler için mutluluk oranı %39.554’e düştüğü gözlenmiştir. En son olarak evliliğinden memnun olmayanlar için bir bölünme gerçekleşmiştir. Son bölünme “Kimin üzerinden sosyal güvenlik kapsamındasınız?” sorusuna göre olmuştur. Bu soruya verilen “Anne/Baba” cevabı bir grup ve “Kendi, Eşi ve Çocukları” değişkenleri bir grup olacak şekilde bölünme işlemi sonlamıştır. Anne/Baba cevabını veren grup için mutluluk oranı %54.467 olmuştur ancak “Kendi,

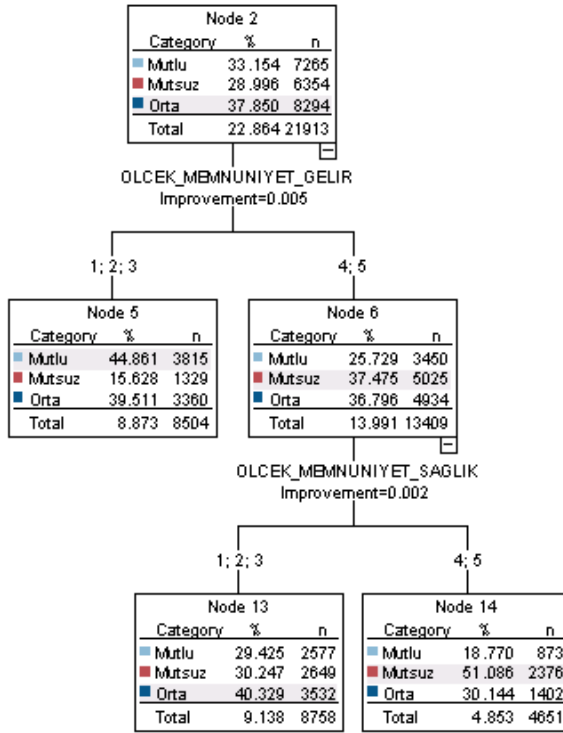
eşi, çocukları” cevabını veren grup için mutluluk oranı %24.282’ye kadar düştüğü gözlenmiştir (Şekil 5.22).



Şekil 5.22. C&RT node1 bölünmesi.

Node2’ye göre “Kendi geleceğinizden ne kadar umutlusunuz?” sorusuna verilen “umutlu değil, hiç umutlu değil” cevaplarına göre bölünme gerçekleşmiş ve mutluluk oranı %33.154 olarak elde edilmiştir. Daha sonra “Aylık hanehalkı gelirinizden memnun musunuz?” sorusuna göre bölünme olmuş ve “çok memnun, memnun, orta” cevabını verenler bir grup olmuş ve bu grup için mutluluk oranı %44.861 olduğu gözlenmiştir, “memnun değil, hiç memnun değil” diyenler için mutluluk oranı %25.729 olmuştur. Gelirinden memnun olmayanlar için bir bölünme daha gerçekleşmiştir ve bu bölünme “Sağlığınızdan memnun musunuz?” sorusuna göre olmuştur ve “çok memnun, memnun, orta” cevabını verenler bir grup oluşturmuş ve bu grup için mutluluk oranı

%29.425 olmuştur. “memnun değil ve hiç memnun değil” diyenler için mutluluk oranı %18.770’e düştüğü gözlenmiştir (Şekil 5.23).



Şekil 5.23. C&RT node2 bölünmesi.

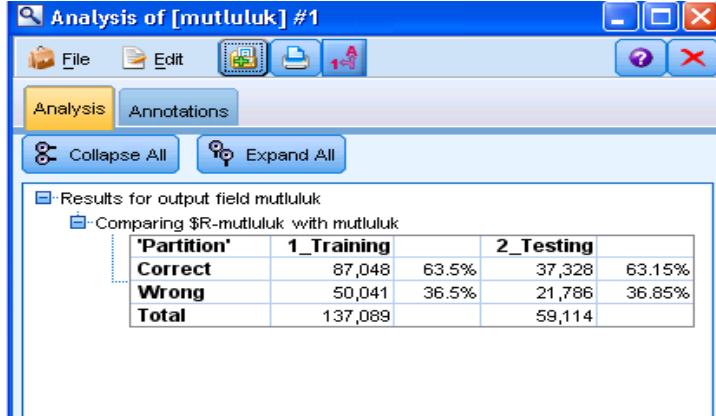
C&RT ile oluşturulan karar ağacı incelendiğinde en yüksek mutluluk düzeyi; geleceğinden umutlu olanlar ve hanehalkı gelirinden memnun olanlar için gözlenmiş ve mutluluk oranının %79.031 olduğu görülmüştür.

En düşük mutluluk oranı ise; geleceğinden umutlu olmayanlar, hanehalkı gelirinden ile sağlığından memnun olmayanlar için gözlenmiş ve mutluluk oranı %18.770 olmuştur.

5.3.7 C&RT algoritmasının başarısının test edilmesi

C&RT algoritması da eğitim ve test olarak ayrılan veri seti üzerinde analiz işlemcisi ile incelenir. Eğitim verileri ile oluşturulan ağacın doğru sınıflama yüzdesi %63.5 ve modelin geliştirilmesinde hiç görmediği test verileri üzerinde oluşturulan

ağacın doğru sınıflama yüzdesi %63.15 olduğu gözlenmiştir. Eğitim ve test verilerine ait sınıflama yüzdelерinin birbirlerine çok yakın sonuçlar verdiği görülmüştür (Şekil 5.24).



The screenshot shows the 'Analysis of [mutluluk] #1' window in SPSS. The 'Results for output field mutluluk' section is expanded, showing a comparison of the predicted vs. actual values for the 'mutluluk' variable. The results are presented in a table with columns for 'Partition', '1_Training', and '2_Testing'. The table shows the number of correct and wrong classifications and the total number of cases for both training and testing sets.

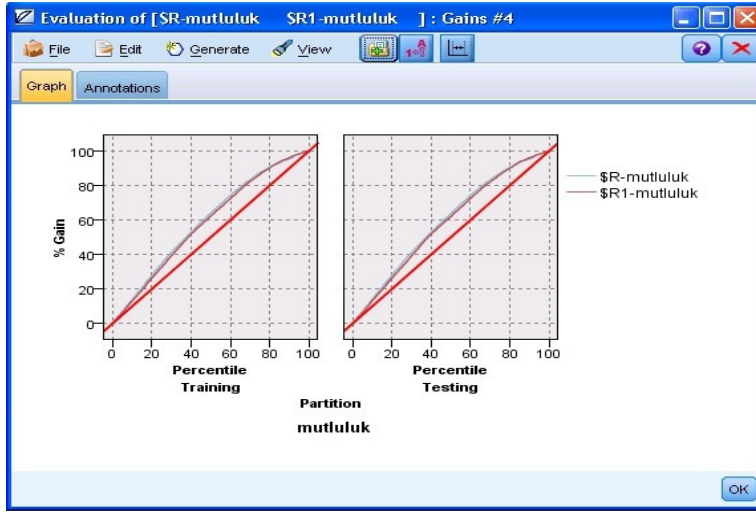
Partition	1_Training		2_Testing	
Correct	87,048	63.5%	37,328	63.15%
Wrong	50,041	36.5%	21,786	36.85%
Total	137,089		59,114	

Şekil 5.24. C&RT algoritmasının test edilmesi.

5.3.8 C&RT ve CHAID algoritmasının karşılaştırılması

IBM SPSS Modeler’da birden fazla modelin performanslarını değerlendirmek için kullanılan yöntemler arasında; doğru sınıflandırma yüzdesi, korelasyon katsayısı, ortalama standart hata ve “evaluation” işlemcisi (grafığı) bulunmaktadır. Bağımsız ve bağımlı değişkenler kategorik verilerden oluştuğu için değerlendirme işleminde doğru sınıflandırma yüzdesi ve “evaluation” işlemcisi ile yapılmıştır.

Bu kapsamda CHAID ve C&RT algoritmaları ile oluşturulan karar ağaçlarının doğru sınıflama skorları sırası ile; %63,52 ve %63,15 olduğu ve birbirlerine yakın sonuçlar verdiği gözlenmiştir. Daha sonra “evaluation” işlemcisi ile bu iki algoritma kıyaslanmış ve modelin her iki eğitim ve test örneklemeleri içinde de birbirlerine yakın sonuçlar verdiği gözlenmiştir (Şekil 5.25).



Şekil 5.25. C&RT ve CHAID algoritmalarının karşılaştırılması.

Çizelge 5.1’de CHAID ve C&RT algoritması ile oluşturulan karar ağaçlarında kullanılan değişkenler verilmiştir. Ek 1’de bu değişkenlere ait sorular yer almaktadır.

Çizelge 5.1. CHAID ve C&RT ağacındaki değişkenler

Soru	CHAID	Soru	C&RT
S1	Gelir Memnuniyeti	S1	Gelir Memnuniyeti
S2	Gelecek Mali Durum	S3	Umut
S3	Umut	S6	Sağlık
S4	Umut Basamak	S9	Evlilik
S5	Kişisel Bakım	S15	SGK kimden
S6	Sağlık		
S7	Eğitim Memnuniyeti		
S8	AB Referandum		
S9	Evlilik Memnuniyeti		
S10	Ucuz Ürün Tercihi		
S11	Cinsiyet		
S12	Ev’de Güvende Hissetme		
S13	Sosyal Hayat		
S14	Konut Memnuniyeti		

6. SONUÇ VE TARTIŞMA

Tez çalışmasında veri madenciliği tanımından, sürecinden ve istatistik bilimi ile olan ilişkisinden bahsedildikten sonra karar ağaçlarından ve sık kullanılan algoritmaların özelliklerinden bahsedilmiştir. Daha sonra TÜİK tarafından 2013 yılında yapılan yaşam memnuniyeti araştırmasından elde edilen B grubu mikro veriler ile uygulama yapılmıştır. Uygulama aşamasında veri madenciliği uygulamaları yapılan IBM SPSS Modeler 14.2 kullanılmıştır. Bu kapsamda program kullanımı hakkında önemli olan işlemciler için tanımlamalar yapılmış ve karar ağacı algoritmaları ile modeller oluşturulmuştur.

Uygulamada kullanılan veri seti 196.203 kayıttan ve 239 değişkenden oluşmaktadır. Değişken sayısı özellik seçimi (feature selection) işlemcisi ile azaltılmış ve hedef değişkeni açıklamada önemli rol oynadığı belirlenen 135 değişken ile karar ağacı algoritmaları uygulanmıştır. Anket soru formunda yer alan “Yaşamınızı bir bütün olarak düşündüğünüzde ne kadar mutlusunuz?” sorusu hedef değişken olarak belirlenmiştir. Bu soruya verilen “Çok mutlu, mutlu, orta, mutsuz ve çok mutsuz” cevapları reclassify işlemcisi ile “mutlu, orta, mutsuz” olacak şekilde tekrardan kodlanmıştır ve bu üç kategori hedef sınıfları oluşturulmuştur.

Auto Classifier işlemcisi ile karar ağacı algoritmaları test edilmiş. Çıkan sonuca göre elimizdeki veri yapısına uygun olarak CHAID ve C&RT algoritmaları ile model oluşturulabileceği belirlenmiştir ve bu algoritmalar ile karar ağaçları oluşturulmuştur.

CHAID ve C&RT algoritmaları ile oluşturulan karar ağaçları incelendiğinde doğru sınıflama doğruluk oranlarının birbirlerine yakın sonuçlar verdiği gözlenmiştir. Ancak CHAID algoritması mutluluk bağımlı değişkeni açıklamada 14 bağımsız değişken ile ağaç oluştururken; C&RT algoritması mutluluk hedef değişkenini açıklamada 5 bağımsız değişken ile ağaç oluşturmuştur. CHAID algoritmasının daha fazla açıklayıcı değişken ile karar ağacı oluşturmasından dolayı CHAID algoritması daha araştırmacı yapıya sahip olduğu gözlenmiştir.

KAYNAKLAR

- Akpınar, H., 2000. Veri tabanlarında bilgi keşfi ve veri madenciliği, *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, 29 (1): 1-22.
- Albayrak, S. A., Yılmaz, K. Ş., 2009. Veri madenciliği: karar ağacı algoritmaları ve İMKB verileri üzerine bir uygulama. *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 14 (1): 31-52.
- Alkan A., 2007. Finansal uygulamalarda veri madenciliği. <http://docplayer.biz.tr/1672580-8-haziran-2007-tbd-istanbul-bilisim-kongresi.html>. *İstanbul Bilişim Kongresi*, 7-9 Haziran 2007, İstanbul.
Erişim Tarihi: 21.12.2015.
- Alpaydın, E., 2000, *Zeki Veri Madenciliği: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri*, Bilişim 2000 Eğitim Semineri.
- Babadağ, K. K., 2003. *Veri Madenciliği Yaklaşımı ve Veri Kalitesinin Artması İçin Kullanılması* (uzmanlık tezi, basılmamış). Türkiye İstatistik Kurumu, Ankara.
- Berry A. J. M., Linoff S. G., 2004. *Data Mining Techniques*. Wiley Publication.
- Bounsaythip, C., Runsala, R. E., 2001. *Overview of Data Mining for Customer Behavior Modeling*. VTT Information Technology.
- Brandão, A., Pereira, E., Prtela, F., Santos, M., Abelha, A., Machado, J., 2015. *Predicting the Risk Associated to Pregnancy Using Data Mining*. Algoritmi Research Centre, Universidade do Minho.
- Chen, S., 2016. *Detection Of Fraudulent Financial Statements Using The Hybrid Data Mining Approach*. Springer Plus.
- Coşkun, C., 2010. *Veri Madenciliği Algoritmaları Karşılaştırılması* (yüksek lisans tezi, basılmamış). DÜ, Fen Bilimleri Enstitüsü, Diyarbakır.
- Çakır, Ö., 2008. *Veri Madenciliğinde Sınıflandırma Yöntemlerinin Karşılaştırılması "Bankacılık Müşteri Veri Tabanı Üzerine Uygulama"* (doktora tezi, basılmamış). MÜ, Sosyal Bilimler Enstitüsü, İstanbul.
- Çalış, A., Kayapınar, S., Çetinyokuş, T., 2014. Veri madenciliğinde karar ağacı algoritmaları ile bilgisayar ve internet güvenliği üzerine bir uygulama. *Endüstri Mühendisliği Dergisi*, 25 (3-4): 2-19.

- Dalkılıç, H., Dalkılıç, F., 2015. Karar ağaçları destekli vadeli mevduat hesabı. <http://ab.org.tr/ab15/kabul.html> 17. Akademik Bilişim Konferansı, 4-5 Şubat 2015, Eskişehir. Erişim Tarihi: 21.12.2015
- Dondurmacı, G., 2011. *Veri Madenciliğinde Regresyon ağaçları ile Sınıflandırma ve Bir Uygulama* (doktora tezi, basılmamış). MSGSU, Fen Bilimleri Enstitüsü, İstanbul.
- Emel, G., G., Taşkın, Ç., 2005. Veri madenciliğinde karar ağaçları ve bir satış analizi uygulaması. *Eskişehir Osmangazi Üniversitesi Sosyal Bilimler Dergisi*, 6 (2): 221-239.
- Erşahin, B., 2008. Veri madenciliği; veriden bilgiye, masraftan değere. <http://www.arge.com/wpcontent/uploads/2013/02/VeriMadenciligi.pdf> ARGE Danışmanlık, İstanbul. Erişim Tarihi: 21.12.2015.
- Feelders, A., Daniels, H., 2000. Methodological and practical aspects of data mining. *Information and Management*, 37 : 271-281.
- Güler, K., G., Emeç, H., 2006. Yaşam memnuniyeti ve akademik başarıda iyimserlik etkisi. *Dokuz Eylül Üniversitesi İktisadi ve İdari Birimler Fakültesi Dergisi*, 21 (2): 129-149.
- Gürsoy Şimşek U. T., 2011. *Uygulamalı Veri Madenciliği Sektörel Analizler*, Pegem Akademi.
- Han, J., Kamber, M., 2006. *Data Mining Concepts and Techniques*, Second Edition, Morgan Kaufman Publishers.
- Kavurkacı, Ş., Aydın G. Z., Şamlı, R., 2011. Büyük ölçekli veri tabanlarında bilgi keşfi. *XIII. Akademik Bilişim Konferans Yayını*, 2-4 Şubat 2011, Malatya.161-169.
- Kaya, K., Köymen K., 2008. Veri madenciliği ve uygulama alanları. *Fırat Üniversitesi Doğu Anadolu Bölgesi Araştırmaları Dergisi*. 6 (2): 159-164.
- Kayri, M., 2014. Karar ağaçları. *Karar Ağaçları Çalıştayı*, Muş Alparslan Üniversitesi. Sunum tarihi: 6 Haziran 2014.
- Kocabaş, M., F., 2010. *Veri Madenciliği Süreci ve Gerçek Bir Veri Seti Üzerine Uygulama* (yüksek lisans tezi, basılmamış). HÜ, Fen Bilimleri Enstitüsü, Ankara.
- Koyuncugil, S., A., 2006. *Bulanık Veri Madenciliği Ve Sermaye Piyasalarına Uygulanması* (doktora tezi, basılmamış). AÜ, Fen Bilimleri Enstitüsü, Ankara.

- Köktürk, F., 2012. *K-En Yakın Komşuluk, Yapay Sinir Ağları Ve Karar Ağaçları Yöntemlerinin Sınıflandırma Başarılarının Karşılaştırılması* (doktora tezi, basılmamış). BEÜ, Sağlık Bilimleri Enstitüsü, Zonguldak.
- Kuzey, C., 2012. *Veri Madenciliğinde Destek Vektör Makinaları ve Karar Ağaçları Yöntemlerini Kullanarak Bilgi Çalışanlarının Kurum Performansı Üzerine Etkisinin Ölçülmesi ve Bir Uygulama* (doktora tezi, basılmamış). İÜ, Sosyal Bilimler Enstitüsü, İstanbul.
- Lakshmanan, C. B., Srinivasan, V., Ponaraja, C., 2015. Data mining with decision tree to evaluate the pattern on effectiveness of treatment for pulmonary tuberculosis: a clustering and classification techniques. *Scientific Research Journal (SCIRJ)*, 3: 43-48.
- Larose, T. D., 2005. *Discovering Knowledge in Data; An Introduction to Data Mining*. Wiley Publication.
- Miller, B., Fridline, M., Liu, Y., P., Marino, D., 2014. *Use of CHAID Decision Trees to Formulate Pathways for the Early Detection of Metabolic Syndrome in Young Adults*. Hindawi Publishing Corporation Computational and Mathematical Methods in Medicine.
- Miner, G., Nisbet, R., Elder, J., 2009. *IV. Handbook of Statistical Analysis and Data Mining Applications*. Academic Press Elsevier.
- Mitra, S., Acharya T., 2003. *Data Mining Multimedia, Soft Computing and Bioinformatics*, Wiley Publication.
- Muenchen, A., R., 2015. *The Popularity of Data Analysis Software*. <http://r4stats.com/articles/popularity/> Erişim Tarihi: 30.11.2015.
- Murthy, K. S., 1998. *Automatic Construction of Decision Tree from Data: A Multi-Disciplinary Survey*. Kluwer Academic Publishers. Siemens Corporate Research.
- Myatt, J. G., 2007. *Making Sense of Data*. Wiley Publication.
- OECD, 2014. Data mining-related scientific articles, 1995-2014. http://www.oecd-ilibrary.org/science-and-technology/measuring-the-digital-economy/data-mining-related-scientific-articles-1995-2014_9789264221796-graph6-en. OECD online kütüphanesi. Erişim Tarihi: 21.12.2015.
- Oğuzlar, A., 2003. Veri ön işleme, *Erciyes Üniversitesi İktisadi Ve İdari Bilimler Fakültesi Dergisi*, 21: 67-76.

- Öğüdücü, G. Ş., 2008. Veri madenciliği temel sınıflandırma yöntemleri. <http://ninova.itu.edu.tr/tr/dersler/bilisim-enstitusu/195/bbl-606/ekkaynaklar?g29172> İstanbul Teknik Üniversitesi, İstanbul. Erişim Tarihi: 14.12.2015.
- Özbudak, Ö., 2009. *Yüz Resimlerinden Cinsiyet Tayini* (yüksek lisans tezi, basılmamış). İTÜ, Fen bilimleri Enstitüsü, İstanbul.
- Özekes, S., 2003. Veri madenciliği ve uygulama alanları. *İstanbul Ticaret Üniversitesi Sosyal Bilimler Dergisi*, 2 (3): 65-82.
- Özkan, Y., 2008. *Veri Madenciliği Yöntemleri*, Papatya Yayınevi.
- Pektaş, O. A., 2013. *SPSS 20 ile Veri Madenciliği*, Dikeyksen Yayınevi.
- Quinlan, J. R., 1986. Induction of decision trees. *Machine Learning* 1: 81-106.
- Rokack, L., Maimon O., 2008. *Data Mining with decision Trees Theory and Applications*. Series in Machine Perception and Artificial Intelligence, Volume 69. World Scientific Publishing.
- Shafer, J., Agrawal, R., Mehta, M., 1996. *SPRINT: A Scalable Parallel Classifier for Data Mining*. IBM Almaden Research Center.
- Shih, S., Y., Loh, Y., W., 1997. Split selection methods for classification trees. *Published in Statistica Sinica*, 7: 815-840.
- Shmueli, G., 2015. Classification trees: CART vs. CHAID. <http://www.bzst.com/2006/10/classification-trees-cart-vs-chaid.html> Erişim Tarihi: 08.03.2016
- Silahtaroglu, G., 2013. *Veri Madenciliği Kavram ve Algoritmaları*, Papatya Yayınevi.
- Soner, V., 2015. IBM SPSS Modeler 15 (SPSS Clementine) <https://volkansoner.wordpress.com/category/veri-madenciligi/> Erişim Tarihi: 30.11.2015
- Song, Y., Y., Lu, Y., 2015. Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, 27 (2): 130-135.
- Souza, J., Matwin S., Japkowicz., 2002. *Evaluation data mining Models: A Pattern Language*. School of Information Technology and Engineering. University of Ottawa.
- SPSS, 2000. *CRISP-DM 1.0. Step-by-Step Data Mining Guide*.
- SPSS, 2012. *IBM SPSS Modeler 15 Algorithms Guide*.
- SPSS, 2012. *IBM SPSS Modeler 15 Modelling Nodes*.

- SPSS, 2012. *IBM SPSS Modeler 15 Application Guide*.
- Şeker, E., S., 2013. *WEKA ile ve Veri Madenciliği*. Cinius Yayınları.
- Şenesen, Ü., 2013. *İstatistik, Sayıların arkasını Anlamak*. Literatür Yayınları.
- TDK, 2015. <http://tdk.gov.tr/> Erişim Tarihi: 21.12.2015
- Terlemez, L., 2008. *Eş İşlem Stratejisi Yöntemiyle İMKB’de Portföy Oluşturmada Veri Madenciliği Uygulaması* (doktora tezi, basılmamış). AÜ, Fen Bilimleri Enstitüsü, Eskişehir.
- Timor, M., Şimşek, U. T., 2008. Veri madenciliği sepet analizi ile tüketici davranışı modellemesi. *İstanbul Üniversitesi İşletme Fakültesi İşletme İktisadi Enstitüsü Dergisi*, 59: 3-10
- TÜİK (Türkiye İstatistik Kurumu), 2011. SPSS Modeler, Veri Madenciliği Hizmet İç Eğitim Notları. Türkiye İstatistik Kurumu, Ankara.
- TÜİK (Türkiye İstatistik Kurumu), 2013. Yaşam Memnuniyeti Araştırması. ISBN 978-975-19-6010-8, Türkiye İstatistik Kurumu Matbaası, Ankara.
- Tüzüntürk, S., 2010. Veri madenciliği ve istatistik. *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*. 1: 65-90.
- Edelstein, A. H., 1999. *Introduction to Data Mining and Knowledge Discovery*, Two Crows Corporation, Third Edition.
- Vercellis C., 2009. *Business intelligence : Data Mining and Optimization for Decision Making*. Wiley Publication.
- Witten, I.H., Frank E., Hall A. M., 2011. *Data Mining : practical machine learning tools and techniques*. Morgan Kaufmann Publishers.
- Yakut, E., 2012. *Veri Madenciliği Tekniklerinden C5.0 Algoritması Ve Destek Vektör Makineleri İle Yapay Sinir Ağlarının Sınıflandırma Başarılarının Karşılaştırılması: İmalat Sektöründe Bir Uygulama* (doktora tezi, basılmamış). AÜ, Sosyal Bilimler Enstitüsü, Erzurum.
- Zhao, C., M. ve Luan, J., 2006. Data mining going beyond traditional statistics. *New Directions for Institutional Research*, 131: 7-16.
- Zong, N. ve Zhou L., 1999. *Methodologies for Knowledge Discovery and Data Mining: Third Pasific-Asia Conference*, Pakdd-99. Beijing, China, April, 26-28, Springer Verlag.

EKLER

Ek-1: Oluşturulan Karar Ağaçlarında Kullanılan Değişkenlerin Soruları

S1	Aylık hanehalkı gelirinizden memnun musunuz?
S2	Önümüzdeki yıl, hanenizin mali durumu sizce nasıl olacak?
S3	Kendi geleceğinizden ne kadar umutlusunuz?
S4	Türkiye'de yaşayan insanların refah düzeyini "0" basamağı en düşük "10" basamağı en yüksek" düzey olarak düşündüğünüzde, kendinizi hangi düzeyde görüyorsunuz?
S5	Kendinize ayırdığınız zamandan memnun musunuz?
S6	Sağlığınızdaki memnun musunuz?
S7	Şimdiye kadar almış olduğunuz eğitimden memnun musunuz?
S8	Türkiye'nin Avrupa Birliği'ne üye olması konusunda bir referandum (halk oylaması) yapılırsa, siz ne yönde oy kullanırsınız?
S9	Evliliğinizden memnun musunuz?
S10	Son bir yılda, Daha ucuz ürünlerin tüketimine yöneldim
S11	Görüşülen kişinin cinsiyeti
S12	Evinizde yalnızken kendinizi ne kadar güvende hissediyorsunuz?
S13	Sosyal hayatınızdan (eğlence, kültürel ve sportif faaliyetler gibi) memnun musunuz?
S14	Oturduğunuz konuttan memnun musunuz?
S15	Kimin üzerinden sosyal güvenlik kapsamındasınız?

ÖZGEÇMİŞ

1985 yılı Mayıs ayında Sakarya'nın Hendek ilçesinde doğdu. İlk ve orta öğretimini Sakarya'da tamamlamıştır. 2007-2011 yılları arasında Samsun Ondokuz Mayıs Üniversitesi'nde İstatistik Bölümü'ne devam etti ve 2011 yılında mezun oldu. 2012 Aralık ayından itibaren Türkiye İstatistik Kurumu Van Bölge Müdürlüğü'nde İstatistikçi olarak görev yapmaktadır.

