VERİ MADENCİLİĞİ ALGORİTMALARI İLE AİLE YAPISI ARAŞTIRMASI VERİLERİNİN SINIFLANDIRILMASI

Yüksek Lisans Tezi

Ferdi KARAKÜTÜK

Eskişehir 2024

VERİ MADENCİLİĞİ ALGORİTMALARI İLE AİLE YAPISI ARAŞTIRMASI VERİLERİNİN SINIFLANDIRILMASI

Ferdi KARAKÜTÜK

Yüksek Lisans Tezi

İstatistik Anabilim Dalı

Danışman: Doç. Dr. Özer ÖZDEMİR

Eskişehir

Eskişehir Teknik Üniversitesi

Lisansüstü Eğitim Enstitüsü

Kasım 2023

JÜRİ VE ENSTİTÜ ONAYI

Ferdi KARAKÜTÜK'ün VERİ MADENCİLİĞİ ALGORİTMALARI İLE AİLE YAPISI ARAŞTIRMASI VERİLERİNİN SINIFLANDIRILMASI başlıklı çalışması 13/11/2023 tarihinde aşağıdaki jüri tarafından değerlendirilerek "Eskişehir Teknik Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği"nin ilgili maddeleri uyarınca, İstatistik Anabilim dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Üyeleri	**Unvan Adı Soyadı**	**İmza**
Üye	: Doç. Dr. Özer ÖZDEMİR	
Üye	: Dr. Öğr. Üyesi Gültekin ATALIK	
Üye	: Dr. Öğr. Üyesi Mustafa ÇAVUŞ	

Prof. Dr. Semra KURAMA

Lisansüstü Eğitim Enstitüsü Müdürü

13/11/2023

DANIŞMAN ONAYI

Danışmanlığını yürüttüğüm Yüksek Lisans öğrencisi Ferdi KARAKÜTÜK, VERİ MADENCİLİĞİ ALGORİTMALARI İLE AİLE YAPISI ARAŞTIRMASI VERİLERİNİN SINIFLANDIRILMASI başlıklı tez çalışmasını tamamlamıştır. Hazırlamış olduğu tez tarafımca incelenmiş ve öğrencinin tez savunma sınavına alınması bilimsel ve etik açıdan uygun görülmüştür.

Tez Danışmanı

Doç. Dr. Özer ÖZDEMİR

ÖZET

VERİ MADENCİLİĞİ ALGORİTMALARI İLE AİLE YAPISI ARAŞTIRMASI VERİLERİNİN SINIFLANDIRILMASI

Ferdi KARAKÜTÜK

İstatistik Anabilim Dalı

Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Kasım 2023

Danışman: Doç. Dr. Özer ÖZDEMİR

Madencilik terimi ile benzer anlam taşıyan veri madenciliği, sorunların çözülmesine, eğilimlerin tahmin edilmesine, risklerin azaltılmasına ve yeni fırsatlar bulunmasına yardımcı olmak için muazzam miktarda bilgi ve veri setini analiz etme, yararlı zekayı keşfetme sürecidir. Aynı zamanda veri madenciliği ilişkilerin kurulmasını, sorunlarla başa çıkabilmek için korelasyonlar bulmayı ve süreçte eyleme geçirilebilir bilgiler oluşturmayı da içermektedir. Bu tez çalışmasında veri madenciliğinin muazzam yeteneklerinden faydalanarak Likert ölçekli veri tiplerinde bilgi keşfi yapılması amaçlanmıştır. Farklı veri madenciliği tekniklerinin Likert ölçekli veri türleri üzerinde sınıflandırma başarısını karşılaştırmak üzere veri seti olarak Türkiye İstatistik Kurumu (TUİK) Başkanlığı tarafından yürütülen Türkiye Aile Yapısı Araştırması (TAYA) seçilmiştir. İki aşamada gerçekleştirilen deneylerde ilk olarak öznitelik seçimi yapılmış ve Bilgi Kazancı kriteri ile 10 değerli öznitelik belirlenmiştir. Sınıflandırma aşamasında ilk olarak veri setindeki kategori sayısı değiştirilerek algoritmaların sınıflandırma başarısı ölçümlenmiştir. Ardından yapısı gereği dengesiz olan veri seti üzerinde sınıflar arası dengesizlik giderilmiş ve sınıflama analizine etkisi gözlemlenmiştir. Dengesizlik giderilmeden yapılan sınıflandırmada beşli kategoriye sahip olan veri setinde en başarılı sınıflandırma performansı CART algoritmasında, üçlü kategoriye sahip olan veri setinde RepTree algoritmasında görülmüştür. Sınıflar arası dengesizliği giderebilmek amacıyla yeniden örnekleme ve veri tamamlama yöntemi ile toplam örnek hacmi değiştirilerek üç farklı veri seti oluşturulmuştur. Oluşturulan veri setlerinde sınıflandırma başarısı en yüksek olan algoritmanın CART algoritması olduğu görülmüştür.

Anahtar Sözcükler: Veri madenciliği, Sınıflandırma, Anket verileri, Türkiye Aile Yapısı Araştırması.

ABSTRACT

CLASSIFICATION OF FAMILY STRUCTURE RESEARCH DATA BY DATA MINING ALGORITHMS

Ferdi KARAKÜTÜK

Department of Statistic

Eskişehir Technical University, Institute of Graduate Programs, November 2023

Supervisor: Doç. Dr. Özer ÖZDEMİR

Similar to the term mining, data mining is the process of discovering useful intelligence, analyzing enormous amounts of information and datasets to help solve problems, predict trends, mitigate risks, and find new opportunities. At the same time, data mining involves building relationships, finding correlations to deal with problems, and generating actionable insights in the process. In this thesis, it is aimed to discover information in Likert scale data types by taking advantage of the enormous capabilities of data mining. To compare the classification success of different data mining techniques on Likert scale data types, Turkey Family Structure Survey (TAYA) conducted by the Turkish Statistical Institute (TURKSTAT) was chosen as the data set. In the experiments carried out in two stages, first feature selection was made, and 10 valuable features were determined with the Information Gain criterion. In the classification phase, firstly, the number of categories in the dataset was changed and the classification success of the algorithms was measured. Then, the imbalance between the classes on the dataset, which is imbalanced due to its structure, was removed and its effect on the classification analysis was observed. The most successful classification performance was observed in the CART algorithm for the dataset with five categories and in the RepTree algorithm for the dataset with three categories. To eliminate the imbalance between classes, three different data sets were created by changing the total sample volume with resampling and data completion method. It was observed that the algorithm with the highest classification success in the created data sets was the CART algorithm.

Keywords: Data mining, Classification, Survey data, Türkiye Family Structure Research.

TEŞEKKÜR

Bu çalışmanın gerçekleştirilmesinde, üç yıl boyunca değerli bilgilerini, tecrübelerini ve zamanını benimle paylaşan saygıdeğer danışman hocam; Doç. Dr. Özer ÖZDEMİR'e, çalışmam boyunca benden bir an olsun yardımlarını esirgemeyen biricik eşim Öğr. Gör. Dr. Aslı KAYA KARAKÜTÜK'e sonsuz teşekkürlerimi sunarım.

Çalışmam kapsamında kullanılan verilerin temininde yardımını esirgemeyen kurumum Türkiye İstatistik Kurumu Başkanlığına ve sabırları için TÜİK Zonguldak Bölge Müdürlüğü çalışma arkadaşlarıma teşekkürü borç bilirim.

Ferdi KARAKÜTÜK

13/11/2023

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın Eskişehir Teknik Üniversitesi tarafından kullanılan "bilimsel intihal tespit programı"yla tarandığını ve hiçbir şekilde "intihal içermediğini" beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

Ferdi KARAKÜTÜK

İÇİNDEKİLER

Sayfa

BAŞLIK SAYFASI ..... **I**

JÜRİ VE ENSTİTÜ ONAYI ..... **II**

DANIŞMAN ONAYI ..... **III**

ÖZET ..... **IV**

ABSTRACT ..... **V**

TEŞEKKÜR ..... **VI**

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ ..... **VII**

İÇİNDEKİLER ..... **VIII**

TABLOLAR DİZİNİ ..... **X**

ŞEKİLLER DİZİNİ ..... **XI**

SİMGELER VE KISALTMALAR DİZİNİ ..... **XII**

1. GİRİŞ **1**

2. LİTERATÜR TARAMASI **3**

3. VERİ MADENCİLİĞİ **7**

3.1. Veri Madenciliği Teknolojisi ve Önemi **7**

3.2. Veri Madenciliği Modelleri **9**

3.2.1. Sınıflandırma **10**

3.2.1.1. Model performans değerlendirme ölçüleri ***11***

3.2.2. Kümeleme **13**

3.2.3. Birliktelik Kuralı **14**

4. KARAR AĞAÇLARI **15**

4.1. Bölme Türleri **18**

4.1.1. Bilgi kazancı **19**

4.1.2. Gini indeksi **19**

4.2. Büyüme ve Budama **19**

4.3. Karar Ağaçları İndükleyicileri **22**

4.3.1. ID3 Algoritması 22
4.3.2. C4.5 Algoritması (J48) 23
4.3.3. CART (C&RT) Algoritması 24
4.3.3.1. Twoing kriteri..... 25
4.3.3.2. Entropi kriteri 25
4.3.4. CHAID (Ki-Kare Otomatik Algılama Dedektörü) Algoritması 26
4.3.5. NbTREE 28
4.3.6. RepTREE 29
4.3.7. RandomTREE 29

5. BULGULAR..... 30

5.1. Dengeli Olmayan Veri Seti İçin Sınıflandırma Sonuçları 33
5.1.1. Beşli Likert ölçeğe sahip değişkenler için sınıflandırma sonuçları..... 33
5.1.2. Üçlü Likert ölçeğe sahip değişkenler için sınıflandırma sonuçları..... 37
5.2. Dengeli Hale Dönüştürülen Veri Setleri İçin Sınıflandırma Sonuçları..... 44
5.2.1. En yüksek örnek sayısına sahip mutlu sınıfı örnek sayısı baz alınarak yeniden örneklenen veri setinde mutluluk düzeyi sınıflandırma sonuçları 44
5.2.2. Orta mutluluk düzeyi sınıf örnek sayısı baz alınarak yeniden örneklenen veri setinde mutluluk düzeyi sınıflandırma sonuçları 47
5.2.3. En düşük örnek sayısına sahip mutsuz sınıfı örnek sayısı baz alınarak yeniden örneklenen veri setinde mutluluk düzeyi sınıflandırma sonuçları 50

6. TARTIŞMA VE SONUÇ 54

KAYNAKÇA..... 57

ÖZGEÇMİŞ

TABLOLAR DİZİNİ

Sayfa

Tablo 3.1. Karışıklık matrisi 11
Tablo 3.2. Kappa istatistiğinin değerlerinin yorumu 13
Tablo 5.1. Değişken grubu özet bilgisi 30
Tablo 5.2. Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (beşli Likert ölçek, genel) 35
Tablo 5.3. Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (beşli Likert ölçek, sınıf bazlı) 36
Tablo 5.4. CART algoritması ile elde edilen karışıklık matrisi 37
Tablo 5.5. Revize edilmiş değişkenler 38
Tablo 5.6. Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (üçlü Likert ölçek, genel) 42
Tablo 5.7. Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (üçlü Likert ölçek, sınıf bazlı) 43
Tablo 5.8. RepTREE algoritması ile elde edilen karışıklık matrisi 44
Tablo 5.9. Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (mutlu sınıfı örnek sayısı baz alınan, genel) 46
Tablo 5.10. Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (mutlu sınıfı örnek sayısı baz alınan, sınıf bazlı) 46
Tablo 5.11. CART algoritması ile elde edilen karışıklık matrisi 47
Tablo 5.12. Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (orta sınıfı örnek sayısı baz alınan, genel) 48
Tablo 5.13. Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (orta sınıfı örnek sayısı baz alınan, sınıf bazlı) 49
Tablo 5.14. CART algoritması ile elde edilen karışıklık matrisi 49
Tablo 5.15. Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (mutsuz sınıfı örnek sayısı baz alınan, genel) 51
Tablo 5.16. Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (mutsuz sınıfı örnek sayısı baz alınan, sınıf bazlı) 52
Tablo 5.17. CART algoritması ile elde edilen karışıklık matrisi 52
Tablo 5.18. Üçlü Likert ölçekli veri setlerinde CART algoritmasına ait sınıf bazlı sonuçları 52

ŞEKİLLER DİZİNİ

Sayfa

Şekil 3.1. Veri tabanı teknolojisinin gelişimi 8
Şekil 3.2. Veri madenciliği işlevi ve ürünleri 9
Şekil 4.1. Karar ağacı örneği 16
Şekil 5.1. Beşli Likert ölçeğine sahip veri setinde mutluluk düzeyine ait sınıf frekansları 34
Şekil 5.2. Mutluluk sınıflandırması için bilgi kazancı metodu ile seçilen 10 adet öznitelik (beşli Likert ölçekli veri setinde) 35
Şekil 5.3. Üçlü Likert ölçeğine sahip veri setinde mutluluk düzeyine ait sınıf frekansları 41
Şekil 5.4. Mutluluk sınıflandırması için bilgi kazancı metodu ile seçilen 10 adet öznitelik (üçlü Likert ölçekli veri setinde) 42
Şekil 5.5. En çok örneğe sahip sınıf örnek sayısı baz alınarak elde edilen veri setine ait mutluluk düzeyi sınıf frekansları 45
Şekil 5.6. En yüksek sınıf örneğine eşitlenmiş veri setinde mutluluk sınıflandırması için bilgi kazancı metodu ile seçilen 10 adet öznitelik (üçlü Likert ölçekli) 45
Şekil 5.7. Mutluluk düzeylerinden Orta sınıfı örnek sayısı baz alınarak yeniden örneklenen veri setine ait mutluluk düzeyi frekansları 47
Şekil 5.8. Orta sınıf örneğine eşitlenmiş veri setinde mutluluk sınıflandırması için bilgi kazancı metodu ile seçilen 10 adet öznitelik (üçlü Likert ölçekli) 48
Şekil 5.9. Mutluluk düzeylerinden Mutsuz sınıfı örnek sayısı baz alınarak yeniden örneklenen veri setine ait mutluluk düzeyi frekansları 50
Şekil 5.10. Mutsuz sınıf örneğine eşitlenmiş veri setinde mutluluk sınıflandırması için bilgi kazancı metodu ile seçilen 10 adet öznitelik (üçlü Likert ölçekli) 51

SİMGELER VE KISALTMALAR DİZİNİ

AID	:	Otomatik Etkileşim Dedektörü
CART	:	Sınıflandırma ve Regresyon Ağacı
CHAID	:	Ki-Kare Otomatik Etkileşim Detektörü
DT	:	Karar Ağacı
GUI	:	Grafiksel Kullanıcı Arayüzü
Nb	:	Naive Bayes
NP	:	Deterministik Olmayan Polinom Zamanı
Rep	:	Azaltılmış Hata Budama
RepT	:	Azaltılmış Hata Budama Ağacı
TAYA	:	Türkiye Aile Yapısı Araştırması
TDIDT	:	Yukarıdan Aşağıya Karar Ağacı İndüksiyonu
THAID	:	Teta Otomatik Etkileşim Dedektörü
TUİK	:	Türkiye İstatistik Kurumu

1. GİRİŞ

Dijitalleşme çağı, sayısallaştırılmış bilgilerin işlenmesini, saklanmasını, dağıtılmasını ve iletilmesini kolaylaştırmıştır [1]. Bilgi işlem ve ilgili teknolojilerdeki önemli ilerlemeler ve bunların hayatın farklı alanlarında sürekli genişleyen kullanımları ile, farklı özelliklerde çok büyük miktarda veri toplanmaya ve veri tabanlarında saklanmaya devam edilmektedir. Bu tür verilerin depolanma hızı da paralel olarak artış göstermektedir. Devasa veri hacimlerinden bilgi keşfi gerçekten de ciddi iş ve zaman yükü doğurmaktadır. Veri madenciliği kavramı, devasa veri hacmine gömülü bilgi patlamasını anlamlandırma girişimi olarak hayatımıza girmiştir.

Veri madenciliği kavramına ilişkin literatürde birçok tanım bulunmaktadır. Bazı araştırmacılar veri madenciliğini şu şekilde tanımlamıştır: Gundecha ve Tan'a göre büyük ölçekli verilerde yararlı veya eyleme geçirilebilir bilgileri keşfetme sürecidir [2, 3]. Zaki ve Meira'ya [4] göre veri madenciliği, büyük ölçekli verilerden açıklayıcı, anlaşılır ve tahmine dayalı modellerin yanı sıra anlayışlı, ilginç ve yeni kalıpları keşfetme sürecidir [4]. Veri madenciliğinin diğer bir tanımı Özer ve Sprinkhuizen-Kuyper [5] ve Garcia v.d. [6], büyük miktarda veriden ilginç (önemsiz olmayan, üstü örtülü, önceden bilinmeyen ve potansiyel olarak yararlı) kalıpların veya bilginin çıkarılmasıdır.

Veri madenciliğinin son yıllarda bilgi endüstrisinde büyük ilgi görmesinin en büyük nedeni, büyük miktarda verinin yaygın olarak bulunması ve bu tür verileri yararlı bilgi ve bilgiye dönüştürmeye yönelik duyulan acil ihtiyaçtır [7]. Veri madenciliği alanında, işlenecek veriler çok büyük olma eğilimindedir. Bu nedenle, hesaplama açısından verimli ve ölçeklenebilir algoritmalar tasarlamak oldukça arzu edilmektedir. Depolanan büyük miktarda güncel ve geçmiş veri bulunmaktadır. Bu nedenle veri tabanları büyüdükçe karar vermeyi desteklemek giderek zorlaşmaktadır. Ek olarak, veriler birden çok kaynaktan ve birden çok alandan olabilmektedir. Bir problemin planlama ve diğer işlevlerini desteklemek için verileri analiz etmeye açık bir ihtiyaç bulunmaktadır.

Veri madenciliği teknikleri ile verilerin görselleştirilmesi, keşfedilmeyi bekleyen örüntülerin ve bilgilerin ortaya çıkarılması, veriler arasındaki ilişkinin çıkarılması ve belirlenen ilişki türüne göre tahminlerin yapılabilmesi mümkündür. Veri madenciliği tekniklerinden bazıları, diğerlerinin yanı sıra sert (crisp) kümelere, tümevarımsal mantık programlamaya, makine öğrenimine ve sinir ağlarına dayalı olanları içermektedir. Veri madenciliği problemleri, sınıflandırmayı (verileri gruplara ayırmak için kurallar bulma),

ilişkilendirme (veriler arasında ilişkilendirmeler yapmak için kurallar bulma) ve sıralamayı (verileri sıralamak için kurallar bulma) içerir.

Sınıflandırma önemli bir veri madenciliği problemidir. Bir sınıflandırma probleminde, her biri bir dizi özniteliğe sahip bir dizi örnekten oluşan eğitim seti adı verilen bir girdi veri kümesi bulunmaktadır. Nitelikler, nitelik değerleri sıralandığında süreklidir veya nitelik değerleri sıralanmadığında kategoriktir. Kategorik niteliklerden biri, sınıf etiketi veya sınıflandırma niteliği olarak adlandırılır. Amaç, modelin eğitim veri kümesinden değil yeni verileri sınıflandırmak için kullanılabileceği şekilde, diğer niteliklere dayalı olarak bir sınıf etiketi modeli oluşturmak için eğitim veri kümesini kullanmaktır [8].

Veri madenciliği teknikleri ile anket verileri- bir başka deyişle Likert ölçekli veri- üzerinde gerçekleştirilmiş çalışmaların sayısının yeterli olmaması nedeniyle ve literatüre bu bağlamda kıymetli bilgiler kazandırmak amacıyla bu çalışma gerçekleştirilmiştir. Tez çalışması kapsamında veri madenciliği algoritmalarının Üçlü ve Beşli Likert (Ordinal) ölçeğine sahip veri seti üzerindeki sınıflandırma performansını ölçümlemek, sınıflar arası dengesizliğin olduğu veri seti üzerinde sınıflandırma performanslarını karşılaştırmak ve dengesizliğin giderildiği durumda algoritmaların sınıflandırma performansının nasıl değiştiğini ortaya koymak amaçlanmıştır. Bu nedenle çalışmada, veri seti olarak Türkiye İstatistik Kurumu Başkanlığı tarafından yürütülen Türkiye Aile Yapısı Araştırması (TAYA) seçilmiştir. TAYA veri seti yapısı gereği dengesiz bir veri setidir ve anketin amacı, Türkiye'deki ailelerin yapısını, bireylerin aile ortamındaki yaşam biçimlerini ve bireylerin aile hayatına ilişkin değer yargılarını tespit etmektir. Dolayısıyla bu çalışma, aynı zamanda sınıflandırma analizi sonucunda "Türk aile yapısının güncel durumunu ve demografik çerçevesini" sunmaktadır.

Çalışmanın geriye kalan bölümünde ilk olarak gerçekleştirilen literatür taramasından örnekler sunulacaktır. Metodoloji bölümünde veri madenciliği süreci ile veri madenciliği algoritmalarından bahsedilecektir. Dördüncü bölümde veri madenciliği süreçlerinde sıklıkla kullanılan karar ağaçları algoritmalarından bahsedilecektir. Beşinci bölümde veri seti hakkında detaylı bilgi verilecektir. Bu bölümde ayrıca çeşitli veri madenciliği algoritmaları anket verileri üzerinde uygulanacak ve diğer algoritmalar ile karşılaştırılıp elde edilen sonuçlar yorumlanacaktır. Son bölümde ise elde edilen sonuçlar genel olarak tartışılacak ve değerlendirilecektir.

2. LİTERATÜR TARAMASI

Bu bölümde, tezin amacı doğrultusunda, anket verileri üzerinde çeşitli veri madenciliği algoritmalarının uygulandığı çalışmalar ve aile yapısı verilerinin kullanıldığı çalışmalar örneklendirilmiştir.

Sohn ve Moon (2004) yapmış oldukları çalışmada, veri zarflama analizi ve karar ağaçlarını kullanarak etkili teknoloji ticarileştirme projelerine yol haritası sağlamaya çalışmışlardır. Veri zarflama analizi sonuçları karar ağaçları için girdi değişkeni olarak kullanılmıştır. Bu kapsamda performans ölçütünün yanı sıra çevresel faktörleri de dikkate alan yeni bir yaklaşım önermişlerdir [9].

Koufakou ve arkadaşları (2005) yapmış oldukları çalışmada, kendi kurumlarındaki öğrenciler üzerinde yeni bir ders değerlendirme anketi hakkında bir ön çalışma yürütmüşlerdir. Likert ölçekli ve serbest metin soruları içeren anketten bilgi çıkarmak için veri madenciliği tekniklerini uygulamışlar ve böylece eğitimcilerin ve yöneticilerin öğrenci duyguları ve görüşleri hakkında fikir edinmelerine yardımcı olmuşlardır. Ayrıca, yorumlardaki önemli anahtar terimleri ve terimlerin ilişkilerini çıkarmak için çeşitli öğrenci yorumlarına birliktelik kuralı madenciliği ile metin madenciliği uygulamışlardır. Elde edilen sonuçlar öğrenci anketlerindeki açık uçlu yorumlardan bilgi çıkarmak için veri madenciliği tekniklerini kullanmanın yararlılığını göstermektedir. [10]

Brefelean (2007) çalışması, bir fakültedeki farklı uzmanlık öğrencileri (yüksek lisans, doktora vb.) üzerinde yapılan anketlerden toplanan verilerle, karar ağaçları aracılığıyla üniversite sonrası çalışmalarla eğitimlerine devam etme tercihlerini farklılaştırmak ve tahmin etmek amacıyla bir J48 algoritması analiz aracının uygulanmasını temsil etmektedir [11].

Mardikyan ve Batur (2011) yaptıkları çalışmada, iki farklı veri madenciliği tekniği olan adımsal regresyon ve karar ağaçlarını kullanarak öğretim elemanlarının öğretim performansının değerlendirilmesiyle ilişkili faktörleri araştırmışlardır. Veriler Boğaziçi Üniversitesi Yönetim Bilişim Sistemleri bölümü öğrencilerinin anket değerlendirmelerinden anonim olarak toplanmıştır. Sonuçlar, değerlendirme formundaki öğretim elemanı ile ilgili soruları özetleyen bir faktörün, öğretim elemanının çalışma durumunun, dersin iş yükünün, öğrencilerin devam durumunun ve formu dolduran öğrencilerin yüzdesinin öğretim elemanının öğretim performansının önemli boyutları olduğunu göstermektedir [12].

Esen ve Kuzey (2012) yapmış oldukları doktora tezinde, karar destek sistemleri ve veri madenciliği araçlarından olan karar ağaçları ve destek vektör makineleri (DVM) yöntemlerini 2011 yılında bilgi çalışanları üzerinde yapılan anket yolu ile elde edilmiş veriler üzerinde çalıştırmışlardır. Elde edilen ilgili veri kümesi kullanılarak DVM ve Karar ağaçları modelleri uygulanarak, modellerin performansları değişik araçlar ile kıyaslanmıştır. Analiz sonuçlarına göre, DVM ile C&R Tree karar ağacı modelli en yüksek doğruluk oranına sahip olduğu görülmüştür [13].

Fokoué ve Gündüz (2013) yapmış oldukları çalışmada, destek vektör makineleri, sınıflandırma ve regresyon ağaçları, boosting, rastgele orman, faktör analizi, k-means kümeleme ve hiyerarşik kümeleme gibi bazı son teknoloji istatistiksel veri madenciliği tekniklerini kullanarak Öğrenciler, eğitmenlerinin öğretme becerilerini/yeteneklerini geliştirmek için faydalı ve güvenilir geri bildirim sağlayacak kadar olgun ve bilgili midir? Dersin zorluk seviyesi ile öğrencinin eğitmene verdiği not arasında güçlü bir ilişki var mıdır? gibi soruları yanıtlamaya çalışmışlardır. Verilerin çeşitli yönlerini hem denetimli hem de denetimsiz öğrenme perspektifinden keşfetmişlerdir. Analiz sonuçları, öğrencilerin ciddiyeti ve eğitime bağlılığı (devamlılıkla ölçülen) ile eğitmenlerine verme eğiliminde oldukları puanlar arasındaki güçlü bir ilişki olduğunu ortaya çıkarmaktadır [14].

Alhendawi ve Baharudin (2014) yapmış oldukları çalışmada Bilgi Sistemi etkinliğinin değerlendirilmesinde, özellikle sınıflandırma yönteminde veri madenciliği tekniklerini kullanmayı amaçlamışlardır. Beş kalite faktörü (sistem kalitesi, bilgi kalitesi, hizmet kalitesi, kullanıcı arayüzü kalitesi ve iletişim kalitesi) ve kullanıcı memnuniyeti olmak üzere altı boyuttan oluşan bir anket kullanılarak 255 denekten oluşan makul bir veri seti toplanmıştır. Çalışmada sonuçlar, ağaç sınıflandırma algoritması J48'in iki değerin denetlenen hedefi durumunda sınıflandırma yapmada en iyisi olduğunu göstermektedir. Sonuçların regresyon analizi ile tutarlı olduğunu ve ilgili ampirik çalışmalara katkı sağlayabileceği belirtilmiştir [15].

Çalış ve arkadaşları (2014) yapmış oldukları çalışmada bilgisayar ve internet güvenliği üzerine anket düzenlemiştir. Çalışmada karar ağaçları kullanılarak farklı demografik özellikteki kişiler için çıkarım yapılması hedeflenmiş ve bu bağlamda SPSS Clementine'de C5.0, C&RT, CHAID ve QUEST algoritmaları uygulanmış ve her bir değişken için algoritmaların doğruluk oranları hesaplanmıştır. Ardından en yüksek

doğruluk oranını veren algoritma ile karar ağaçları oluşturularak sonuçlar yorumlanmıştır [16].

Şehribanoğlu ve Saygın (2016) yapmış oldukları yüksek lisans tezinde, veri madenciliği süreci, yöntemleri ve sınıflandırma yöntemleri altında yer alan karar ağaçları algoritmalarını TÜİK tarafından yürütülen Yaşam Memnuniyeti Araştırması B grubu mikro verileri kullanarak incelemişlerdir. CHAID algoritmasının daha fazla açıklayıcı değişken ile karar ağacı oluşturmasından dolayı CHAID algoritması daha araştırmacı yapıya sahip olduğu gözlemlenmiştir [17].

Maroño ve arkadaşları (2017) yapmış oldukları çalışmada, anket verilerinden öğrenilen karar ağaçlarının etmenler için davranışsal modeller olarak kullanımını yansıtmayı hedeflemişlerdir. Bir başka deyişle, aracı tabanlı bir modelde karar vermeyi uygulamak amacıyla ampirik olarak türetilen karar ağaçlarının inşası incelenmiştir. Ağacı elde etmek için doğrudan C4.5 algoritmasını uygulanmış çalışmada ortaya çıkan ağaçların daha düşük genelleme yetenekleri ve yüksek sayıda düğüm ve dal sergiledikleri ve bunların yorumlanmasının zorlaştırdığı gösterilmiştir [18].

Bajdor ve Paweloszek (2020) yapmış oldukları çalışmada, işletmeler arasındaki benzerlikleri araştırmaya ve sürdürülebilir girişimcilik temelinde belirli karakteristik davranış kalıplarını belirlemeye odaklanmış ve bu kapsamda veri toplama aracı olarak Likert ölçekli bir anketten yararlanılmıştır. Veri setindeki çok sayıda nesne ve özellik nedeniyle Orange Veri Madenciliğini kullanılmıştır. Elde edilen sonuçlar, sürdürülebilir girişimcilik kavramının varsayımlarının uygulanmasında stratejilerini geliştirme yönlerini göstermektedir [19].

Koçak (2020) yapmış olduğu çalışmada veri madenciliği teknikleri kullanılarak örgütsel bağlılığın tahmin edilmesini amaçlamaktadır. 2019 yılında psikolojik sözleşme ihlal algısı ve örgütsel güvenin örgütsel bağlılık üzerindeki etkisini belirlemeye yönelik olarak yapılan bir araştırmanın anket yöntemi ile elde edilen verileri üzerinde CART karar ağacı algoritması uygulanarak örgütsel bağlılık verileri üzerinde sınıflandırma yapılmıştır. Karar ağaçlarının örgütsel bağlılığın tahmin edilmesinde yüksek oranda başarı sağladığı görülmüştür [20].

Beernaert (2021) yapmış olduğu tez çalışmasında, anket verilerinin analizini genişletmeyi sağlayan bazı teknikler denemiştir. Özellikle, daha sağlam olan ve anket verilerinin analizine olanak tanıyan bazı veri madenciliği ve makine öğrenimi teknikleri doğrusal olmayanlar için önerilmiştir. Çalışmada kullanılan veriler, 1010 Slovakyalı'nın

müzik ve film tercihlerinden fobilere, hayata bakış açılarından harcama alışkanlıklarına kadar hayatın birçok farklı yönüyle ilgili 150 soruyu yanıtladığı Genç İnsanlar Anketi'dir. Herhangi bir boyut indirgemesi olmaksızın Extreme Gradient Boosting algoritmasının en iyi performansı gösterdiği ve onu Rastgele Orman algoritmasının izlediği ortaya çıkmıştır. Destek Vektör Makineleri kullanılması durumunda PCA hariç, boyut azaltma yöntemlerinin burada tahmin performansı üzerinde hiçbir faydası olmadığı kanıtlanmıştır [21].

2006 yılında TÜİK tarafından 18 ve daha yukarı yaştaki bireylerle yüz yüze görüşme tekniğiyle Aile Yapısı Araştırması ilk kez gerçekleştirilmiştir. 2011 ve 2016 yıllarında tekrarlanan araştırma, betimsel ve çeşitli ileri istatistiksel teknikler (çoklu regresyon, lojistik regresyon vb.) aracılığı ile anlamlandırılmaya çalışılmıştır [22]. Ancak, yapılan tarama sürecinde ilgili araştırma üzerinde veri madenciliği teknikleri ile çalışılmadığı görülmüştür. Bu husus bu tez çalışmasının özgünlüğünü oluşturmaktadır.

3. VERİ MADENCİLİĞİ

3.1. Veri Madenciliği Teknolojisi ve Önemi

Güçlü veri analiz araçlarına duyulan ihtiyaçla birleşen veri bolluğu, tarih açısından zengin ancak bilgi yönünden fakir bir durum olarak görülmektedir. Büyük ve sayısız veri tabanlarında toplanan ve depolanan, hızla büyüyen, muazzam miktardaki veri, güçlü araçlar olmadan insan kavrayışını çok aşmıştır. Buna bağlı olarak, büyük veri tabanlarında toplanan veriler, nadiren yeniden ziyaret edilen veri arşivleri olan veri yığınları haline gelmiştir. Sonuç olarak, önemli kararlar genellikle veri tabanlarında depolanan bilgi açısından zengin verilere değil, karar vericinin sezgisine dayalı olarak alınır, çünkü karar vericinin çok büyük miktardaki verilerin içine gömülü değerli bilgileri çıkaracak araçları yoktur. Bununla birlikte, bilgileri bilgi tabanına manuel girmek ne yazık ki, önyargılara ve hatalara eğilimlidir ve son derece zaman alıcı ve maliyetlidir. Veri analizi yapan veri madenciliği araçları, iş stratejilerine, bilgi tabanlarına ve bilimsel araştırmalara büyük ölçüde katkıda bulunan önemli veri modellerini ortaya çıkarmaktadır. Veri ve bilgi arasındaki genişleyen boşluk, veri mezarlarını bilginin "altın külçelerine" dönüştürecek veri madenciliği araçlarının sistematik bir şekilde geliştirilmesini gerektirmektedir.

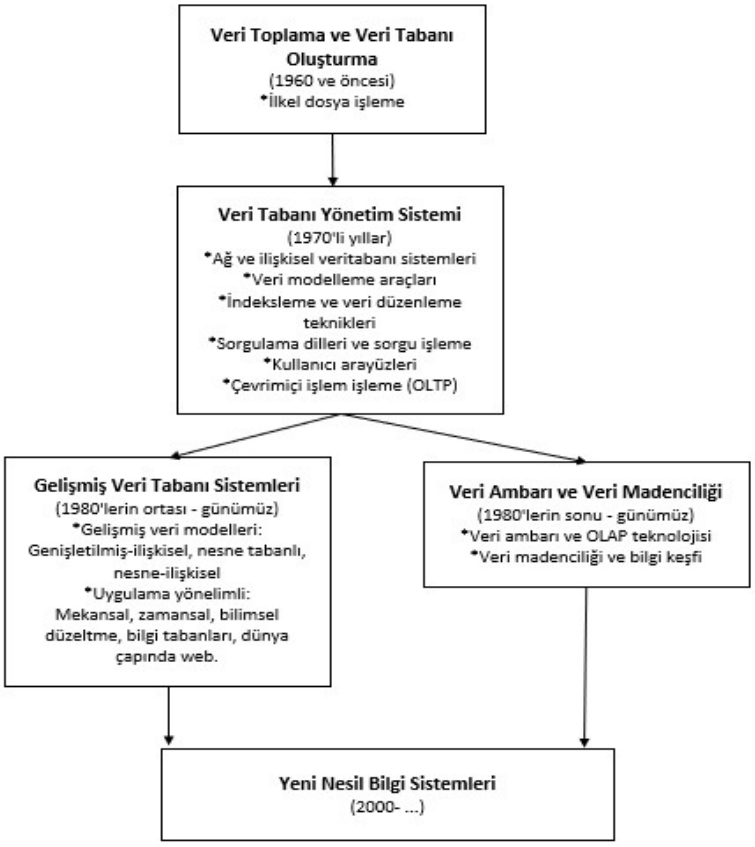
Basitçe ifade etmek gerekirse, veri madenciliği, büyük miktarda veriden bilgi çıkarma veya madencilik yapma anlamına gelmektedir. Veri madenciliği ifadesi tam olarak doğru isimlendirilememektedir. Kayalardan veya kumdan altın madenciliği yapıldığında, kaya veya kum madenciliği yerine altın madenciliği olarak adlandırılmaktadır. Bu nedenle, "veri madenciliği" biraz uzun olan "büyük miktarda veriden yani bilgi dağından madencilik yapılarak elde edilen bilgi" olarak daha uygun bir şekilde adlandırılmalıydı. Yine de madencilik, büyük miktarda ham maddeden küçük bir set değerli külçe bulma sürecini karakterize eden canlı bir terimdir. Böylece, hem "veri" hem de "madencilik" içeren böyle bir yanlış isim popüler bir seçim haline geldi. Veri madenciliğine benzer veya biraz farklı anlam taşıyan, veri tabanlarından bilgi madenciliği, bilgi çıkarımı, veri/örgü analizi, veri arkeolojisi ve veri tarama gibi birçok başka terim vardır.

Veri madenciliği, çeşitli sorgular oluşturma ve muhtemelen veri tabanlarında depolanan büyük miktarlardaki verilerden genellikle daha önce bilinmeyen faydalı bilgiler, modeller ve eğilimleri çıkarma işlemidir. Edinilen bilgi ve birikim, işletme

yönetimi, üretim kontrolü ve pazar analizinden mühendislik tasarımına ve bilim keşfine kadar uzanan uygulamalar için kullanılmaktadır.

Veri madenciliği, bilgi teknolojisinin doğal evriminin bir sonucudur. Veri tabanı endüstrisinde veri toplama ve veri tabanı oluşturma, veri yönetimi, veri analizi ve anlama (veri depolama) veri madenciliği işlevlerinin geliştirilmesinde evrimsel bir yola tanık olmuştur. Örneğin, veri toplama ve veri tabanı oluşturma mekanizmalarının erken gelişimi, veri depolama, alma ve sorgulama, işleme için etkili mekanizmaların daha sonra geliştirilmesi için bir ön koşul olarak hizmet etmiştir. Yaygın uygulama olarak sorgu ve işlem işleme sunan çok sayıda veri tabanı sistemiyle, veri analizi ve anlaşılması doğal olarak bir sonraki hedef haline gelmiştir.

1960'lardan bu yana, veri tabanı ve bilgi teknolojisi sistematik olarak ilkel dosya işleme sistemlerinden gelişmiş ve güçlü veri tabanı sistemlerine doğru gelişmektedir. 1960'lı yıllardan itibaren veri tabanı teknolojisinin gelişimi Şekil 3.1 ile gösterilmiştir.



Şekil 3.1. *Veri tabanı teknolojisinin gelişimi*

Son otuz yılda bilgisayar donanımı teknolojisindeki istikrarlı ve şaşırtıcı ilerleme, güçlü, uygun fiyatlı ve büyük miktarda bilgisayar, veri toplama ekipmanı ve depolama ortamına yol açmıştır. Bu teknoloji, veri tabanı ve bilgi endüstrisine büyük bir destek

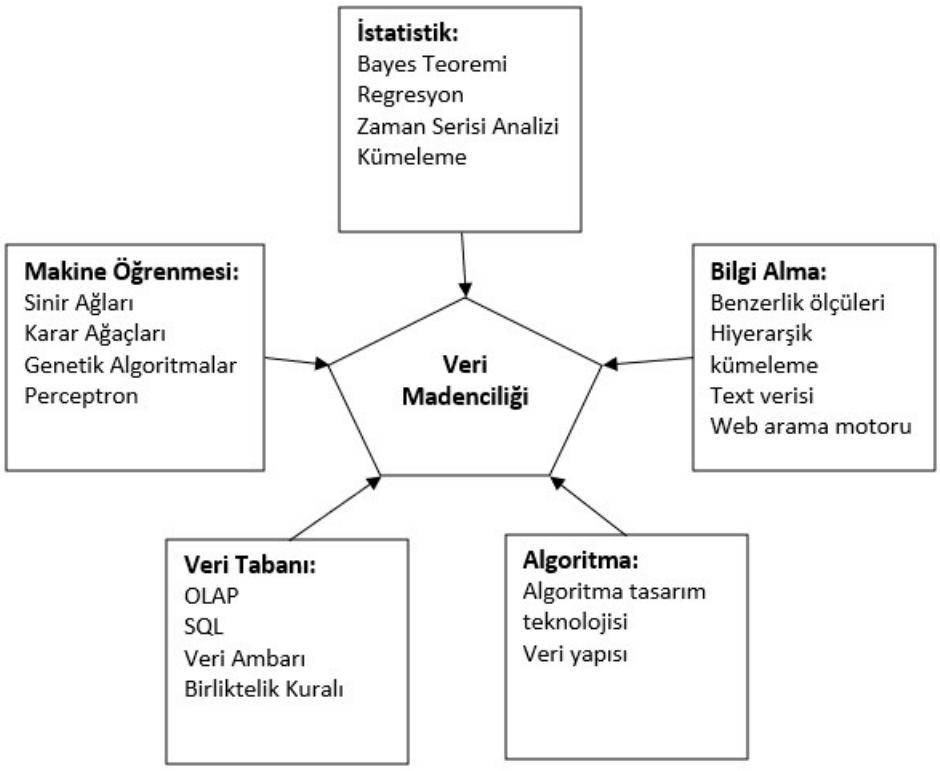
sağlamaktadır ve işlem yönetimi, bilgi alma ve veri analizi için çok sayıda veri tabanı ve bilgi havuzu sağlamaktadır.

Veriler artık birçok farklı veri tabanı türünde saklanabilmektedir. Son zamanlarda ortaya çıkan bir veri tabanı mimarisi, yönetimin karar vermesini kolaylaştırmak için tek bir sitede birleşik bir şema altında düzenlenen, birden çok heterojen veri kaynağının bir havuzu olan veri ambarıdır. Veri ambarı teknolojisi, veri temizleme, veri entegrasyonu ve çevrimiçi analitik işlemeyi, konsolidasyon ve toplama gibi işlevlere sahip analiz tekniklerini ve ayrıca bilgileri farklı açılardan görüntüleme becerisini içermektedir.

3.2. Veri Madenciliği Modelleri

Veri madenciliği, istenen sonucu elde etmek için gerçekleştirilen bir görevdir. Yorumlama/değerlendirme aşaması, son derece önemli olan veri madenciliği sonuçlarının kullanıcılara nasıl sunulduğudur çünkü sonucun kullanışlılığı buna bağlıdır. Bu adımda çeşitli görselleştirme ve GUI stratejileri kullanılır. Farklı bilgi türleri, sınıflandırma, kümeleme, birliktelik kuralı gibi farklı türde temsiller gerektirmektedir.

Veri madenciliği işlevi ve ürünleri, Şekil 3.2 ile gösterildiği gibi veri tabanı, bilgi alma, istatistik, algoritma ve makine öğrenimini içermektedir.



Şekil 3.2. *Veri madenciliği işlevi ve ürünleri*

Fonksiyonlarına göre veri madenciliği Sınıflandırma (Classification), Kümeleme (Clustering) ve Birliktelik Kuralları (Association Rules) şeklinde üç ana başlık altında toplanmaktadır [23]; Bu modellerin her biri için geliştirilmiş teknik ve algoritmalar vardır [30].

3.2.1. Sınıflandırma

En ünlü veri madenciliği modellerinden biri olan sınıflandırma, dünya hakkındaki bilgilerimizi düzenlemek ve uygulamak için kullanılan temel bilişsel süreçlerden biridir. Bir başka ifadeyle sınıflandırma, etiketi bilinmeyen nesnelerin sınıfını tahmin etmek için kullanabilmek amacıyla veri sınıflarını veya kavramlarını tanımlayan ve ayırt eden bir dizi model (veya işlev) bulma sürecidir. Hem günlük yaşamda hem de bilimsel olan/olmayan çalışmaların önceden tanımlanmış bir dizi anlamlı sınıf veya kategoride sınıflandırmak gündelik yaşamın planlı hale gelmesine olanak sağlamaktadır. Bu nedenle, mevcut verileri analiz ederek sınıflandırma modelleri oluşturmanın, bir alanda incelenen diğer tüm görevlerden daha fazla araştırma ilgisi çeken ve daha fazla uygulama bulan merkezi veri madenciliği görevlerinden biri olması şaşırtıcı değildir.

Sınıflandırmanın görevi, belirli bir alandan, bir dizi ayrık veya sürekli değerli öznitelik tarafından açıklanan örnekleri, hedef kavram olarak da adlandırılan, seçilmiş bir ayrık hedef özniteliğin değerleri olarak kabul edilebilecek bir sınıflar kümesine atamaktan oluşmaktadır. Bir başka ifadeyle sınıflandırma denetimli öğrenme görevidir. Doğru sınıf etiketleri genellikle bilinmez, ancak etki alanının bir alt kümesi için sağlanır. Aynı etki alanından, aynı öznitelik kümesi tarafından tanımlanan herhangi bir olası örneği sınıflandırmak için gereken bilginin makine dostu bir temsili olan sınıflandırma modelini oluşturmak için kullanılabilmektedir. Bu, sınıflandırma görevinin en yaygın örnekleme olduğu tümevarımsal öğrenmenin genel varsayımlarını takip etmektedir. Sınıf etiketlerinin varsayılan genel olarak mevcut olmaması, ancak alanın belirli bir alt kümesi için mevcut olmaları ilk başta tutarsız görünebilir, ancak tüm veri madenciliği yöntemlerinin dayandığı tümevarımsal çıkarım fikri için esastır.

Sınıflandırmada temel iki adım bulunmaktadır:

Model yapımı: Önceden belirlenmiş sınıflardan oluşmaktadır. Her öğenin önceden tanımlanmış bir sınıfa ait olduğu varsayılmaktadır. Model oluşturmak için kullanılan demet setine, eğitim seti ifadesi kullanılmaktadır. Model, sınıflandırma kuralları, karar ağaçları veya matematiksel formüller olarak temsil edilmektedir [31].

Model kullanımı: Bu model gelecekteki veya bilinmeyen nesneleri sınıflandırmak için kullanılır. Test örneğinin bilinen etiketi, modelin sınıflandırılmış sonucuyla karşılaştırılmaktadır. Doğruluk oranı, model tarafından doğru şekilde sınıflandırılan test seti örneklerinin yüzdesidir. Test seti, eğitim setinden farklıdır, aksi takdirde aşırı uyum meydana gelmektedir [31].

Ek olarak, karar ağaçları, yapay sinir ağları, sezgisel algoritmalar, bayesian sınıflandırma, k- en yakın komşuluk algoritması teknikler sınıflandırma modellerinde sıklıkla kullanılan tekniklerden bazılarıdır.

3.2.1.1. Model performans değerlendirme ölçüleri

Bu bölümde, sınıflandırma modellerinin performansının karşılaştırılmasında sıklıkla kullanılan ölçütler detaylı olarak incelenmiştir.

Karışıklık Matrisi

Karışıklık matrisi, makine öğreniminde tahmine dayalı analiz için bir araçtır. Sınıflandırmaya dayalı bir makine öğrenimi modelinin performansını kontrol etmek için karışıklık matrisi kullanılmaktadır. Karışıklık matrisi, her bir sınıf için kaç tane örnek atandığını gösterir [24]. Karışıklık matrisinin diğer adı hata matrisidir.

Karışıklık matrisi benzer bir manada, ikili sınıflandırma görevleri için bir sınıflandırıcı tarafından üretilen doğru ve yanlış tahminlerin sayısının konsolide edilmiş bir tablosudur. Karışıklık matrisi, bir sınıflandırma modelinin performansını değerlendirmek için kullanılan bir NxN boyutlu matrisidir; burada N, hedef sınıfların toplam sayısıdır [25]. İki sınıflı bir sınıflandırıcı için karışıklık matrisi Tablo 3.1 ile gösterilmektedir.

Tablo 3.1. *Karışıklık matrisi*

		Tahmin Edilmiş Sınıf	
		Pozitif	Negatif
Gerçek Sınıf	Pozitif	TP	FN
	Negatif	FP	TN

Tablo 3.1 incelendiğinde, sınıflandırma problemi için 4 değere sahip $2\ x\ 2$'lik bir matris oluştuğu görülmektedir. Tablo 3.1'de TP gösterimi doğru pozitif ifadesine, TN gösterimi doğru negatif ifadesine, FP gösterimi yanlış pozitif ifadesine ve FN gösterimi yanlış negatif ifadesine karşılık gelmektedir. Doğru Pozitif (TP) ifadesinde, tahmin edilen değer gerçek değerle eşleşmekte veya tahmin edilen sınıf gerçek sınıfla eşleşmektedir.

Doğru Negatif (TN) ifadesinde, tahmin edilen değer gerçek değerle eşleşir. Bir başka deyişle, gerçek değer negatif ise model negatif bir değer öngörmektedir. Yanlış Pozitif (FP) ifadesi birinci tip hata, Yanlış Negatif (FN) ifadesi ise ikinci tip hata olarak da bilinir.

Doğruluk Değeri

Hata matrisinden elde edilen en popüler ölçümlerden biri genel doğruluktur. Genel doğruluk, doğru şekilde sınıflandırılan vakaların yüzdesini değerlendirir, dolayısıyla yorumu doğrudandır. Doğru tahminlerin sayısının toplam tahmin sayısına oranıdır. Doğruluk değeri, denklem 3.1 ile hesaplanmaktadır:

$$Doğruluk \quad = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

Kesinlik

Kesinlik ölçüsü, doğru tahmin edilen gerçek gözlemlerin ölçüsüdür, bir başka ifadeyle pozitif sınıftan kaç gözlemin aslında pozitif olarak tahmin edildiğidir. Hassasiyet olarak da bilinir. Kesinlik ölçüsü, doğru sınıflandırılmış pozitif sınıfların toplam sayısının, pozitif sınıfların toplam sayısına bölünmesiyle elde edilen oran olarak hesaplanmaktadır ve denklem 3.2 ile gösterilmektedir:

$$Kesinlik = \frac{TP}{TP + FP} \tag{3.2}$$

Kesinlik, "Yanlış Negatifin, Yanlış Pozitifi gölgede bıraktığı durumlarda hatırlama yararlı bir ölçümdür" ve "İstediğimi ne ölçüde alırım?" sorusunu cevaplamaktadır [26].

F-Ölçütü

F ölçütü, 0 (neredeyse tüm ilgili öğeler alındığında) ile 1 (hemen hemen hiçbir ilgili öğe alınmadığında) arasında bir sayıdır ve kesinlik ile anmanın harmonik ortalamasıdır [27]. F ölçütü, sınıflandırıcı için hassasiyet ve anmanın arasında bir nevi denge sağlamaktadır.

$$F - \text{Ölçütü} = \frac{2 * Anma * Kesinlik}{Anma + Kesinlik} \tag{3.3}$$

Kappa İstatistiği

Cohen'e (1960) bağlı Kappa istatistiği, sınıf dengesizliği altında sınıflandırma doğruluğunu kıyaslamak için popüler bir ölçüdür ve akış veri sınıflandırmasının yanı sıra statik sınıflandırma senaryolarında da kullanılmaktadır. Tahmin edilen ve gözlenen değerler arasındaki tutarlılık derecesini ölçen Kappa istatistiği, sonucun gerçekten doğru

bir sonuç olup olmadığını veya tesadüfen meydana gelip gelmediğini görmek için modeli değerlendirmektedir [28]. Kappa istatistiği denklem 3.4 ile hesaplanmaktadır:

$$\kappa = \frac{p - p_{ran}}{1 - p_{ran}} \tag{3.4}$$

Denklem 3.4'te p, söz konusu sınıflandırıcının (referans sınıflandırıcı) doğruluğu ve p_{ran}, Rastgele sınıflandırıcının doğruluğudur. Elde edilen κ istatistiğinin değeri Tablo 3.2 ile anlamlandırılır.

Tablo 3.2. *Kappa istatistiğinin değerlerinin yorumu*

< 0	Hiç uyum olmaması
$0{,}01 - 0{,}20$	Önemsiz derecede uyum
$0{,}21 - 0{,}40$	Zayıf derecede uyum
$0{,}41 - 0{,}60$	Orta derecede uyum
$0{,}61 - 0{,}80$	İyi derecede uyum
$0{,}81 - 1{,}00$	Çok iyi derecede uyum

ROC Eğrisi

ROC eğrisi sınıflandırma problemleri için çok önemli bir performans ölçümüdür. ROC bir olasılık eğrisidir ve altında kalan alan olan AUC ayrılabilirliğin derecesini veya ölçüsünü temsil etmektedir. Eğrinin altında kalan arttıkça sınıflar arasında ayırt etme performansı artmaktadır.

ROC eğrisi; testin ayırt etme gücünün belirlenmesine, çeşitli testlerin etkinliklerinin kıyaslanmasına, uygun pozitiflik eşiğinin belirlenmesine, laboratuvar sonuçlarının kalitesinin izlenmesine, uygulayıcının gelişiminin izlenmesine ve farklı uygulayıcıların tanı etkinliklerinin kıyaslanmasına olanak sağlamaktadır [29].

3.2.2. Kümeleme

Kümeleme, aynı aileden gelen sınıflandırma ve regresyon görevlerinden tahmin edilecek önceden belirlenmiş bir hedef özelliğinin olmamasıyla ayrılan tümevarımsal bir denetimsiz bir öğrenme görevidir. Verilerde tanımlanan benzerlik örüntülerine dayanan, önceden tanımlanmış sınıflardan ziyade otonom olarak keşfedilen sınıflandırmalar olarak düşünülebilir [32].

Kümeleme görevi, belirli bir alandan, bir dizi ayrık veya sürekli değerli öznitelik tarafından tanımlanan bir örnek kümesini, benzerliklerine dayalı olarak bir kümeye ayırmaktan ve bu kümelere aynı etki alanından keyfi örnekleri eşleyebilen bir model

oluşturmaktır. Bu durum, küme oluşumu ve küme modelleme gibi iki alt görevin üst üste binmesi olarak düşünülebilir. “Küme oluşumu” analiz edilen verilerde benzerlik temelli grupların belirlenmesi süreci iken ikinci alt görev verilerin üyelik değerlerini tahminlemek için bir modelin oluşturması aşamasıdır. Bu iki alt görevi ayırmamak genellikle daha uygundur ve çoğu kümeleme algoritması hem küme oluşumunu hem de küme modellemeyi ele almaktadır. Kümeleri tanımlamak için kullanılan kriterlerin daha sonra küme üyelik tahmini için yeniden kullanılmasını mümkün kılmaktadırlar.

Literatürde kümeleme analizi için birçok algoritma öne sürülmüştür. Algoritmalar, veri türüne ve kümelerin oluşturulma şekline göre geleneksel (sert/crisp) ve bulanık kümeleme olarak iki alt kategoride incelenmektedir. Geleneksel kümeleme algoritmaları genel olarak hiyerarşik ve hiyerarşik olmayan olarak ikiye ayrılmaktadır. Bulanık kümeleme teknikleri ise geleneksel bulanık kümeleme teknikleri ve prototipi farklı geometrik şekle sahip kümeleme teknikleri olarak alt dallara ayrılmaktadır [33].

3.2.3. Birliktelik Kuralı

Birliktelik analizi (İlişkilendirme), belirli bir veri kümesinde sıklıkla birlikte ortaya çıkan nitelik-değer koşullarını gösteren birliktelik kurallarının keşfidir. Birliktelik kuralları, veri öğeleri arasındaki ilişkiyi göstermek için kullanılmaktadır. Madencilik ilişkilendirme kuralları, formun kurallarının bulunmasına izin verir: Öncül ise, o zaman (muhtemel) sonuç, burada öncül ve sonuç, bir veya daha fazla öğenin kümeleri olan öğe kümeleridir. Birliktelik analizi, pazar sepeti veya işlem verisi analizi için yaygın olarak kullanılmaktadır.

Birliktelik kuralı oluşturma genellikle iki ayrı adıma ayrılır: İlk olarak, bir veri tabanındaki tüm sık kullanılan öğe kümelerini bulmak için minimum destek uygulanır. İkincisi, bu sık öğe kümeleri ve minimum güven kısıtlaması, kuralları oluşturmak için kullanılır. Çok düşük olasılığa sahip olan bir kuralın tesadüfen oluşmasını önlediği için destek önemlidir. Güven ise bir kural tarafından yapılan çıkarımın güvenilirliğini ölçmektedir [34].

Destek ve güven, birliktelik kuralının kalitesini ölçmek için kullanılan normal yöntemdir. İlişkilendirme kuralı $X \rightarrow Y$ için destek, $X \cup Y$ içeren veritabanındaki işlemin yüzdesidir. İlişkilendirme kuralı için güven $X \rightarrow Y$, $X \cup Y$ içeren işlem sayısının X içeren işlem sayısına oranı şeklinde hesaplanmaktadır.

4. KARAR AĞAÇLARI

Basit bir ifade ile karar ağacı, denetimli sınıflandırma problemlerinde yaygın olarak kullanılan önceden tanımlanmış bir hedef değişkene sahip bir öğrenme algoritması türü olarak tanımlanabilir. Sınıflandırma problemlerine mükemmel bir şekilde uydukları için karar ağaçları güçlü yön olarak kabul edilmektedir [35].

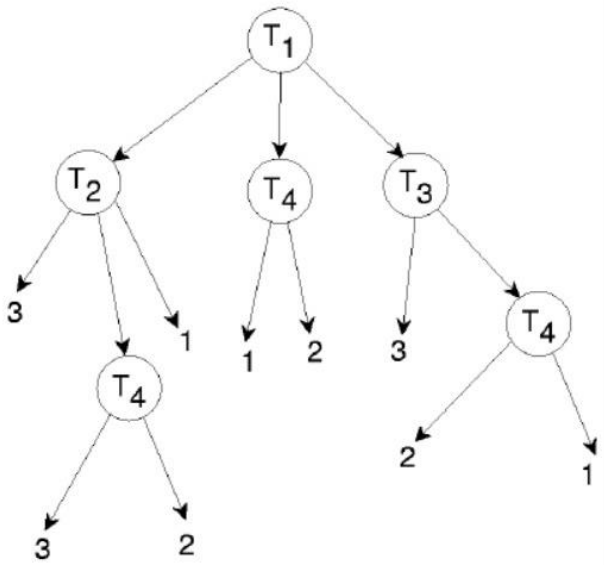
Birçok uygulamada, örnekleri doğru bir şekilde sınıflandırmak için yalnızca oluşturulan sınıflandırma modelini kullanmak yetmez, aynı zamanda modelin incelenmesi de istenebilir. Bu durum tahminlerin açıklanması, değiştirilmesi veya mevcut bazı arka plan bilgileriyle birleştirilmesi ile gerçekleşmektedir. Modelin hem yüksek sınıflandırma doğruluğunun hem de insanlar tarafından okunabilirliğinin gerekli olduğu bu tür uygulamalarda, çoğu veri madencisinin tercih edeceği bariz yöntem karar ağaçları olmaktadır. Karar ağacı algoritmaları uzun yıllardır incelenmektedir ve özellikle çok sayıda iyileştirme ve varyasyonun önerildiği veri madenciliği algoritmalarına sahiptir.

Bu nedenle, aynı model temsilini ve algoritma işlem şemalarını paylaşan, ancak birkaç ayrıntıda farklılık gösterebilen bir algoritma ailesinden bahsedilebilir. Bu çeşitlilik için alan, genellikle karar ağacı modelleri oluşturmak için gerçekleştirilen, karar ağacı büyütme ve budamadan (pruning) oluşan iki aşamalı işlemle artırılmaktadır.

Karar ağacı, bir sınıflandırma modelini temsil eden hiyerarşik bir yapıdır. İç ağaç düğümleri, etki alanını bölgelere ayırmak için uygulanan bölmelere karşılık gelir ve uç düğümler, yeterince küçük veya yeterince tek biçimli olduğuna inanılan bölgelere sınıf etiketlerini atamaktadır [30].

Karar ağaçlarının farklılık gösterebileceği çeşitli boyutlar vardır:

- Test, çok değişkenli (bir kerede girdinin birkaç özelliğini test etme) veya tek değişkenli (özelliklerden yalnızca birini test etme) olabilir.
- Testin iki veya ikiden fazla sonucu olabilir. (Eğer tüm testlerin iki sonucu varsa, bir ikili karar ağacımız var demektir.)
- Özellik veya nitelikler kategorik veya sayısal olabilir. (Binary değerli olanlar herhangi biri olarak kabul edilebilir.)
- İki veya ikiden fazla sınıf olabilir. İki sınıf ve ikili girdi varsa, ağaç bir Boolean işlevi uygular ve buna Boolean karar ağacı denir.



Şekil 4.1. *Karar ağacı örneği*

Şekil 4.1'de bir karar ağacı örneği bulunmaktadır. Şekil 4.1 incelendiğinde T1 kök, T2, T3 ve T4 düğüm olarak adlandırılmaktadır. Bir karar ağacı modeli ağaçtaki testler aracılığıyla aşağı doğru filtreleyerek bir girdi modeline bir sınıf numarası atamaktadır. Her testin birbirini dışlayan ve kapsamlı sonuçları bulunmaktadır. Örneğin, Şekil 4.1 ile gösterilen ağaç yapısındaki T_2 testinin ağaç sonuçları vardır; en soldaki, giriş modelini sınıf 3'e atamakta, ortadaki, giriş modelini T_4'ü test etmek için aşağı göndermekte ve en sağdaki, deseni sınıf 1'e atamaktadır.

Karar ağaçlarının görsel yapısı yapıların yorumlamasını ve anlaşılır olmasını sağlamaktadır. Bu duruma benzer olarak karar ağaçlarına özgü avantajlar da mevcuttur. Örneğin;

- Karar ağaçları diğer istatistiksel testler gibi varsayımlara bağlı çalışmamaktadır. Yani başka bir ifade ile karar ağaçları parametrik olmayan yapıdadır.
- Ağaç yapıları çoklu çıktı sorunlarını çözebilmektedir.
- İstatistiksel testler kullanarak bir modeli doğrulamak mümkündür. Bu, modelin güvenilirliğini açıklamayı mümkün kılmaktadır.
- Beyaz kutu modeli kullanır. Belirli bir durum bir modelde gözlemlenebilirse, durumun açıklaması Boolen mantığı ile kolayca

açıklanır. Buna karşılık, bir kara kutu modelinde (örneğin, bir yapay sinir ağında), sonuçların yorumlanması daha zor olabilir [36], [37].

Paralel olarak karar ağaçlarının da tıpkı diğer yöntemler gibi dezavantajları bulunmaktadır. Örneğin;

- Karar ağacı öğrenenler, verileri iyi genellemeyen aşırı karmaşık ağaçlar oluşturabilirler. Buna fazla öğrenme (overfitting) denir. Bu sorunu önlemek için budama, bir yaprak düğümünde gereken minimum örnek sayısını ayarlama veya ağacın maksimum derinliğini ayarlama gibi mekanizmalar gereklidir.
- Verilerdeki küçük değişiklikler tamamen farklı bir ağacın üretilmesine neden olabileceğinden karar ağaçları kararsız olabilmektedir.
- Karar ağaçlarının tahminleri, Şekil 4.1'de görüldüğü gibi ne pürüzsüz ne de süreklidir, ancak parçalı sabit yaklaşımlardır. Bu nedenle, ekstrapolasyonda iyi değillerdir.
- Optimal bir karar ağacı öğrenme probleminin, optimalliğin çeşitli yönleri altında ve hatta basit kavramlar için bile NP-zor (Deterministik olmayan algoritmalar yardımıyla polinom zamanda çözülebilen karar problemlerin sınıfı) olduğu bilinmektedir. Sonuç olarak, pratik karar ağacı öğrenme algoritmaları, her düğümde yerel olarak en uygun kararların verildiği açgözlü algoritma gibi buluşsal algoritmalara dayanır. Bu tür algoritmalar, küresel olarak en uygun karar ağacını döndürmeyi garanti edemez. Bu, özelliklerin ve örneklerin değiştirilerek rasgele örneklendiği bir topluluk öğrenicisinde birden çok ağacın eğitilmesiyle azaltılabilir [38], [39].

Karar ağaçları, her testte sayısal bir özelliğin bir eşik değerle karşılaştırıldığı, bir dizi temel testi verimli ve tutarlı bir şekilde birleştiren ardışık bir modeldir [40]. Kavramsal kuralların oluşturulması, sinir ağındaki düğümler arasında yer alan bağlantılara ait ağırlıkların oluşturulmasından çok daha kolaydır [41]. Ağırlıklı olarak gruplama amaçları için karar ağaçları kullanılır. Ayrıca karar ağaçları, veri madenciliğinde sıklıkla kullanılan bir sınıflandırma modelidir [42]. Sürecin en kritik hususu, karar ağacı yapısını oluştururken hangi algoritmadan yararlanılacağının belirlenmesidir. Literatürde çeşitli karar ağacı algoritması bulunmaktadır.

4.1. Bölme Türleri

Karar ağaçları için kullanılan çeşitli bölme türleri vardır. Birkaç özniteliğin test edilmesine dayalı çok değişkenli bölmeler bazen daha iyi ağaçlara yol açabilir, ancak bunlar, bölünmüş seçimin hesaplama masrafını daha büyük veri kümeleri için kabul edilebilir sınırların ötesine çıkarır ve yaygın pratik kullanımda değildir. Farklı bölme türleri, kullanılan test fonksiyonunun biçimi ile karakterize edilir. t test fonksiyonu ile a özniteliği üzerinde tek değişkenli bir bölünmeyi tanımlamak, tüm $x \in X$ için $t(x)$'in $a(x)$'e dayalı olarak nasıl belirlendiğini belirtmeyi gerektirir.

Nominal nitelikler için, aşağıdaki iki ayırma türünden birini kullanmak yaygındır:

- ✓ Değer tabanlı: $t(x) = a(x)$ olarak tanımlanan bir test fonksiyonu ile.
- ✓ Eşitlik tabanlı: $t(x) = \begin{cases} 1, & a(x) = v \\ 0, & d.durumlarda \end{cases}$ olarak tanımlanan bir test fonksiyonu ile.

Değere dayalı bir bölme yalnızca bir özelliktir ve her sonuç bir özellik değerine karşılık gelmektedir. Eşitliğe dayalı bir ayırma ikilidir ve bir sonucu belirli bir öznitelik değerine sahip örneklere, diğerini de kalan örneklere atamaktadır. Aynı ağaçta hem değere dayalı hem de eşitliğe dayalı bölmeleri kullanmanın pek bir anlamı yoktur, bu nedenle karar ağacı algoritması uygulamalarında birini veya diğerini dikkate almak standart bir uygulamadır.

Sürekli nitelikler için en yaygın tek bölme türü, eşitsizlik ilişkisini kullanır:

- ✓ Eşitsizlik tabanlı: $t(x) = \begin{cases} 1, & a(x) \leq v \\ 0, & d.durumlarda \end{cases}$.
- ✓ Aralık tabanlı: $t(x) = \begin{cases} 1 & eğer\ a(x) \in I_1 \\ 2 & eğer\ a(x) \in I_2 \\ \dots & \dots \end{cases}$

Eşitsizlik tabanı daha esnek yapıdadır, ancak seçimi daha maliyetli olan bir ayırma türüdür. Bir özelliğin ortak etki alanının birkaç aralığına farklı sonuçlar atamaktadır. Aralık tabanlı bölme türünde $I_1, I_2, \dots, I_k \subset A$, a'nın ortak tanım kümesinin ayrık bir bölümünü oluşturan aralıklardır. Bu, sürekli bir özniteliğe uygulanan altküme tabanlı bölme ile açıkça aynıdır.

Bazı özellikleri hem nominal hem de sürekli olanlarla paylaşan sıralı öznitelikler, kişinin mevcut düzen ilişkisinden yararlanmak isteyip istemediğine bağlı olarak bu türlerden herhangi biri olarak ele alınabilmektedir.

4.1.1. Bilgi kazancı

Bir dizi örnekte sınıf dağılımının safsızlığını karakterize etmek için birkaç farklı ölçü kullanılmaktadır. Yalnızca bölmeden sonra elde edilen sınıf dağılım safsızlığını değil, daha çok safsızlığın azalmasını ölçen, biraz değiştirilmiş bölünmüş değerlendirme fonksiyonlarının kullanılması alışılmadık bir durum değildir [43].

Bilgi kazancı, bölmenin uygulanacağı düğüme karşılık gelen örneklerin alt kümesi için safsızlık ile bölme sonuçlarına karşılık gelen alt kümelerin ağırlıklı ortalama safsızlığı arasındaki fark olarak tanımlanır. Entropi için fark denklem 4.1 ile yazılmaktadır.

$$\Delta E_{T_n}(c|t) = E_{T_n}(c) - E_{T_n}(c|t) \tag{4.1}$$

Denklem 4.1'de entropi artık bilgi kazancı olarak adlandırılmaktadır. İlk terim değerlendirilen bölünmeye bağlı olmadığından, bilgi kazancı minimize edilmek yerine maksimize edildiği sürece entropi ile tam olarak aynı bölünmelerin seçilmesine yol açmaktadır. Bilgi kazancının avantajı, yalnızca hangi ayrımın en iyi olduğunu göstermesi değil, aynı zamanda ne kadar iyileştirme sağladığını da göstermesidir. Bu bazen, mevcut en iyi bölünmeden kaynaklanan iyileştirme çok küçük olduğunda bir düğümü yaprağa dönüştüren ek bir durdurma kriteri tanımlamak için kullanılmaktadır.

4.1.2. Gini indeksi

Gini indeksi, hedef niteliklerin değerlerinin olasılık dağılımları arasındaki ıraksamayı ölçen katışkı tabanlı (impurity - based) bir ölçüttür. S veri kümesindeki $a: X \to V$ özniteliğinin Gini indeksi denklem 4.2 ile hesaplanmaktadır [44]:

$$GI_s(\alpha) = \sum_{v \in A} P_s(a = v)\big(1 - P_s(a = v)\big) = 1 - \sum_{v \in A} P^2{}_s(a = v) \tag{4.2}$$

Entropi için benzer şekilde, maksimum ve minimum değerleri, sırasıyla maksimum ve minimum safsızlığa karşılık gelmektedir.

4.2. Büyüme ve Budama

Belirli bir eğitim setinden bir karar ağacı modeli oluşturmak için gereken en önemli sürece "büyüme" adı verilir. "Gerçek" (biyolojik) ağaçlardan ödünç alınan bu terimden de anlaşılacağı gibi, genellikle yeni düğümlerin veya yaprakların adım adım eklendiği sıralı bir süreçtir. Yeni düğümlerin veya yaprakların eklenmesi, tek bir kök düğümden başlayarak yukarıdan aşağıya doğru gerçekleştirilir. Geniş pratik kullanımda tüm karar ağacı algoritmaları tarafından takip edilen bu paradigma, yukarıdan aşağıya karar ağacı indüksiyonu (TDIDT) olarak adlandırılır.

Karar ağacı budama işlemi ağacın genelleme yeteneğini geliştirmek amacıyla aşırı büyümüş bazı alt ağaçların kesilmesi ve bunların yapraklarla değiştirilmesiyle sonuçlanan büyümenin tersi olarak düşünülebilir. Çoğu durumda, zayıf düğümlerin oluşmasını ve işe yaramaz alt ağaçların büyümesini önleyecek daha rafine durdurma kriterleri kullanılarak israftan kaçınmak istenebilir. Bu yaklaşım ön budama olarak bilinmektedir. Bir budama algoritmasını tam olarak tanımlamak için, aşağıdaki ana bileşenlerin belirtilmesi gerekir:

Budama operatörleri: Ağaçtan düğümleri kesme işleminin tam olarak nasıl gerçekleştirildiğini belirlemektedir. Yaprağa, karşılık gelen eğitim örnekleri alt kümesinin çoğunluk sınıf etiketi atanmalıdır. Ağaç büyümesi sırasında düğümlere sınıf etiketleri atamanın uygun olmasının bir nedeni de budur- budama söz konusu olduğunda, etiketler zaten oradadır. Bu işleç, aşırı büyümüş bir alt ağacın "büyümemesi" olarak düşünülebilir [44].

Budama kriteri: Belirli bir düğüme bir budama operatörünün uygulanıp uygulanmayacağının nasıl yargılanacağını belirlemektedir. Bu, bazı kalite ölçütlerine göre, düğümde köklenen orijinal alt ağaç ile yeni bir alt ağaç (özellikle en yaygın alt ağaç kesme operatörü için tek bir yaprak) karşılaştırılarak yapılmaktadır [45].

Farklı budama teknikleri mevcuttur. ***İndirgenmiş hata budama*** fikri, doğrudan bir karar ağacının genelleme kabiliyetini geliştirmek olan budama amacından gelmektedir. Performans ölçüsü olarak yanlış sınıflandırma hatasının kullanıldığı varsayılırsa, kriter, orijinal alt ağacın ve onun yerini alacak olan yaprağın hatasını karşılaştırmaya dayanmaktadır. Aslında, hata eğitim setinde hesaplanırsa orijinal alt ağaç (neredeyse) her zaman kazanır, ancak bu açıkça genelleme hakkında hiçbir şey söylemez [46].

Kötümser budama fikri, alt ağacın üzerinde büyüdüğü aynı örnekler kümesi üzerinde tahmin edilen bir alt ağacın hatasının iyimser bir şekilde yanlı olması gerektiği ve bu nedenle kötümser bir düzeltmeye ihtiyaç duyduğu şeklindeki bariz gözleme dayanmaktadır. Düzeltme, denklem 4.3 ile ifade edilen düzeltme terimi eklenerek elde edilir:

$$\tilde{e}_T(n) = e_T(n) + \sqrt{\frac{\hat{e}_T(n) - (1 - \hat{e}_T(n))}{|T_n|}} \tag{4.3}$$

Düzeltme terimi, hatanın standart sapmasının bir tahmini olarak düşünülebilir ve bunu orijinal hataya eklemek, karşılık gelen güven aralığının bir üst sınırını almakla eşdeğerdir ancak bu, kesin ve resmi bir gerekçelendirmeden çok, bu yaklaşımın sezgisel bir açıklaması olarak düşünülmelidir [47].

Minimum hata budama kriteri, kötümser hata budama ile aynı hedeflere ulaşmanın daha zarif ve daha iyi kontrol edilebilir bir yoludur, yani, incelenmekte olan düğümün ve değiştirilen yaprağın eğitim seti hatalarını doğal iyimser önyargı için bir miktar telafi ile karşılaştırmaktadır. Herhangi bir yaprak I'nin hatasını, m-tahmini doğruluğunun I'in tümleyeni olarak tahmin etmektedir. Kriter m-tahmin tekniği kullanılarak denklem 4.4'deki gibi hesaplanır;

$$\hat{e}_T(I) = 1 - \frac{|\{x \in T_I \mid c(x) = d_I\}| + mp}{|T_I| + m} \tag{4.4}$$

Denklem 4.4'te iyi bilgi yokluğunda, d_I sınıfının apriori olasılığı $p = {}^{1}/_{|C|}$olarak alınmaktadır [48]. Yaprakların hataları, m-tahmin edilen doğruluklarına göre hesaplandığından, "gerçek" örneklerinin daha küçük alt kümesi olan m "hayali" örneklerden daha fazla etkilenmektedir. Bu nedenle, orijinal alt ağacın tamamen doğru olan yaprakları bile, yeterince büyük m için oldukça zayıf hata tahminleri elde edebilir ve bu tahminler, tüm orijinal alt ağacın hata tahminini elde etmek için yukarı doğru yayılabilmektedir. Bu yorumun açıkça gösterdiği gibi, m parametresi budamanın agresifliğini ayarlamak için kullanılabilir (ne kadar büyükse, o kadar fazla düğüm budanır) ve belirli bir veri kümesi için ayarlanmalıdır. Aşırı uydurma riskinin daha yüksek olduğu gürültülü veri kümeleri genellikle daha büyük m değerleri gerektirmektedir [48].

Budama kontrol stratejisi: Hangi aday düğümlerin budama için dikkate alınacağı sırayı belirlemektedir [49]. Sıranın, budamanın genel etkisi üzerinde önemli bir etkisi olabilir ve farklı kontrol stratejilerinin farklı kalitede farklı nihai ağaçlar üretmesi muhtemeldir. Kontrol stratejisi genellikle aşağıdakilerden biridir:

Alt üst: Bu, son seviyeden başlayıp yukarı doğru giden budama için düğümleri dikkate almaktadır [49].

Yukarıdan aşağıya: Kök düğümden başlayıp aşağı doğru giden budama için düğümleri dikkate almaktadır [49].

En iyi-ilk: Bu, budama kriteri tarafından belirtilen olası sonuç iyileştirmesinin ima ettiği sırayla budama için düğümleri ve operatörleri dikkate almaktadır [49].

Başarılı bir budama algoritması, budama kriteri ile budama kontrol stratejisinin iyi bir kombinasyonuna ihtiyaç duymaktadır. Bazı budama kriterleri, bir kontrol stratejisiyle iyi, diğeriyle kötü çalışma eğilimindedir. Bir budama algoritması tasarlanırken, önce

budama kriteri gelmektedir ve bu kritere en iyi uyan kontrol stratejisi seçilmektedir. Aşağıdan yukarıya ve en iyiye öncelik veren stratejiler (saf veya birleşik formda), genellikle çok agresif olan yukarıdan aşağıya stratejiden çok daha sık görünmektedir. Özellikle, yukarıda sunulan en yaygın budama kriterlerinin tümü tipik olarak aşağıdan yukarıya stratejisiyle birleştirilmektedir.

4.3. Karar Ağaçları İndükleyicileri

4.3.1. ID3 Algoritması

ID3 karar ağacı algoritması klasik bir algoritmadır; J. Ross Quinlan [46] tarafından bulunmuştur. Algoritma kök düğümden başlamaktadır. Kök düğüm en iyi özelliklerden biridir. ID3 algoritması, Hunt'ın algoritmasına dayalıdır ve seri olarak uygulanmaktadır. ID3 ağacı iki aşamada inşa edilmektedir: ağaç oluşturma ve ağaç budama [50], [51], [52]. Veriler, en iyi bölme tek özniteliğini seçmek için ağaç oluşturma aşamasında her düğümde sıralanmaktadır. ID3 algoritmasının arkasındaki ana fikirler şunlardır:

1. Bir karar ağacının yaprak olmayan her düğümü, bir girdi niteliğine karşılık gelmekte ve her dal, bu özelliğin olası bir değerine karşılık gelmektedir. Bir yaprak düğüm, girdi öznitelikleri kökten o yaprak düğüme giden yolla açıklandığında, çıktı özniteliğinin beklenen değerine karşılık gelmektedir.
2. "İyi" bir karar ağacında, yaprak olmayan her düğüm, kökten o düğüme giden yolda henüz dikkate alınmayan tüm girdi nitelikleri arasında nitelik hakkında en bilgilendirici olan girdi özelliğine karşılık gelmelidir. Bunun nedeni, ortalama olarak mümkün olan en az sayıda soruyu kullanarak çıktı niteliğini tahmin etmek istemektir ve
3. Entropi, belirli bir girdi niteliğinin, bir altkümesi için çıktı niteliği hakkında ne kadar bilgi verici olduğunu belirlemek için kullanılmaktadır.

ID3, entropi işlevini ve bilgi kazancını ölçüt olarak kullanılmaktadır [53], [54].

ID3 algoritması matematiksel olarak incelendiğinde; S'nin bir veri örneği kümesi olduğunu varsayalım. Sınıf etiketi özniteliğinin m farklı değeri olduğunu varsayalım, m farklı sınıfın Ci $(i = 1 \dots m)$ tanımıdır. Si, Ci sınıfındaki örnek sayısıdır. Denklem 4.5, bilgilerin beklentilerine göre belirli bir örnek sınıflandırması üzerindedir.

$$I(s_1, s_2, \dots, s_m) = -\sum_{i=1}^{m} p_i log_2(p_i) \qquad (4.5)$$

Denklem 4.5'te p_i Ci' ye ait herhangi bir numunenin olasılığıdır [53].

A kümesi özniteliği v farklı $\{a_1,a_1,\dots,a_1\}$ değerlere sahiptir. Bir özellik v altkümeye bölünebilir $S\{S_1,S_2,\dots,S_v\}$. Burada, S_j, bu örnekte bir dizi S içermektedir, A'da a_j değerine sahiptirler. Test özniteliği A olarak seçilirse, bu alt kümeler, S kümesine karşılık gelen dallanma büyümesinden düğümler içermektedir. S_j varsayımı S_{ij}, Ci sınıfı örneklerinin bir alt kümesidir. A'ya göre entropinin alt kümelere bölünmesi veya beklenen bilgiler denklem 4.6 ile verilmektedir:

$$E(A) = \sum_{i=1}^{v} \frac{s_{1j} + \cdots + s_{mj}}{s} I(s_{1j}, \dots, s_{mj}) \tag{4.6}$$

$(s_{1j} + \cdots + s_{mj})/s$ madde altkümesi birinci j'nin sağındadır ve örneklemin altkümesi sayısının örneklemdeki toplam S sayısına bölünmesine eşittir. Denklem 4.7, S_j'nin verilen bir alt kümesidir.

$$I(s_{1j}, \dots, s_{mj}) = -\sum_{i=1}^{m} p_{ij} log_2(p_{ij}) \tag{4.7}$$

Denklem 4.7'de $p_{ij} = s_{ij}/|s_j|$ Ci sınıfına ait olma olasılığındaki bir S_j örneğidir. Denklem 4.8 bir dal olup kodlama bilgisinde kullanılmaktadır. ID3 algoritması, doğru dallandırma yapmak için bilgi kazancını kullanmaktadır.

$$Kazanç(A) = I(s_{1j}, \dots, s_{mj}) - E(A) \tag{4.8}$$

Başka bir deyişle, Kazanç (A), sıkıştırma entropisinin beklentilerinin bir sonucu olan bu özelliğin değerinden kaynaklanmaktadır. Bundan, entropi değeri ne kadar küçükse, korelasyon o kadar düşük, daha yüksek saflığın bölünmesinin bir alt kümesi, karşılık gelen bilgi kazancı o kadar yüksek olmaktadır. Bu nedenle, test niteliği karar ağacı en yüksek bilgi kazancına sahip özellikler için seçilmiştir.

4.3.2. C4.5 Algoritması (J48)

C4.5, en iyi bilinen ve en yaygın kullanılan karar ağacı algoritmalarından biridir [55]. Doğruluk seviyesi, işlenecek veri hacminden bağımsız olarak yeterince yüksektir. Karar ağaçlarını ve diğer öğrenme algoritmalarını karşılaştıran en son çalışmalardan biri, C4.5'in çok iyi bir hata oranı ve hız kombinasyonuna sahip olduğunu göstermektedir [56], [57]. Algoritma, tamamlanmamış bir eğitim veri setini işleme ve boyutunu küçültmek ve karar yolunu optimize etmek için sonuçtaki karar ağacını budama yeteneğine sahiptir [58]. C4.5 ayrıca sürekli niteliklerle başa çıkma yeteneğine de sahiptir. Binarizasyon sürecini kullanarak sürekli öznitelikleri yönetmektedir. Bu nitelikler, verileri iki aralığa ayıran eşik değerleri kullanan ayrıklarla değiştirilir [59].

C4.5 algoritması ID3 algoritmasının varisidir ancak bu algoritma çalışmasında kayıp verileri hesaba dahil etmeyerek ID3 algoritmasından farklılaşmaktadır. Algoritma kazanım oranı hesabında sadece eksik verisi olmayan değerleri kullanmakta ve böylece daha hassas ve anlamlı kurallar çıkarabilmektedir. İlgili algoritmada ağacın budanması için iki yöntem kullanılmaktadır; ağaç içindeki alt ağaçların yapraklara dönüştürülmesi ve bir alt ağacın bu ağacı en çok kullanan ağacın yerini almasıdır.

4.3.3. CART (C&RT) Algoritması

Sınıflandırma ve Regresyon Ağacı (kısaca CART), yaygın kullanımda olan çok popüler bir ağaç tabanlı yöntemdir. CART her zaman bir ikili ağaç oluşturur, yani terminal olmayan her düğümün iki alt düğümü vardır. Bu, birden çok alt düğüme izin verebilen genel ağaç tabanlı yöntemlerin tersidir. Ağaç tabanlı yöntemin ve özellikle CART'ın büyük bir çekiciliği, karar sürecinin biz insanların nasıl karar verdiğimize çok benzemesidir. Bu nedenle, ağaç tarzı karar sürecinden çıkan sonuçları anlamak ve kabul etmek kolaydır. Bu sezgisel açıklayıcı güç, ağaç yönteminin asla ortadan kalkmayacağının en büyük nedenlerinden biridir. CART algoritmasının bir başka çekici yönü de lojistik regresyon veya destek vektör makinesi gibi çeşitli girdi verisi türlerine izin vermesidir. Böylece girdi verileri, fiyat veya alan gibi sayısal değişkenleri ve ev tipi veya konumu gibi kategorik değişkenleri karıştırabilir. Bu esneklik, CART algoritmasını çeşitli uygulamalarda tercih edilen bir araç haline getirmektedir [44].

CART mekanizması opsiyonel otomatik sınıf dengelemesi ve otomatik eksik değer (missing value) işlemlerini içermektedir. Farklı çapraz-sağlama klasörlerindeki ağaçların terminal düğümlerin sayısı ile hizaya gelemeyebileceğini farz edersek, CART yazarları budama dizisinde her bir ağaç için performans ölçümünde çapraz-sağlamanın nasıl yapıldığının göstererek çığır açmışlardır [60].

CART'ın belirli bir ağaç oluşturma yöntemi vardır. İlk olarak, her zaman girdinin tek bir özelliği (değişkeni) boyunca bölünmektedir. Giriş vektörü x'in $x = (x_1, \cdots, x_d)$ biçimindedir ve burada her x_i, sayısal, kategorik (nominal) veya ayrık sıralı değişken olabilmektedir. Her düğümde, CART her zaman bu değişkenlerden birini seçmekte ve düğümü yalnızca o değişkenin değerlerine göre bölmektedir. Başka bir deyişle, CART, lojistik regresyonun yaptığı gibi değişkenleri birleştirmez. CART'ın bölme yaptığı başka bir özel yol da her zaman iki alt düğüm oluşturmasıdır- ikili bölme- ki bu da bir ikili ağaçla sonuçlanmaktadır. Tabii ki, genel ağaç tabanlı yöntemler, düğümleri birden çok (ikiden fazla) alt düğüme bölebilir. Ancak CART için bölme her zaman ikilidir. Bunun

nedeni yine basitliktir, çünkü tek bir çoklu bölme birkaç ardışık ikili bölme ile eşdeğer olarak yapılabilmektedir [60].

İkili bölmenin her aşamasında düğümlerin kendinden daha homojen alt dallara ayrılması sağlanmaktadır. Ayrılma kriteri için entropy, Gini, twoing, simetrik Gini, en küçük kareler sapması gibi çeşitli yöntemler bulunmaktadır. En sık kullanılan ayırma kriteri Entropy ve Twoing kuralıdır, bu kurallar detaylı açıklanmıştır [61].

4.3.3.1. Twoing kriteri

Twoing kriteri, CART algoritmasında [44] nesnelerin her düğümde alt bölümlere en iyi şekilde bölünmesini seçmek için kullanılan bir ölçüdür. $a \in A$ özniteliğine göre tanımlanan $X_1, X_2 \subseteq U$ bölümünün Twoing kriteri denklem 4.9'daki gibi hesaplanır;

$$Twoing_a(X|X_1, X_2) = \frac{|X_1|.|X_2|}{4}\left(\sum_i |p_1^i - p_2^i|\right)^2 \tag{4.9}$$

Denklem 4.9'da p_j^i, X_j, $j \in \{1, 2\}$ kümesindeki i. karar sınıfındaki nesnelerin bir olasılığıdır. CART algoritması, yukarıdaki ikili bölme kuralını en üst düzeye çıkaran öznitelikleri seçmektedir. Twoing kriteri, her düğümde bir ikili alt bölünme verirken, entropi kriteri, koşullu öznitelikteki değişkenlere bağlı olarak birden fazla alt bölünme verebilmektedir. Bu nedenle, karar ağacı oluşturma sürecinde ikileme algoritması kullanıldığında üretilen ağacın yüksekliği açısından karmaşıklık, entropi algoritmasına göre daha yüksektir. Twoing kriteri hem kategorik hem de sayısal değerleri ele alırken, entropi yalnızca kategorik değeri ele almaktadır [62].

4.3.3.2. Entropi kriteri

Karar ağaçlarının oluşturulmasında, öncelikle tüm koşullu nitelikler arasından nesnelerin bölünmesi için temel olarak kullanılacak en iyi öznitelik seçilerek kök düğüm adı verilen bir başlangıç düğümü oluşturulur. Entropi, nesneler koşullu özniteliğe göre bölündüğünde, bir alt düğümdeki düzensizlik seviyesinin ölçüsüdür. $X \subseteq U$ nesneleri kümesi için entropi değeri denklem 4.10 kullanılarak tanımlanır;

$$E(X) = \sum_{i=1}^{c} -p_i log_2 p_i \tag{4.10}$$

Denklem 4.10'da p_i, X kümesindeki i karar sınıfına ait nesnelerin bir olasılığıdır ve c, karar sınıflarının sayısıdır. Karar sınıfı sayısı ikiye eşit olduğunda entropi değeri 0 ile 1 arasındadır. Entropinin 0'a eşit olması X kümesinde düzensizlik olmadığını, 1'e eşit olması ise X kümesinin oldukça düzensiz olduğunu göstermektedir. Ağaç yapımının her

seviyesinde, değerlerine göre bölmeler oluşturan çeşitli nitelikler dikkate alınmaktadır. Bir düğümü tanımlamak için minimum entropiye sahip bir öznitelik seçilmektedir. Avantaj olarak, entropi ölçüsünün kullanılması, çok sınıflı eğitim veri kümeleri için etkilidir. Karar ağacı modellerinde nesneleri çeşitli konularda sınıflandırmak için entropi kriteri kullanılmış ve bu da çok iyi doğruluk oranları sağlamıştır [63].

4.3.4. CHAID (Ki-Kare Otomatik Algılama Dedektörü) Algoritması

Adından da anlaşılacağı gibi, CHAID bir Ki-kare bölme kriterine dayanmaktadır. Ki-kare otomatik etkileşim detektörü (CHAID) algoritması, eldeki her türlü problemi çözmeye uyarlanabildiği için en çok kullanılan denetimli öğrenme yöntemlerinden biri olarak kabul edilmektedir. Daha spesifik olarak, Ki-karenin p-değerini kullanmaktadır. Kass tarafından tanıtılan CHAID algoritması, AID ve THAID'in bir evrimi olarak, günümüzde bu daha önceki istatistiksel denetimli ağaç yetiştirme teknikleri arasında en popüler olanıdır [64].

CHAID algoritmasının temel fikri, örnekleri verilen hedef değişkene ve seçilen özellik indeksine (tahmin değişkeni gibi) göre en uygun şekilde bölmek ve ki-kare testinin önemine göre otomatik olarak yargılamak için olasılık tablosunu gruplandırmaktır. Kriter, bölme tarafından oluşturulan g düğümleri arasındaki ortalama değerlerdeki fark için F istatistiğinin p değeridir ve denklem 4.11 ile hesaplanmaktadır:

$$F = \frac{BSS/(g-1)}{WSS/(n-g)} \sim F_{(g-1),(n-g)} \tag{4.11}$$

Denklem 4.11'de WSS notasyonu, varyans analizinde genellikle kalıntı veya kareler toplamı olarak bilinmektedir ve denklem 4.12 ile hesaplanmaktadır:

$$WSS = \sum_{j=1}^{g} \sum_{i=1}^{n_j} \left(y_{ij} - \bar{y}_j\right)^2 \tag{4.12}$$

Denklem 4.12'de $\bar{y}_j$, j düğümündeki y_{ij}'lerin ortalama değeridir. BSS notasyonu ise gruplar arası kareler toplamı olarak adlandırılır ve $BSS = TSS - WSS$ formülü ile hesaplanmaktadır.

CHAID'in popülaritesine katkıda bulunan temel özellikleri şunlardır:

1. Her düğümde, CHAID her bir potansiyel öngörücü için üreteceği optimal $n-ary$ bölünmesini belirler ve öngörücüyü bu optimum bölmelere dayanarak seçer.

2. CHAID, bölme kriteri olarak Bonferroni düzeltmeli p-değerlerini kullanır.

Büyüyen kriterler olarak $p-$değerlerine başvurmak, istatistiksel anlamlılığı otomatik olarak açıklayan durdurma kuralları sağlamaktadır. Eşikler doğal olarak istatistiksel anlamlılık için dikkate alınan olağan kritik değerlere, yani $\%1, \%5$ veya $\%10$ olarak ayarlanır. Bu tür p -değeri kriterleri, bölünmeye dahil olan durumların sayısına duyarlıdır ve çok küçük gruplara bölünmeyi önleme eğilimindedir.

CHAID algoritması aşağıdaki süreçlerin uygulanmasını sağlar:

- Girdi değişkenleri kümesinden ilgili bağımsız değişkenlerin, ortaya çıkan hiyerarşik olarak düzenlenmiş yapıda, girdi verilerinin bölümlenmesi için ilk bağımsız değişken olarak en düşük p-değerine sahip değişkenin seçileceği şekilde seçilmesi ve bu nedenle, bağımlı değişkenle en güçlü şekilde ilişkilidir. Hipotez testi prosedüründe, p değeri önceden tanımlanmış önem düzeyine (α) eşit veya bundan düşükse, o zaman değişkenler arasında bir bağımlılık öneren alternatif hipotez kabul edilmektedir; bu, ağaç geliştirme bağlamında, düğümü belirtmektedir. Aksi takdirde, düğüm uç düğüm olarak kabul edilmektedir. Ağaç oluşturma, gözlemlenen tüm bağımsız değişkenlerin p değerleri belirli bir bölünme eşiğinden yüksek olduğunda sona ermektedir.
- Her bir bağımsız değişkenin kategorilerini, aralarında istatistiksel olarak anlamlı fark bulunan belirli sayıda düğüm ağaçta görünecek şekilde birleştirilmesidir. Aslında, algoritma bağımlı değişkenden en az farklı olan bağımsız değişkenlerin değer çiftlerini tanımlanmaktadır, böylece açıklayıcı değişken kategorilerinin sayısı Ki-kare testi sonuçlarına ve p değerine bağlıdır. Elde edilen p değeri belirli bir birleştirme eşiğinden yüksekse, algoritma, belirli kategorileri istatistiksel olarak anlamlı farklar olmadan birleştirmektedir. Bundan sonra, yeni bir birleşen çift arayışı, p-değerinin tanımlanan anlamlılık seviyesinden (α) daha küçük olan çiftler belirlenemeyene kadar devam etmektedir.

Akışa göre, CHAID analizinde istatistiksel testlerin bireysel değerlerin kombinasyonu ve açıklayıcı değişken kategorilerinin belirlenmesi ve açıklayıcı değişkenlerin bağımlı değişkenle olan ilişkilerinin istatistiksel önemine göre seçilmesi şeklinde iki temel işlevini belirlemek mümkündür [65].

CHAID algoritması, verilerin fazla takılmasını önlemek ve daha fazla karar ağacı bölünmesini engellemek için kullanılmaktadır. CHAID hesaplaması, yeterli bir p değeri elde edildiğinde sona ermektedir. Bu bölme yöntemi, veri analizi için yaygın bir

alternatiftir ve ayrıntılı yöntem olarak da bilinmektedir. Birleştirme, ikili bir bölünme elde edilene kadar devam eder. Ardından en uygun p değerine sahip olan split seçilmektedir [66].

4.3.5. NbTREE

NbTree algoritması, karar ağacı sınıflandırıcıları ile Naive Bayes sınıflandırıcıları arasında bir melezdir. Öğrenilen bilgiyi, yinelemeli olarak oluşturulmuş bir ağaç biçiminde temsil etmektedir. Bununla birlikte, yaprak düğümler, tek bir sınıfı tahmin eden düğümlerden ziyade Naïve Bayes kategorileştiricileridir [67]. Sürekli nitelikler için, entropi ölçüsünü sınırlamak için bir eşik seçilmektedir. Bir düğümün faydası, verilerin ayrıklaştırılması ve düğümde Naïve Bayes kullanılarak beş katlı çapraz doğrulama doğruluğu tahmininin hesaplanmasıyla değerlendirilmektedir. Bölünmenin faydası, düğümlerin faydalarının ağırlıklı toplamıdır ve bu, o düğümden geçen örneklerin sayısına bağlıdır. NBTree algoritması, her yapraktaki Naïve Bayes'in genelleştirme doğruluğunun, geçerli düğümdeki tek bir Naïve Bayes sınıflandırıcısından daha yüksek olup olmadığını tahmin etmeye çalışmaktadır. Hatadaki göreli azalma %5'ten büyükse ve düğümde en az 30 örnek varsa, bir bölünmenin önemli olduğu söylenmektedir [67].

NbTree'nin temel fikri, yüksek boyutlu noktalar için indeks anahtarı olarak Öklid norm değerini kullanmaktır. Saf Bayes'in öznitelik bağımsızlığı varsayımı tüm eğitim verilerinde her zaman ihlal edilse de yerel eğitim verileri içindeki bağımlılıkların tüm eğitim verilerindekinden daha zayıf olması beklenebilir. Böylece, NbTree [68], karar ağacı sınıflandırıcılarının ve Naïve Bayes sınıflandırıcılarının avantajlarını bütünleştiren, oluşturulmuş karar ağacının her bir yaprak düğümü üzerinde saf bir Bayes sınıflandırıcısı oluşturmaktadır. Basitçe söylemek gerekirse, öncelikle, eğitim verilerinin her bir bölümünün bir ağaç yaprak düğümü tarafından temsil edildiği eğitim verilerini bölümlere ayırmak için karar ağacını kullanır ve ardından her bölüm üzerinde saf bir Bayes sınıflandırıcısı oluşturmaktadır. Karar ağaçlarının oluşturulmasında temel bir konu, ağacın terminal olmayan her bir düğümündeki öznitelik seçim ölçüsüdür. Yani, karar ağaçlarının oluşturulmasında uç olmayan her bir düğümün ve bir bölünmenin faydasının ölçülmesi gerekmektedir.

NbTree, gerçekten de sınıflandırma performansı açısından Naïve Bayes'ten önemli ölçüde daha iyi performans göstermektedir. Bununla birlikte, yüksek zaman karmaşıklığına maruz kalmaktadır; çünkü bir bölme oluştururken Naïve Bayes sınıflandırıcılarını tekrar tekrar oluşturması ve değerlendirmesi gerekmektedir.

4.3.6. RepTREE

Azaltılmış hata budama (REP) ile bir karar ağacının birleştirilmesiyle oluşturulan Azaltılmış Hata Budama Ağacı ("REPT") temelde hızlı karar ağacı öğrenimidir ve bilgi kazanımına veya varyansı azaltmaya dayalı bir karar ağacı oluşturmaktadır. Bu yaklaşımda DT, modellemek için eğitim veri setini kullanmaktadır; karar ağacının performansı yüksek olduğunda, REP ağaç yapısı karmaşıklığını en aza indirmektedir. Budama yöntemi, geriye dönük aşırı uyum problemlerini açıklamaktadır [68].

REP ağacı, bilgi kazancını bölme kriteri olarak kullanarak bir karar/gerileme ağacı oluşturan ve azaltılmış hata budama kullanarak budayan hızlı bir karar ağacı öğrenicisidir. Sayısal nitelikler için değerleri yalnızca bir kez sıralamaktadır. RepTree, regresyon ağacı mantığını kullanmaktadır ve farklı yinelemelerde birden çok ağaç oluşturmaktadır. Daha sonra oluşturulan tüm ağaçlardan en iyisini seçmektedir. Bu temsilci olarak kabul edilecektir. Ağacı budamada kullanılan ölçü, ağaç tarafından yapılan tahminlerdeki ortalama karesel hatadır.

4.3.7. RandomTREE

Rastgele Ağaçlar, esasen makine öğrenimindeki iki algoritmanın birleşimidir; tek model ağaçlar ve rastgele orman. Model ağaçları, her bir yaprağın, bu yaprak tarafından tanımlanan yerel altuzay için optimize edilmiş doğrusal bir modele sahip olduğu karar ağaçlarıdır. Rastgele ağaç denetimli bir sınıflandırıcıdır; birçok bireysel öğrenici üreten bir topluluk öğrenme algoritmasıdır. Bir karar ağacı oluşturmak için rastgele bir veri seti üretmek için bir torbalama fikri kullanmaktadır. Standart ağaçta her düğüm, tüm değişkenler arasında en iyi bölünme kullanılarak bölünmektedir. Rastgele bir ormanda, her düğüm, o düğümde rasgele seçilen tahmin edicilerin alt kümesinin en iyisi kullanılarak bölünmektedir. Rastgele ağaçlar Leo Breiman ve Adele Cutler tarafından üretilmiştir [69]. Algoritma hem sınıflandırma hem de regresyon problemleriyle başa çıkabilmektedir. Rastgele ağaçlar, orman adı verilen ağaç tahmincilerinin bir koleksiyonudur. Sınıflandırma şu şekilde çalışır: rastgele ağaçlar sınıflandırıcı, giriş özellik vektörünü almaktadır, ormandaki her ağaçla sınıflandırmaktadır ve "oyların" çoğunu alan sınıf etiketini çıkarmaktadır. Bir gerileme durumunda, sınıflandırıcı yanıtı, ormandaki tüm ağaçların yanıtlarının ortalamasıdır.

5. BULGULAR

Bu tez çalışmasının amacı veri madenciliği algoritmalarının;

- Üçlü ve Beşli Likert (Ordinal) ölçeğine sahip veri seti üzerindeki sınıflandırma performansını,
- Sınıflar arası dengesizliğin olduğu veri seti üzerinde sınıflandırma performansını,
- Dengesizliğin giderildiği durumda algoritmaların sınıflandırma performansını karşılaştırmaktır.

Bu kapsamda veri seti olarak tüm koşulları sağlayan TÜİK tarafından derlenen Türkiye Aile Yapısı Araştırması seçilmiştir. Türkiye Aile Yapısı Araştırması (TAYA) ilki 2006 yılında ve sonrasında beş yıllık periyotlarda gerçekleştirilen içerisinde bulunan farklı soru gruplarıyla ülkedeki ailelerin güncel durumlarıyla ilgili (evlilik, aile içi ilişkiler, akrabalık ilişkileri, çocuk, yaşlı ve diğer toplumsal konular) veri toplamayı amaçlayan bir araştırmadır.

Bu tez çalışmasında veri kaynağı olarak 2021 yılı TAYA verileri kullanılmıştır. 2021 yılında TAYA, ülke genelinde 19429 hane ve bu haneler de yaşayan 15 yaş ve üzeri 42044 fert ile gerçekleştirilmiştir. Çalışma kapsamında ise örneklem olarak evli bireyler seçildiğinden örneklem hacmi 27314 fert olarak belirlenmiştir. Çalışmada kullanılan değişkenlere ilişkin bilgiler Tablo 5.1 ile sunulmuştur.

Tablo 5.1. *Değişken grubu özet bilgisi*

Sıra No	Değişken	Açıklama	Değişken Tipi	Kategori
1	Y (Bağımlı değişken)	Mutluluk düzeyi	Nominal	1. Çok mutlu
				2. Mutlu
				3. Orta
				4. Mutsuz
				5. Çok mutsuz
2	X (Bağımsız değişkenler)	Eğitim durumu	Nominal	1.Bir okul bitirmedi
				2.İlkokul
				3.Genel ortaokul/Mesleki veya teknik ortaokul/İlköğretim
				4.Genel lise/Mesleki veya teknik lise
				511. 2 veya 3 yıllık yüksekokul
				512. 4 yıllık yüksekokul veya fakülte
				521. 5 veya 6 yıllık fakülte

Tablo 5.1.(Devam) *Değişken grubu özet bilgisi*

Sıra No	Değişken	Açıklama	Değişken Tipi	Kategori
2	X (Bağımsız değişkenler)	Eğitim durumu	Nominal	522. Yüksek lisans (5 veya 6 yıllık fakülteler hariç)
				53. Doktora
3		Yaş durumu	Nümerik	[16–94]
4		Aile mutluluk düzeyi	Nominal	1. Çok mutlu
				2. Mutlu
				3. Orta
				4. Mutsuz
				5. Çok mutsuz
5		Kişisel gelirden memnuniyet düzeyi	Nominal	1. Çok mutlu
				2. Mutlu
				3. Orta
				4. Mutsuz
				5. Çok mutsuz
6		Evlilik yıldönümünde hediye verme durumu	Nominal	1. Evet
				2. Hayır
7		Ev ile ilgili sorumluluklarda sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
8		Çocuklarla ilgili sorumluluklarda sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
9		Ailece birlikte vakit geçirememede sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
10		Harcamalarda sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
11		Giyim tarzında sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
12		Dini görüşlerimizin farklılığında sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla

Tablo 5.1.(Devam) *Değişken grubu özet bilgisi*

Sıra No	Değişken	Açıklama	Değişken Tipi	Kategori
12	X (Bağımsız değişkenler)	Dini görüşlerimizin farklılığında sorun yaşama	Nominal	5. Her zaman
13		Ailesi ile ilişkilerde sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
14		Alkol alışkanlığında sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
15		Sigara alışkanlığında sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
16		Kumar alışkanlığında sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
17		İşi ile ilgili sorunların eve taşınmasında sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
18		Bir işte çalışmaması durumunda sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
19		Gelirinin yeterli olmamasında sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
20		Arkadaşlar, görüşülen kişiler konusunda sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
21		Kendine yeterince özen göstermemesi durumunda sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman

Tablo 5.1.(Devam) *Değişken grubu özet bilgisi*

Sıra No	Değişken	Açıklama	Değişken Tipi	Kategori
22	X (Bağımsız değişkenler)	İnternet kullanımında sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
23		Kıskançlık konusunda sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
24		Kültürel farklılıklarda sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
25		Kişisel farklılıklarda sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
26		Cinsel uyumsuzluk konusunda sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman
27		Siyasi görüş farklılıklarında sorun yaşama	Nominal	1. Hiçbir zaman
				2. Nadiren
				3. Bazen
				4. Sıklıkla
				5. Her zaman

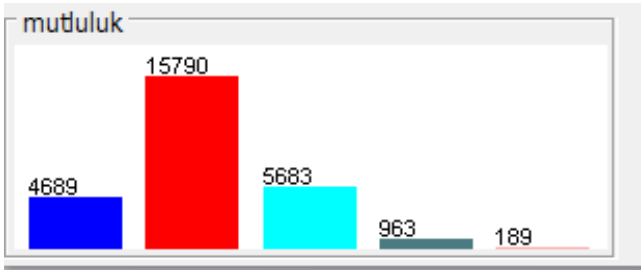
Tablo 5.1. incelendiğinde evli bireylerin mutluluk düzeylerini sınıflandırmak üzere 27 adet öznitelik uzman görüşü yardımıyla belirlenmiştir. Üzerinde çalışılan TAYA veri seti yapısı itibariyle dengeli bir veri seti değildir. Çalışma kapsamında ilk olarak sınıflandırma analizi dengeli olmayan veri seti üzerinde gerçekleştirilmiştir. Daha sonra veri setindeki dengesizlik giderilerek sınıflandırma analizi tekrarlanmıştır.

5.1. Dengeli Olmayan Veri Seti İçin Sınıflandırma Sonuçları

5.1.1. Beşli Likert ölçeğe sahip değişkenler için sınıflandırma sonuçları

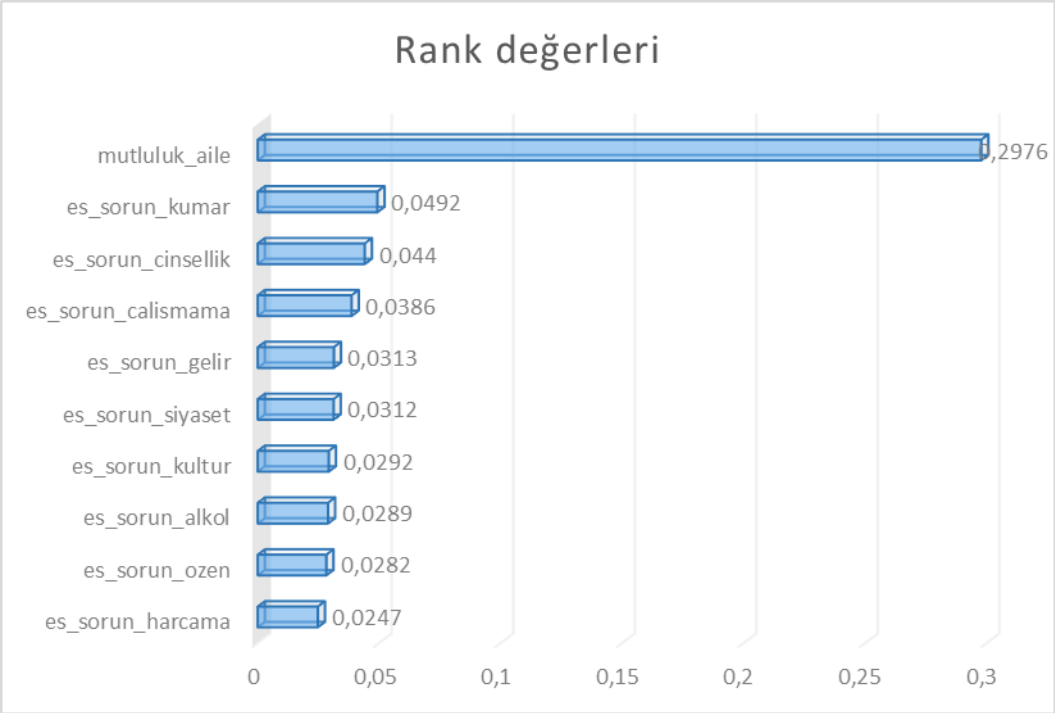
Sınıflandırma algoritmaları için “Mutluluk düzeyi” özniteliği hedef sütun olarak seçilmiştir. Mutluluk düzeyi özniteliğinin sınıf frekansları Şekil 5.1 ile gösterilmiştir.

Şekil 5.1 incelendiğinde sırasıyla “çok mutlu” sınıfında 4689 kişi, “mutlu” sınıfında 15794 kişi, “orta” sınıfında 5672 kişi, “mutsuz” sınıfında 970 kişi, “çok mutsuz” sınıfında 189 kişi bulunmaktadır.



Şekil 5.1. *Beşli Likert ölçeğine sahip veri setinde mutluluk düzeyine ait sınıf frekansları*

Beşli Likert ölçekli veri seti üzerinde gerçekleştirilen veri ön işleme adımı sonrasında veri seti WEKA 3.8.5 paket programında NumericToNominal filtresi kullanılarak sınıfı temsil eden özniteliğin kategorik olarak temsili sağlanmıştır. Ardından AttributeSelection filtresinde Information Gain (Bilgi Kazancı) algoritması ve Ranker (Sıralama) metodu kullanılarak her bir algoritma için en değerli 10 öznitelik seçilmiştir. Seçilen öznitelikler Şekil 5.2’ de sunulmuştur. Şekil 5.2 incelendiğinde “aile içinde mutluluk” değişkeninin sınıflandırma başarısında en etkili öznitelik olduğu görülmüştür.



Şekil 5.2. *Mutluluk sınıflandırması için bilgi kazancı metodu ile seçilen 10 adet öznitelik (beşli Likert ölçekli veri setinde)*

Çalışma kapsamında bahsi geçen J48 (C4.5), CHAID, CART, NbTREE, RepTREE ve RandomTREE algoritmalarının sınıflandırma başarıları değerlendirilmiştir. Sınıflandırma işlemi başlamadan önce tüm algoritmalar kapsamında kullanılmak üzere; eğitim ve test kümeleri 10 katlamalı çapraz doğrulama yöntemi ile oluşturulmuştur. Çapraz doğrulama, makine öğrenimi modellerinin becerisini tahmin etmek için kullanılan istatistiksel bir yöntemdir. Sınıflandırma modellerinin değerlendirilmesi ve modelin eğitilmesi için veri setini parçalara ayırmada sıklıkla kullanılmaktadır. k-katlamalı çapraz doğrulama, veriyi belirlenen bir k sayısına göre eşit parçalara bölmekte, her bir parçanın hem eğitim hem de test için kullanılmasını sağlamakta, böylelikle dağılım ve parçalanmadan kaynaklanan sapma ve hataları asgariye indirmektedir. Eğitim ve test kümeleri oluşturulduktan sonra algoritmaların sınıflandırma sonuçları Tablo 5.2 ile gösterilmiştir.

Tablo 5.2. *Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (beşli Likert ölçek, genel)*

Algoritma	Kappa	TP oranı	FP oranı	Kesinlik	F-ölçütü	ROC alanı	Doğruluk	Süre (s)
J48 (C4.5)	0,5616	0,754	0,219	0,744	0,741	0,794	0,7543	0,34
CHAID	0,5585	0,752	0,217	0,741	0,74	0,809	0,7508	0,18
CART	0,5643	0,755	0,214	0,745	**0,744**	0,795	**0,7546**	15,02

Tablo 5.2.(Devam) *Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (beşli Likert ölçek, genel)*

Algoritma	Kappa	TP oranı	FP oranı	Kesinlik	F-ölçütü	ROC alanı	Doğruluk	Süre
NbTREE	0,5521	0,746	0,214	0,734	0,736	0,807	0,7464	2,63
RandomTREE	0,543	0,742	0,221	0,729	0,731	0,787	0,7423	0,17
RepTREE	0,5618	0,753	0,216	0,744	0,742	0,809	0,7534	0,31

Tablo 5.2 incelendiğinde CART algoritmasının en yüksek doğruluk değerine (% 75,46) sahip olduğu görülmüştür. Kesinlik ve duyarlılık ölçütlerinin harmonik ortalaması olan F-ölçütüne bakıldığında benzer şekilde CART algoritmasının dengesiz veri setinde sınıflandırma başarısının diğer algoritmalardan daha iyi olduğu görülmüştür. Kappa değerine bakıldığında bütün algoritmalar için değerin 0.4-0.6 aralığında olduğu görülmüştür. Bundan dolayı kappa istatistiği sonuçları beşli Likert ölçekli veri seti üzerinde gözlenen uyumun tesadüfen gerçekleşmediğini göstermiştir. Algoritmaların sınıflandırma başarılarına ilişkin detaylı sonuçlar Tablo 5.3 ile gösterilmiştir.

Tablo 5.3. *Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (beşli Likert ölçek, sınıf bazlı)*

Algoritma	TP Oran	FP Oran	Kesinlik	F-Ölçütü	ROC Alanı	Sınıf
J48 (C4.5)	0,705	0,065	0,692	0,698	0,836	Çok mutlu
	0,873	0,338	0,78	**0,823**	0,786	Mutlu
	0,593	0,057	0,733	0,656	0,789	Orta
	0,156	0,004	0,568	0,244	0,763	Mutsuz
	0,011	0	0,286	0,02	0,761	Çok mutsuz
CHAID	0,7	0,065	0,69	0,695	0,845	Çok mutlu
	0,869	0,334	0,781	**0,823**	0,802	Mutlu
	0,59	0,058	0,726	0,651	0,81	Orta
	0,167	0,007	0,462	0,246	0,759	Mutsuz
	0,059	0	0,458	0,105	0,69	Çok mutsuz
CART	0,705	0,065	0,693	0,699	0,837	Çok mutlu
	0,87	0,331	0,783	**0,824**	0,787	Mutlu
	0,593	0,056	0,734	0,656	0,789	Orta
	0,186	0,007	0,504	0,271	0,76	Mutsuz
	0,127	0,001	0,471	0,2	0,765	Çok mutsuz
NbTree	0,692	0,063	0,696	0,694	0,846	Çok mutlu
	0,859	0,327	0,782	**0,819**	0,797	Mutlu
	0,6	0,068	0,698	0,645	0,81	Orta
	0,173	0,008	0,448	0,25	0,765	Mutsuz
	0,074	0,002	0,187	0,106	0,801	Çok mutsuz
RepTree	0,689	0,067	0,681	0,685	0,833	Çok mutlu
	0,862	0,338	0,777	**0,817**	0,784	Mutlu
	0,577	0,064	0,704	0,634	0,78	Orta
	0,158	0,008	0,429	0,231	0,695	Mutsuz

Tablo 5.3.(Devam) *Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (beşli Likert ölçek, sınıf bazlı)*

Algoritma	TP Oran	FP Oran	Kesinlik	F-Ölçütü	ROC Alanı	Sınıf
RepTree	0,063	0,002	0,218	0,098	0,601	Çok mutsuz
RandomTree	0,701	0,065	0,691	0,696	0,847	Çok mutlu
	0,87	0,333	0,782	**0,824**	0,8	Mutlu
	0,592	0,057	0,73	0,654	0,811	Orta
	0,191	0,007	0,503	0,277	0,773	Mutsuz
	0,053	0	0,476	0,095	0,761	Çok mutsuz

Tablo 5.3 incelendiğinde tüm algoritmaların mutlu sınıfında yüksek sınıflandırma başarısı elde ettiği görülmüştür.

Sınıflandırma model performans değerlendirmelerinde sıklıkla kullanılan ölçümlerden biri de karışıklık matrisidir. Karışıklık matrisi aynı zamanda hata matrisi olarak da bilinmektedir. Modelin başarısı verilerin doğru veya yanlış sınıflandırılması ile ilgili olduğundan modelin performansını belirleyebilmek için karışıklık matrisi önemlidir. Beşli Likert ölçekli dengesiz veri setinde sınıflandırma başarısı en yüksek olan CART algoritmasının karışıklık matrisi Tablo 5.4 ile gösterilmiştir.

Tablo 5.4. *CART algoritması ile elde edilen karışıklık matrisi*

Çok mutlu	Mutlu	Orta	Mutsuz	Çok mutsuz	Sınıflar
3306	1240	126	12	5	**Çok mutlu**
1143	13740	811	83	13	**Mutlu**
261	1989	3365	61	7	**Orta**
48	504	233	176	2	**Mutsuz**
15	82	43	25	24	**Çok mutsuz**

Tablo 5.4 incelendiğinde “mutlu” sınıfının en çok “çok mutlu” sınıfı ile yanlış sınıflandırıldığı, diğer sınıfların ise en çok “mutlu” sınıfı ile karıştığı görülmüştür. Karışıklık matrisi incelendiğinde veri setinde “mutlu” sınıfı lehine dengesizliğin olduğu bu dengesizliğin algoritmaların sınıflandırma başarılarını olumsuz yönde etkilediği görülmüştür.

5.1.2. Üçlü Likert ölçeğe sahip değişkenler için sınıflandırma sonuçları

Çalışmanın alt amaçlarından biri de üçlü Likert ölçeğe sahip veri seti üzerinde algoritmaların sınıflandırma performanslarını karşılaştırmaktır. Bu kapsamda, TAYA

veri seti üzerinde oluşturulan değişken grubunun ölçeklerinde yapılan revizyon Tablo 5.5 gösterilmiştir.

Tablo 5.5 incelendiğinde eğitim durumunu gösteren öznitelikte ölçekler herhangi bir okula gitmeyen, ilköğretim, lise ve yükseköğretim mezunu olarak tanımlanmış ve bu tanımlamaya uygun veri birleştirilmesi gerçekleştirilmiştir. Beşli Likert ölçek düzeyiyle veri alınan kişisel mutluluk, aile içi mutluluk ve kişisel gelirden memnuniyet düzeylerinde üçlü Likert ölçek kullanılmış ve veri birleştirme işlemi gerçekleştirilmiştir. TAYA'da eşlerin kendi aralarında yaşadıkları sorunların sıklıklarının düzeyini ölçmek amacıyla kullanılan beşli Likert ölçeği üç düzeyli mutluluk Likert ölçeğine dönüştürülmüş ve bu tanımlamaya uygun veri birleştirme işlemi gerçekleştirilmiştir.

Tablo 5.5. *Revize edilmiş değişkenler*

Sıra No	Açıklama	Kategori	Revize Edilmiş Ölçek Değerleri
1	Eğitim durumu	1.Bir okul bitirmedi	1.Bir okul bitirmedi
		2.İlkokul	2. İlköğretim
		3.Genel ortaokul/Mesleki veya teknik ortaokul/İlköğretim	
		4.Genel lise/Mesleki veya teknik lise	3. Lise
		511. 2 veya 3 yıllık yüksekokul	4. Yükseköğretim
		512. 4 yıllık yüksekokul veya fakülte	
		521. 5 veya 6 yıllık fakülte	
		522. Yüksek lisans (5 veya 6 yıllık fakülteler hariç)	
		53. Doktora	
2	Aile mutluluk düzeyi	1. Çok mutlu	1. Mutlu
		2. Mutlu	
		3. Orta	2. Orta
		4. Mutsuz	3. Mutsuz
		5. Çok mutsuz	
3	Kişisel gelirden memnuniyet düzeyi	1. Çok mutlu	1. Mutlu
		2. Mutlu	
		3. Orta	2. Orta
		4. Mutsuz	3. Mutsuz
		5. Çok mutsuz	
4	Ev ile ilgili sorumluluklarda sorun yaşama	1. Hiçbir zaman	1.Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
5	Çocuklarla ilgili sorumluluklarda sorun yaşama	1. Hiçbir zaman	1. Mutlu

Tablo 5.5.(Devam) *Revize edilmiş değişkenler*

Sıra No	Açıklama	Kategori	Revize Edilmiş Ölçek Değerleri
5	Çocuklarla ilgili sorumluluklarda sorun yaşama	2. Nadiren	1. Mutlu
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
6	Ailece birlikte vakit geçirememede sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
7	Harcamalarda sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
8	Giyim tarzında sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
9	Dini görüşlerimizin farklılığında sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
10	Ailesi ile ilişkilerde sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
11	Alkol alışkanlığında sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
12	Sigara alışkanlığında sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
13	Kumar alışkanlığında sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta

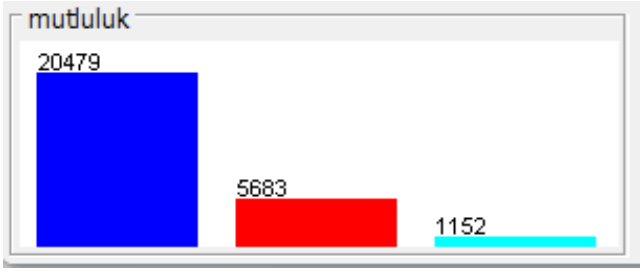
Tablo 5.5.(Devam) *Revize edilmiş değişkenler*

Sıra No	Açıklama	Kategori	Revize Edilmiş Ölçek Değerleri
13	Kumar alışkanlığında sorun yaşama	4. Sıklıkla	3. Mutsuz
		5. Her zaman	
14	İşi ile ilgili sorunların eve taşınmasında sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
15	Bir işte çalışmaması durumunda sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
16	Gelirinin yeterli olmamasında sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
17	Arkadaşlar, görüşülen kişiler konusunda sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
18	Kendine yeterince özen göstermemesi durumunda sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
19	İnternet kullanımında sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
20	Kıskançlık konusunda sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
21	Kültürel farklılıklarda sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	

Tablo 5.5.(Devam) *Revize edilmiş değişkenler*

Sıra No	Açıklama	Kategori	Revize Edilmiş Ölçek Değerleri
22	Kişisel farklılıklarda sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
23	Cinsel uyumsuzluk konusunda sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
24	Siyasi görüş farklılıklarında sorun yaşama	1. Hiçbir zaman	1. Mutlu
		2. Nadiren	
		3. Bazen	2. Orta
		4. Sıklıkla	3. Mutsuz
		5. Her zaman	
	Mutluluk düzeyi	1. Çok mutlu	1. Mutlu
		2. Mutlu	
		3. Orta	2. Orta
		4. Mutsuz	3. Mutsuz
		5. Çok mutsuz	

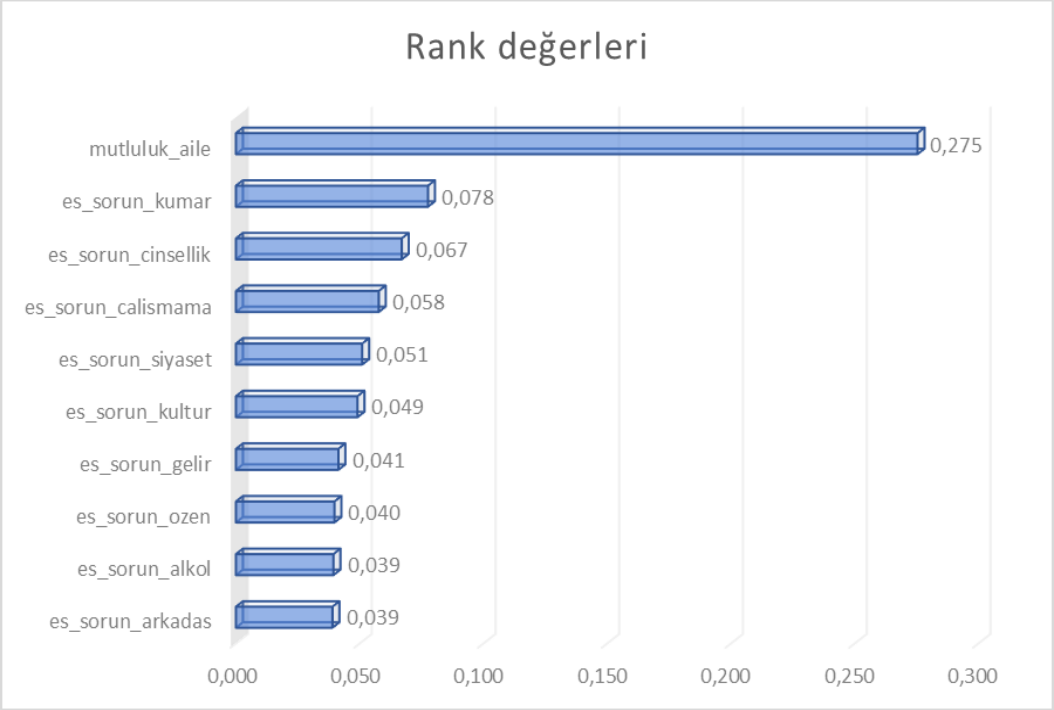
Sınıflandırma algoritmaları için "Mutluluk düzeyi" özniteliği hedef sütun olarak seçilmiştir. Mutluluk düzeyi özniteliğinin sınıf frekansları Şekil 5.3 ile gösterilmiştir. Şekil 5.3 incelendiğinde sırasıyla "mutlu" sınıfında 20479 kişi, "orta" sınıfında 5683 kişi, "mutsuz" sınıfında 1152 kişi bulunmaktadır.



Şekil 5.3. *Üçlü Likert ölçeğine sahip veri setinde mutluluk düzeyine ait sınıf frekansları*

Şekil 5.3 incelendiğinde toplam örnek sayısının hemen hemen çoğu "mutlu" sınıfına ait olduğu görülmektedir. Beşli Likert ölçekli veri seti üzerinde gerçekleştirilen veri ön işleme adımı sonrasında elde edilen üçlü Likert ölçekli veri seti WEKA 3.8.5 paket programında NumericToNominal filtresi kullanılarak sınıfı temsil eden özniteliğin

kategorik olarak temsili sağlanmıştır. Ardından AttributeSelection filtresinde Information Gain (Bilgi Kazancı) algoritması ve Ranker (Sıralama) metodu kullanılarak her bir algoritma için en değerli 10 öznitelik seçilmiştir. Seçilen öznitelikler Şekil 5.4' te sunulmuştur.



Şekil 5.4. *Mutluluk sınıflandırması için bilgi kazancı metodu ile seçilen 10 adet öznitelik (üçlü Likert ölçekli veri setinde)*

Şekil 5.4 incelendiğinde "aile içinde mutluluk" değişkeninin beşli Likert ölçekli veri setinde olduğu gibi sınıflandırma başarısında en etkili öznitelik olduğu görülmüştür. Öznitelik seçme adımından sonra yukarıda bahsi geçen tüm veri madenciliği algoritmaları çalıştırılmış ve bazı temel parametrelere göre genel sınıflandırma sonuçları Tablo 5.6 ve sınıf bazlı sonuçları Tablo 5.7 ile gösterilmiştir.

Tablo 5.6. *Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (üçlü Likert ölçek, genel)*

Algoritma	Kappa	TP oranı	FP oranı	Kesinlik	F-ölçütü	ROC alanı	Doğruluk	Süre (s)
J48 (C4.5)	0.5502	0.843	0.334	0.829	0.828	0.784	0.8426	0.4
CHAID	0.5523	0.843	0.330	0.829	**0.829**	0.801	0.8426	0.17
CART	0.5517	0.843	0.332	0.829	**0.829**	0.779	0.8429	5.97
NbTREE	0.5310	0.831	0.324	0.815	0.819	0.796	0.8310	0.75

Tablo 5.6.(Devam) *Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (üçlü Likert ölçek, genel)*

Algoritma	Kappa	TP oranı	FP oranı	Kesinlik	F-ölçütü	ROC alanı	Doğruluk	Süre (s)
Random TREE	0.5409	0.839	0.337	0.824	0.824	0.784	0.8391	0.18
Rep TREE	0.5520	0.843	0.333	0.830	**0.829**	0.798	**0.8432**	0.34

Tablo 5.6 incelendiğinde üçlü Likert ölçekli veri seti üzerinden elde edilen sonuçlara bakıldığında bütün algoritmaların sınıflandırma başarısının arttığı görülmüş, RepTREE algoritmasının (%84,32) ile en başarılı sonuç verdiği gözlemlenmiştir. F-ölçütünde kapsamında CHAID, CART, RepTREE algoritmalarının sınıflandırmada daha başarılı olduğu gözlemlenmiştir. Kappa değeri ele alındığında bütün algoritmalar için değerin 0,4-0,6 aralığında olduğu görülmüştür. Bundan dolayı Kappa istatistiği sonuçları gözlenen uyumun tesadüfen gerçekleşmediğini göstermiştir.

Tablo 5.7. *Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (üçlü Likert ölçek, sınıf bazlı)*

Algoritma	TP Oran	FP Oran	Kesinlik	F-Ölçütü	ROC Alanı	Sınıf
J48 (C4.5)	0,95	0,43	0,869	**0,908**	0,783	Mutlu
	0,593	0,057	0,733	0,656	0,77	Orta
	0,165	0,005	0,592	0,258	0,75	Mutsuz
CHAID	0,949	0,424	0,87	**0,908**	0,804	Mutlu
	0,594	0,058	0,73	0,655	0,789	Orta
	0,187	0,006	0,586	0,283	0,762	Mutsuz
CART	0,949	0,427	0,869	**0,908**	0,783	Mutlu
	0,593	0,057	0,733	0,656	0,771	Orta
	0,18	0,005	0,597	0,276	0,754	Mutsuz
NbTree	0,932	0,412	0,872	**0,901**	0,8	Mutlu
	0,602	0,072	0,687	0,642	0,788	Orta
	0,168	0,009	0,441	0,244	0,775	Mutsuz
RepTree	0,95	0,428	0,869	**0,908**	0,802	Mutlu
	0,595	0,057	0,733	0,657	0,789	Orta
	0,173	0,005	0,61	0,269	0,772	Mutsuz
RandomTree	0,948	0,433	0,868	**0,906**	0,791	Mutlu
	0,585	0,059	0,723	0,646	0,771	Orta
	0,164	0,006	0,546	0,252	0,719	Mutsuz

Tablo 5.7’ de bütün algoritmalar da en yüksek F ölçütü değerine sahip sınıfın mutlu sınıfı olduğu görülmüştür.

Tablo 5.8. *RepTREE algoritması ile elde edilen karışıklık matrisi*

Mutlu	Orta	Mutsuz	Sınıflar
19450	951	78	**Mutlu**
2253	3381	49	**Orta**
674	279	199	**Mutsuz**

Tablo 5.8 incelendiğinde "mutlu" sınıfının en çok "orta" sınıfı ile yanlış sınıflandırıldığı, "orta" ve " mutsuz" sınıflarının en çok "mutlu" sınıfı ile karıştığı görülmektedir. Karışıklık matrisi incelendiğinde veri setinde "mutlu" sınıfı lehine dengesizliğin olduğu ve bu dengesizliğin algoritmaların sınıflandırma başarısını olumsuz yönde etkilediği görülmüştür.

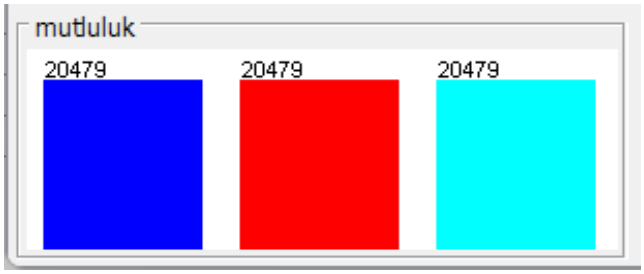
5.2. Dengeli Hale Dönüştürülen Veri Setleri İçin Sınıflandırma Sonuçları

Üçlü Likert ölçekli veri seti üzerinde yapılan uygulama sonuçlarında sınıflar arasında dengesizliğin bulunduğu ve bu dengesizliğin algoritmaların sınıfları doğru sınıflandırmalarında olumsuz yönde etkilediği görülmüştür. Bu bölümde sınıflar arası dengesizliğin ortadan kaldırılması amacıyla önce en yüksek örnek sayısına sahip "mutlu" sınıfı örnek sayısı baz alınarak diğer sınıflarda veri tamamlama işlemi yapılmıştır. Ardından "orta" sınıfı örnek sayısı baz alınarak "mutlu" sınıfı üzerinde yeniden örnekleme (Resample) ve "mutsuz" sınıfı üzerinde veri tamamlama işlemi yapılmıştır. Son olarak "mutsuz" sınıfı örnek sayısı baz alınarak diğer sınıflar üzerinde yeniden örnekleme (Resample) işlemi yapılarak veri setleri dengeli hale dönüştürülmüştür.

5.2.1. En yüksek örnek sayısına sahip mutlu sınıfı örnek sayısı baz alınarak yeniden örneklenen veri setinde mutluluk düzeyi sınıflandırma sonuçları

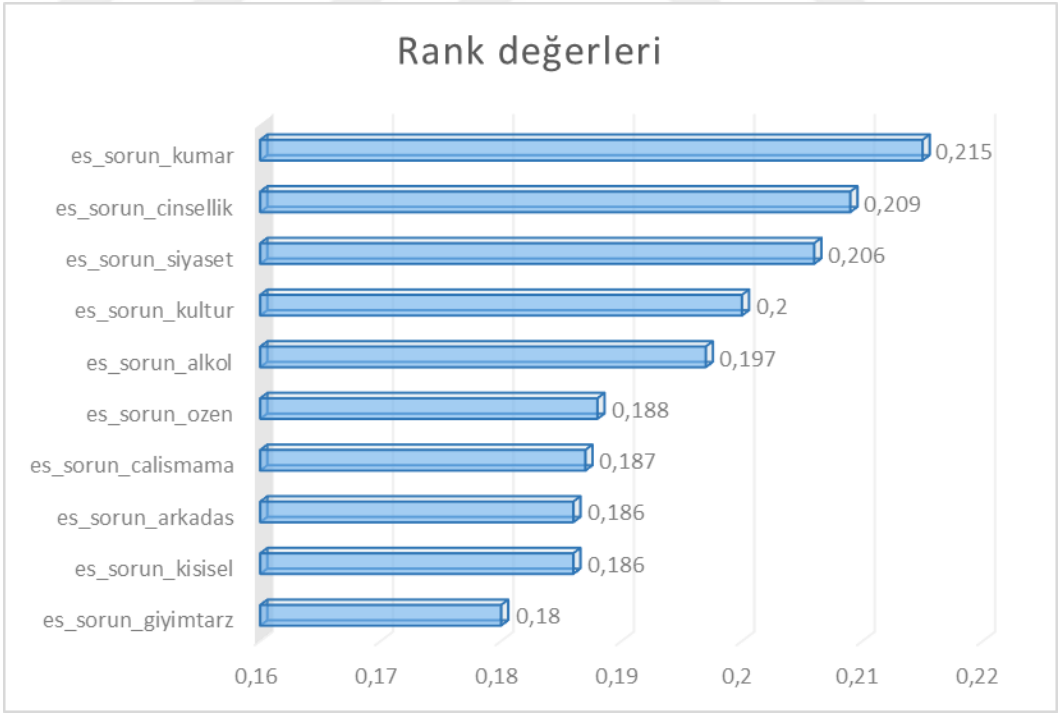
Veri setinde sınıf dengesizliğinin giderilmesi için "orta" ve "mutsuz" sınıflarına ait örnek sayıları artırılarak mutlu sınıfının örnek sayısına eşitlenmiştir. Şekil 5.5 incelendiğinde "mutlu" sınıfı örnek sayısı baz alınarak 5683 örnek sayısına sahip "orta" sınıfında 14796 adet ve 1152 örnek sayısına sahip "mutsuz" sınıfında 19327 adetlik veri

tamamlama işlemi uygulanmıştır. Şekil 5.5'te görüldüğü üzere sınıflar toplam örnek sayısı 61437 olmuştur.



Şekil 5.5. *En çok örneğe sahip sınıf örnek sayısı baz alınarak elde edilen veri setine ait mutluluk düzeyi sınıf frekansları*

Oluşturulan veri seti WEKA 3.8.5 paket programında NumericToNominal filtresi kullanılarak sınıfı temsil eden özniteliğin kategorik olarak temsili sağlanmıştır. Ardından AttributeSelection filtresinde Information Gain (Bilgi Kazancı) algoritması ve Ranker (Sıralama) metodu kullanılarak her bir algoritma için en değerli 10 öznitelik seçilmiştir. Seçilen öznitelikler Şekil 5.6'da sunulmuştur.



Şekil 5.6. *En yüksek sınıf örneğine eşitlenmiş veri setinde mutluluk sınıflandırması için bilgi kazancı metodu ile seçilen 10 adet öznitelik (üçlü Likert ölçekli)*

Şekil 5.6 incelendiğinde "eşler arası kumar sorunu" değişkeninin sınıflandırma başarısında en etkili öznitelik olduğu görülmüştür. Veri seti ile ilgili veri madenciliği

algoritmalarının bazı temel parametrelere göre genel sınıflandırma sonuçları Tablo 5.9 ve sınıf bazlı sonuçları Tablo 5.10 ile gösterilmiştir.

Tablo 5.9. *Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (mutlu sınıfı örnek sayısı baz alınan, genel)*

Algoritma	Kappa	TP oranı	FP oranı	Kesinlik	F-ölçütü	ROC alanı	Doğruluk	Süre(s)
J48 (C4.5)	0,4354	0,624	0,188	0,584	0,584	0,77	0,6236	0,94
CHAID	0,4588	0,639	0,18	0,591	0,553	**0,783**	0,6391	0,29
CART	0,4605	0,64	0,18	0,594	0,536	0,781	**0,6403**	39,22
NbTREE	0,4432	0,629	0,189	0,585	0,582	0,782	0,6288	25,33
RandomTREE	0,4115	0,608	0,196	0,582	0,588	0,682	0,6076	0,33
RepTREE	0,437	0,625	0,188	0,587	0,59	0,776	0,6247	0,8

Tablo 5.9 incelendiğinde CART algoritmasının en yüksek doğrulukla sınıflandırma başarısı elde ettiği görülmüştür. Sınıf dengesizliği giderilmek üzere uygulanan işlem sonrası elde edilen sonuçlar, üçlü Likert ölçekli dengesiz veri seti sonuçlarıyla karşılaştırıldığında Kappa istatistiği, Kesinlik, F ölçütü, ROC alanı ve Doğruluk değerlerinde düşüş gözlemlenmiştir.

Tablo 5.10. *Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (mutlu sınıfı örnek sayısı baz alınan, sınıf bazlı)*

Algoritma	TP Oran	FP Oran	Kesinlik	F-Ölçütü	ROC Alanı	Sınıf
J48 (C4.5)	0,978	0,155	0,759	**0,855**	**0,924**	Mutlu
	0,22	0,143	0,435	0,292	0,618	Orta
	0,673	0,266	0,558	0,61	0,769	Mutsuz
CHAID	0,977	0,156	0,759	**0,854**	**0,926**	Mutlu
	0,073	0,044	0,455	0,126	0,635	Orta
	0,867	0,342	0,559	0,68	0,786	Mutsuz
CART	0,974	0,156	0,757	**0,852**	**0,921**	Mutlu
	0,036	0,02	0,469	0,066	0,636	Orta
	0,911	0,363	0,557	0,691	0,787	Mutsuz
NbTree	0,969	0,147	0,767	**0,857**	**0,927**	Mutlu
	0,18	0,12	0,429	0,253	0,634	Orta
	0,737	0,29	0,559	0,636	0,788	Mutsuz
RepTree	0,976	0,155	0,759	**0,854**	**0,927**	Mutlu
	0,237	0,15	0,441	0,308	0,621	Orta
	0,661	0,258	0,562	0,607	0,779	Mutsuz
RandomTree	0,981	0,161	0,753	**0,852**	**0,922**	Mutlu
	0,371	0,241	0,435	0,401	0,522	Orta
	0,47	0,186	0,558	0,511	0,603	Mutsuz

Tablo 5.10'da bütün algoritmalar da en yüksek F ölçütü değerine sahip sınıfın mutlu sınıfı olduğu görülmüştür.

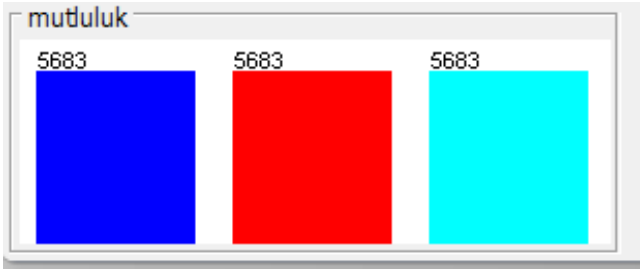
Tablo 5.11. *CART algoritması ile elde edilen karışıklık matrisi*

Mutlu	Orta	Mutsuz	Sınıflar
19952	199	328	**Mutlu**
5213	730	14536	**Orta**
1195	626	18658	**Mutsuz**

Tablo 5.11 incelendiğinde "mutlu" ve "mutsuz" sınıflarında yüksek oranda doğru sınıflandırma yapıldığı görülürken "orta" sınıfının en çok "mutsuz" sınıfı ile karıştığı görülmektedir.

5.2.2. Orta mutluluk düzeyi sınıf örnek sayısı baz alınarak yeniden örneklenen veri setinde mutluluk düzeyi sınıflandırma sonuçları

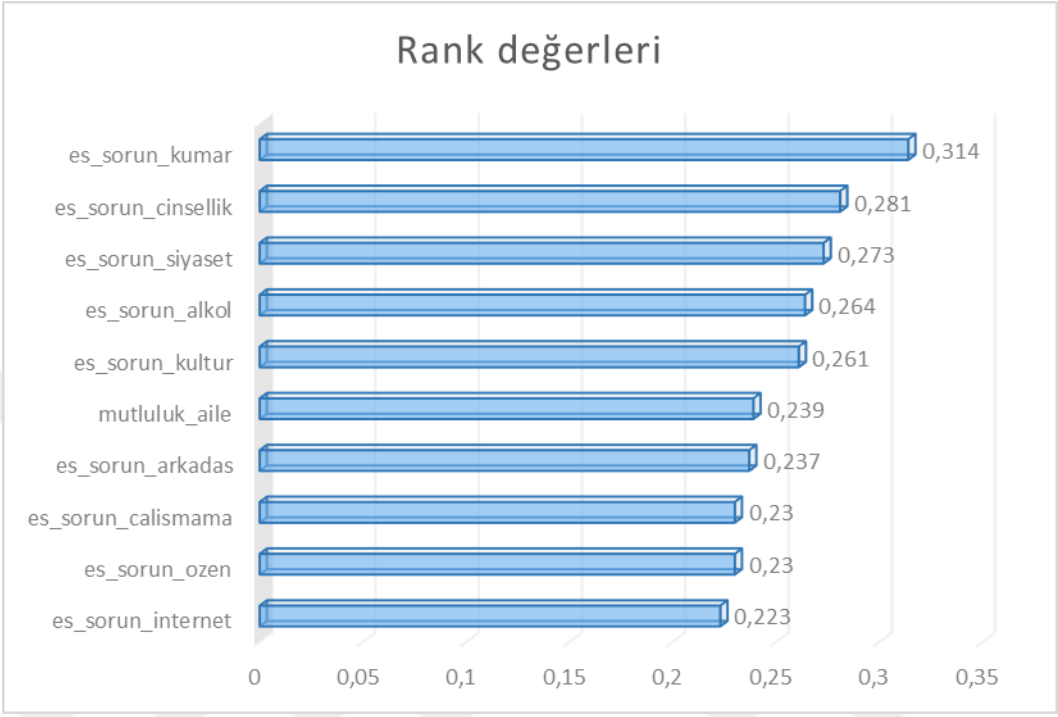
Veri setinde "orta" sınıf örnek sayısı baz alındığında, "mutlu" sınıfı örnek sayısı yeniden örnekleme yöntemi yardımıyla "mutsuz" sınıfı örnek sayısı ise veri tamamlama yöntemi yardımıyla "orta" sınıfı örnek sayısına eşitlenmiştir. Şekil 5.7 incelendiğinde "orta" sınıfı örnek sayısı baz alınarak, 20479 örnek sayısına sahip "mutlu" sınıfı içerisinden rastgele yeniden örnekleme yöntemi ile 5683 adet alınmış ve 1152 örnek sayısına sahip "mutsuz" sınıfında 4531 adetlik veri tamamlama işlemi uygulanmıştır. Şekil 5.7'te görüldüğü üzere sınıfların toplam örnek sayısı 17049 olmuştur.



Şekil 5.7. *Mutluluk düzeylerinden Orta sınıfı örnek sayısı baz alınarak yeniden örneklenen veri setine ait mutluluk düzeyi frekansları*

Sınıf dengesizliğinin ortadan kaldırılması amacıyla oluşturulan veri setlerinden ikincisi olan "orta" sınıfı örnek sayısı baz alınarak oluşturulan veri seti WEKA 3.8.5 paket programında NumericToNominal filtresi kullanılarak sınıfı temsil eden özniteliğin

kategorik olarak temsili sağlanmıştır. Ardından AttributeSelection filtresinde Information Gain (Bilgi Kazancı) algoritması ve Ranker (Sıralama) metodu kullanılarak her bir algoritma için en değerli 10 öznitelik seçilmiştir. Seçilen öznitelikler Şekil 5.8'de sunulmuştur.



Şekil 5.8. *Orta sınıf örneğine eşitlenmiş veri setinde mutluluk sınıflandırması için bilgi kazancı metodu ile seçilen 10 adet öznitelik (üçlü Likert ölçekli)*

Şekil 5.8 incelendiğinde "eşler arası kumar sorunu" değişkeninin sınıflandırma başarısında en etkili öznitelik olduğu görülmüştür. Veri seti ile ilgili veri madenciliği algoritmalarının bazı temel parametrelere göre genel sınıflandırma sonuçları Tablo 5.12 ve sınıf bazlı sonuçları Tablo 5.13 ile gösterilmiştir.

Tablo 5.12. *Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (orta sınıfı örnek sayısı baz alınan, genel)*

Algoritma	Kappa	TP oranı	FP oranı	Kesinlik	F-ölçütü	ROC alanı	Doğruluk	Süre(s)
J48 (C4.5)	0,6768	0,785	0,108	0,798	0,78	0,873	0,7845	0,38
CHAID	0,6693	0,78	0,11	0,794	0,775	**0,878**	0,7786	0,19
CART	0,6777	0,785	0,107	0,799	0,781	0,876	**0,7851**	4,74
NbTREE	0,6453	0,764	0,118	0,784	0,761	0,876	0,7636	4,29
RandomTREE	0,6664	0,778	0,111	0,792	0,774	0,864	0,7776	0,15
RepTREE	0,6739	0,783	0,109	0,797	0,779	0,877	0,7826	0,23

Tablo 5.12 incelendiğinde CART algoritmasının bu dengeli veri setine ait sınıflandırma sonuçlarında da en yüksek doğrulukla sınıflandırma başarısı elde ettiği görülmüştür. Sınıf dengesizliği giderilmek üzere uygulanan işlem sonrası elde edilen sonuçlar, üçlü Likert ölçekli dengesiz veri seti sonuçlarıyla karşılaştırıldığında Kappa istatistiği ve ROC alanı değerlerinde artış, Kesinlik, F ölçütü ve Doğruluk değerlerinde düşüş gözlemlenmiştir.

Tablo 5.13. *Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (orta sınıfı örnek sayısı baz alınan, sınıf bazlı)*

Algoritma	TP Oran	FP Oran	Kesinlik	F-Ölçütü	ROC Alanı	Sınıf
J48 (C4.5)	0,897	0,2	0,691	0,781	0,866	Mutlu
	0,589	0,063	0,825	0,687	0,829	Orta
	0,867	0,06	0,878	**0,873**	**0,924**	Mutsuz
CHAID	0,9	0,206	0,687	0,779	0,874	Mutlu
	0,579	0,063	0,821	0,679	0,833	Orta
	0,86	0,062	0,874	**0,867**	**0,927**	Mutsuz
CART	0,894	0,201	0,69	0,779	0,868	Mutlu
	0,593	0,062	0,828	0,691	0,835	Orta
	0,869	0,06	0,879	**0,874**	**0,926**	Mutsuz
NbTree	0,902	0,236	0,656	0,76	0,874	Mutlu
	0,572	0,073	0,797	0,666	0,827	Orta
	0,816	0,045	0,9	**0,856**	**0,926**	Mutsuz
RepTree	0,896	0,204	0,687	0,778	0,87	Mutlu
	0,589	0,063	0,823	0,686	0,833	Orta
	0,863	0,059	0,88	**0,871**	**0,927**	Mutsuz
RandomTree	0,9	0,209	0,683	0,777	0,867	Mutlu
	0,594	0,075	0,799	0,681	0,826	Orta
	0,839	0,05	0,893	**0,865**	**0,9**	Mutsuz

Tablo 5.13'te daha önce yapılan diğer uygulama sonuçlarından farklı olarak bütün algoritmalar da en yüksek F ölçütü değerine sahip sınıfın mutsuz sınıfı olduğu görülmüştür.

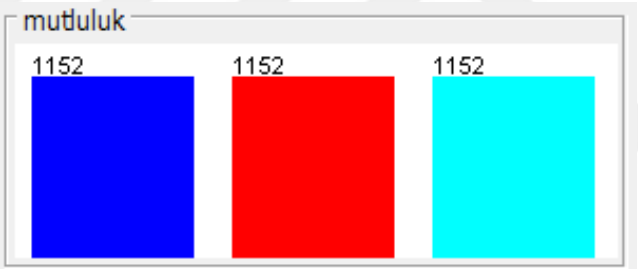
Tablo 5.14. *CART algoritması ile elde edilen karışıklık matrisi*

Mutlu	Orta	Mutsuz	Sınıflar
5079	343	261	**Mutlu**
1894	3369	420	**Orta**
386	359	4938	**Mutsuz**

Tablo 5.14 incelendiğinde “mutlu” ve “mutsuz” sınıflarında yüksek oranda doğru sınıflandırma yapıldığı görülürken “orta” sınıfının en çok “mutlu” sınıfı ile karıştığı görülmektedir.

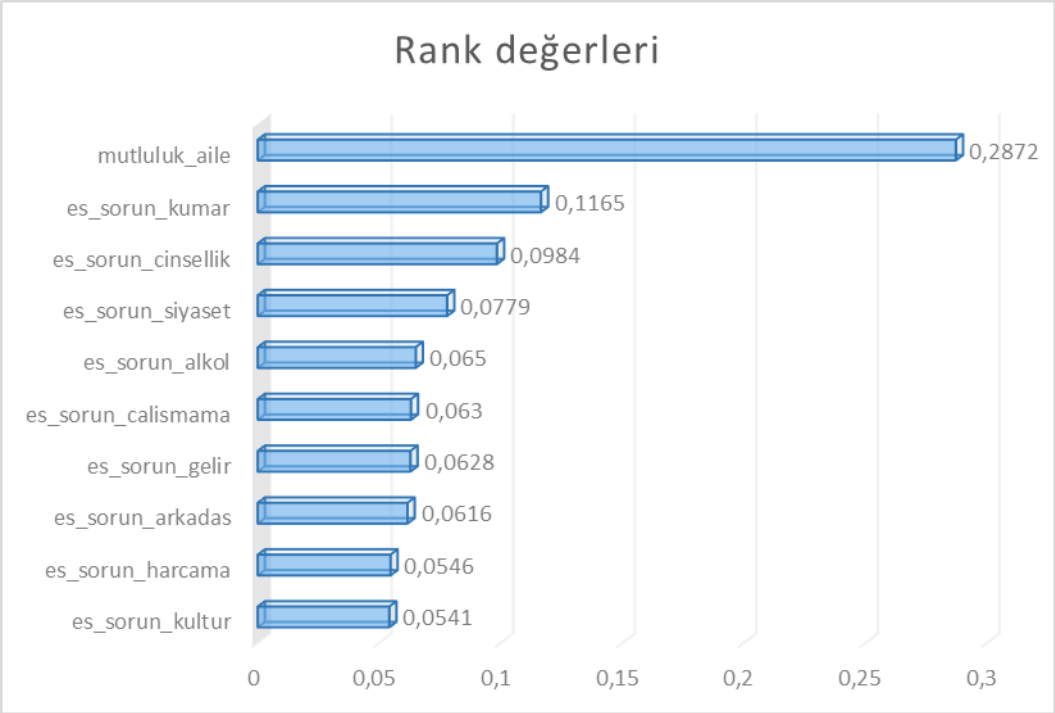
5.2.3. En düşük örnek sayısına sahip mutsuz sınıfı örnek sayısı baz alınarak yeniden örneklenen veri setinde mutluluk düzeyi sınıflandırma sonuçları

Veri setinde “mutlu” ve “orta” sınıfı örnek sayıları rastgele yeniden örnekleme yöntemiyle “mutsuz” sınıfının örnek sayısına eşitlenmiştir. Şekil 5.9 incelendiğinde “mutsuz” sınıfı örnek sayısı baz alınarak, 20479 örnek sayısına sahip “mutlu” sınıfı içerisinden 1152 adet örnek alınmış ve 5683 örnek sayısına sahip “orta” sınıfı içerisinden 1152 adet örnek alınarak veri seti oluşturulmuştur. Şekil 5.9’ da görüldüğü üzere sınıfların toplam örnek sayısı 3456 olmuştur.



Şekil 5.9. *Mutluluk düzeylerinden Mutsuz sınıfı örnek sayısı baz alınarak yeniden örneklenen veri setine ait mutluluk düzeyi frekansları*

Veri seti üzerinde “mutsuz” sınıfı örnek sayısı baz alınarak oluşturulan dengeli veri seti WEKA 3.8.5 paket programında NumericToNominal filtresi kullanılarak sınıfı temsil eden özniteliğin kategorik olarak temsili sağlanmıştır. Ardından AttributeSelection filtresinde Information Gain (Bilgi Kazancı) algoritması ve Ranker (Sıralama) metodu kullanılarak her bir algoritma için en değerli 10 öznitelik seçilmiştir. Seçilen öznitelikler Şekil 5.10’da sunulmuştur.



Şekil 5.10. *Mutsuz sınıf örneğine eşitlenmiş veri setinde mutluluk sınıflandırması için bilgi kazancı metodu ile seçilen 10 adet öznitelik (üçlü Likert ölçekli)*

Şekil 5.10 incelendiğinde "aile içi mutluluk" değişkeninin sınıflandırma başarısında en etkili öznitelik olduğu görülmüştür. Veri seti ile ilgili veri madenciliği algoritmalarının bazı temel parametrelere göre genel sınıflandırma sonuçları Tablo 5.15 ve sınıf bazlı sonuçları Tablo 5.16 ile gösterilmiştir.

Tablo 5.15. *Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (mutsuz sınıfı örnek sayısı baz alınan, genel)*

Algoritma	Kappa	TP oranı	FP oranı	Kesinlik	F-ölçütü	ROC alanı	Doğruluk	Süre(s)
J48 (C4.5)	0,4805	0,654	0,173	0,704	0,642	0,762	0,6536	0,03
CHAID	0,4786	0,652	0,174	0,688	0,643	0,779	0,6522	0,01
CART	0,4848	0,657	0,172	0,693	0,648	0,772	**0,6565**	1,05
NbTREE	0,4683	0,646	0,177	0,663	0,64	**0,78**	0,6455	1,25
RandomTREE	0,4661	0,644	0,178	0,677	0,634	0,755	0,6441	0,01
RepTREE	0,4818	0,655	0,173	0,69	0,646	0,775	0,6545	0,07

Tablo 5.15 incelendiğinde CART algoritmasının bu dengeli veri setine ait sınıflandırma sonuçlarında da en yüksek doğrulukla sınıflandırma başarısı elde ettiği görülmüştür. Sınıf dengesizliği giderilmek üzere uygulanan işlem sonrası elde edilen sonuçlar, üçlü Likert ölçekli dengesiz veri seti sonuçlarıyla karşılaştırıldığında Kappa

istatistiği, ROC alanı, Kesinlik, F ölçütü ve Doğruluk değerlerinde düşüş gözlemlenmiştir.

Tablo 5.16. *Algoritmaların sınıflandırma sonuçlarına ilişkin bilgiler (mutsuz sınıfı örnek sayısı baz alınan, sınıf bazlı)*

Algoritma	TP Oran	FP Oran	Kesinlik	F-Ölçütü	ROC Alanı	Sınıf
J48 (C4.5)	0,931	0,345	0,574	**0,71**	**0,79**	Mutlu
	0,574	0,138	0,675	0,62	0,75	Orta
	0,457	0,037	0,861	0,597	0,745	Mutsuz
CHAID	0,903	0,322	0,584	**0,709**	**0,811**	Mutlu
	0,575	0,146	0,663	0,616	0,766	Orta
	0,48	0,054	0,817	0,604	0,761	Mutsuz
CART	0,911	0,329	0,581	**0,709**	**0,802**	Mutlu
	0,564	0,129	0,686	0,619	0,761	Orta
	0,494	0,057	0,813	0,614	0,755	Mutsuz
NbTree	0,843	0,286	0,596	**0,698**	**0,809**	Mutlu
	0,559	0,159	0,638	0,596	0,761	Orta
	0,535	0,087	0,755	0,626	0,77	Mutsuz
RepTree	0,905	0,325	0,582	**0,709**	**0,806**	Mutlu
	0,564	0,137	0,674	0,614	0,761	Orta
	0,494	0,056	0,814	0,615	0,757	Mutsuz
RandomTree	0,905	0,329	0,579	**0,706**	**0,8**	Mutlu
	0,551	0,143	0,658	0,6	0,738	Orta
	0,477	0,061	0,796	0,596	0,728	Mutsuz

Tablo 5.16'da bütün algoritmalar da en yüksek F ölçütü değerine sahip sınıfın mutlu sınıfı olduğu görülmüştür.

Tablo 5.17. *CART algoritması ile elde edilen karışıklık matrisi*

Mutlu	Orta	Mutsuz	Sınıflar
1050	47	55	**Mutlu**
426	650	76	**Orta**
332	251	569	**Mutsuz**

Tablo 5.17 incelendiğinde "mutlu" sınıfının yüksek oranda doğru sınıflandırıldığı, "orta" ve "mutsuz" sınıfının en çok "mutlu" sınıfı ile karıştığı görülmektedir.

Tablo 5.18. *Üçlü Likert ölçekli veri setlerinde CART algoritmasına ait sınıf bazlı sonuçları*

Veri Seti	TP Oran	FP Oran	Kesinlik	F-Ölçütü	ROC Alanı	Sınıf
Üçlü Likert Ölçekli Dengesiz Veri Seti	0,949	0,427	0,869	**0,908**	**0,783**	**Mutlu**
	0,593	0,057	0,733	0,656	0,771	**Orta**
	0,18	0,005	0,597	0,276	0,754	**Mutsuz**

Tablo 5.18. (Devam) *Üçlü Likert ölçekli veri setlerinde CART algoritmasına ait sınıf bazlı sonuçlar*

Veri Seti	TP Oran	FP Oran	Kesinlik	F-Ölçütü	ROC Alanı	Sınıf
"Mutlu" Sınıfı Örnek Sayısı Baz Alınan Dengeli Veri Seti	0,974	0,156	0,757	**0,852**	**0,921**	**Mutlu**
	0,036	0,02	0,469	0,066	0,636	**Orta**
	0,911	0,363	0,557	0,691	0,787	**Mutsuz**
"Orta" Sınıfı Örnek Sayısı Baz Alınan Dengeli Veri Seti	0,894	0,201	0,69	0,779	0,868	**Mutlu**
	0,593	0,062	0,828	0,691	0,835	**Orta**
	0,869	0,06	0,879	**0,874**	**0,926**	**Mutsuz**
"Mutsuz" Sınıfı Örnek Sayısı Baz Alınan Dengeli Veri Seti	0,911	0,329	0,581	**0,709**	**0,802**	**Mutlu**
	0,564	0,129	0,686	0,619	0,761	**Orta**
	0,494	0,057	0,813	0,614	0,755	**Mutsuz**

Üçlü Likert ölçekli dengesiz veri seti ile yeniden örnekleme ve veri tamamlama yöntemleri kullanılarak dengeli hale dönüştürülen üç farklı veri setinde CART algoritmasına ait sınıf bazlı sonuçlar Tablo 5.18'de yer almaktadır. Tablo 5.18 incelendiğinde üçlü Likert ölçekli veri setinde bulunan sınıflara ait TP metrik oranları ve F-ölçütü değerleri karşılaştırıldığında "mutlu" sınıfının en yüksek başarı ile "mutsuz" sınıfının ise en düşük başarı ile sınıflandırılan kategori olduğu görülmektedir.

Üçlü Likert ölçekli dengesiz veri setine ait sonuçlar ile sınıflar arasında bulunan dengesizliğin giderilmesi amacıyla oluşturulan veri setlerine ait sonuçlar karşılaştırıldığında;

"Orta" sınıfı örnek sayısı baz alınan veri setinde "mutsuz" sınıfının en yüksek F-ölçütü ve ROC alanı değeriyle diğer veri setlerine ait sonuçlardan farklı olarak CART algoritması tarafından en başarılı sınıflandırılan sınıf olduğu görülmüştür.

6. TARTIŞMA VE SONUÇ

Bu tez çalışmasının amacı, veri madenciliği algoritmalarının üçlü ve beşli Likert (Ordinal) ölçeğine sahip veri seti üzerindeki sınıflandırma performansını ölçümlemek, sınıflar arası dengesizliğin olduğu veri seti üzerinde sınıflandırma performanslarını karşılaştırmak ve dengesizliğin giderildiği durumda algoritmaların sınıflandırma performansının nasıl değiştiğini ortaya koymaktır. Bu nedenle çalışmada, veri seti olarak Türkiye İstatistik Kurumu Başkanlığı tarafından yürütülen Türkiye Aile Yapısı Araştırması (TAYA) seçilmiştir.

İki aşamada gerçekleşen deneylerden ilkinde genel memnuniyet ifade eden özniteliklerden en değerli 10 öznitelikler bulunmuştur. İkinci aşamada ise mutluluk sınıflandırması değerlendirilmiş, yeniden örnekleme ve veri tamamlama yöntemleri ile sınıflandırma başarı yüzdesinin nasıl değiştiği incelenmiştir.

Özniteliklerden "Genel mutluluk durumu düşünüldüğünde hangisi ailenizi en iyi ifade eder?" özniteliği beşli Likert ölçekli dengesiz veri seti, üçlü Likert ölçekli dengesiz veri seti ve "mutsuz" sınıfı örnek sayısı baz alındığında yeniden örnekleme ile oluşturulan dengeli veri seti için en yüksek rank değerine sahip olduğu görülmüştür. Paralel olarak "mutlu" sınıfı örnek sayısı baz alınarak oluşturulan dengeli veri seti ve "orta" sınıfı örnek sayısı baz alınarak oluşturulan dengeli veri seti için en yüksek rank değeri "Eşinizle kumar alışkanlığı sebebiyle ne sıklıkla sorun yaşadınız?" özniteliğine aittir.

Öznitelik seçimi sonrası veri madenciliği teknikleri kullanılarak sınıflandırma analizi gerçekleştirilmiştir. Beşli Likert ölçekli dengesiz veri seti için mutluluk düzeyi özniteliği J48 (C4.5), CHAID, CART, NbTREE, RandomTREE ve RepTREE algoritmaları ile sınıflandırılmış ve sınıflandırma başarıları sırasıyla %75,43, %75,08, %75,46, %74,64, %74,23 ve %75,34 olarak bulunmuş ve sınıflar arası dengesizliğin ve kategori sayısının fazla olduğu veri seti türünde CART algoritmasının daha başarılı sınıflandırma yaptığı gözlemlenmiştir. Sınıflandırma başarısının yanı sıra Kappa istatistiği ve dengesiz veri setleri için F-ölçütü parametreleri de karşılaştırılmıştır. Kappa istatistiğine göre tüm algoritmalar için sınıflar arasında orta düzeyde uyum olduğu görülmüş olup F-ölçütü parametresine göre CART algoritmasının diğer algoritmalara göre daha doğru sınıflandırma yaptığı gözlemlenmiştir.

Üçlü Likert ölçekli dengesiz veri seti için mutluluk düzeyi özniteliği aynı algoritmalar ile sınıflandırılmış ve sınıflandırma başarıları sırasıyla %84,26, %84,26, %84,29, %83,10, %83,91 ve %84,32 olarak bulunmuş ve sınıflar arası dengesizliğin ve

kategori sayısının az olduğu veri seti türünde RepTREE algoritmasının daha başarılı sınıflandırma yaptığı gözlemlenmiştir. Sınıflandırma başarısının yanı sıra Kappa istatistiği ve dengesiz veri setleri için F-ölçütü parametreleri de karşılaştırılmıştır. Kappa istatistiğine göre tüm algoritmalar için sınıflar arasında orta düzeyde uyum olduğu görülmüş olup F-ölçütü parametresine göre CART, CHAID, RepTREE algoritmasının diğer algoritmalara göre daha doğru sınıflandırma yaptığı gözlemlenmiştir.

"Mutlu" sınıfı örnek sayısı baz alınarak diğer sınıflar üzerinde veri tamamlama yöntemiyle dengeli hale dönüştürülen veri seti için mutluluk düzeyi özniteliği aynı algoritmalar ile sınıflandırılmış ve sınıflandırma başarıları sırasıyla %62,36, %63,91, %64,03, %62,88, %60,76 ve %62,47 olarak bulunmuş ve dengeli veri seti türünde CART algoritmasının daha başarılı sınıflandırma yaptığı gözlemlenmiştir. Sınıflandırma başarısının yanı sıra Kappa istatistiği ve dengeli veri setleri için ROC alanı parametreleri de karşılaştırılmıştır. Kappa istatistiğine göre tüm algoritmalar için sınıflar arasında orta düzeyde uyum olduğu görülmüştür. ROC eğrisi altındaki alanlar incelendiğinde en başarılı sınıf ayrımı yapan algoritma CHAID algoritması olmuştur.

"Orta" sınıfı örnek sayısı baz alınarak diğer sınıflar üzerinde veri tamamlama ve yeniden örnekleme yöntemleriyle dengeli hale dönüştürülen veri seti için mutluluk düzeyi özniteliği aynı algoritmalar ile sınıflandırılmış ve sınıflandırma başarıları sırasıyla %78,45, %77,86, %78,51, %76,36, %77,76 ve %78,26 olarak bulunmuş ve dengeli veri seti türünde CART algoritmasının daha başarılı sınıflandırma yaptığı gözlemlenmiştir. Kappa istatistiğine göre tüm algoritmalar için sınıflar arasında iyi düzeyde uyum olduğu görülmüştür. ROC eğrisi altındaki alanlar incelendiğinde en başarılı sınıf ayrımı yapan algoritma CHAID algoritması olmuştur.

"Mutsuz" sınıfı örnek sayısı baz alınarak diğer sınıflar üzerinde yeniden örnekleme yöntemiyle dengeli hale dönüştürülen veri seti için mutluluk düzeyi özniteliği aynı algoritmalar ile sınıflandırılmış ve sınıflandırma başarıları sırasıyla %65,36, %65,22, %65,65, %64,55, %64,41 ve %65,45 olarak bulunmuş ve dengeli veri seti türünde CART algoritmasının daha başarılı sınıflandırma yaptığı gözlemlenmiştir. Kappa istatistiğine göre tüm algoritmalar için sınıflar arasında orta düzeyde uyum olduğu görülmüştür. ROC eğrisi altındaki alanlar incelendiğinde en başarılı sınıf ayrımı yapan algoritma CHAID ve NbTREE algoritmaları olmuştur.

Yapılan çalışmalar sonucunda;

- CART algoritmasının üçlü Likert ölçekli veri seti hariç tüm durumlarda en başarılı sınıflandırma sonuçları verdiği görülmüştür. Anket verilerinin sınıflandırılması probleminde CART algoritmasının kullanılmasının daha uygun olacağı düşünülmektedir.
- Veri setindeki kategori sayısı arttıkça sınıflandırma başarısı ters orantılı olarak etkilenmekte olduğu görülmüştür.
- Likert ölçekli veri setlerinde sınıflar arası dengesizlik giderildiğinde FP metriğinin dengesiz veri setlerine göre daha düşük olduğu görülmüştür. Bir başka ifadeyle dengeli veri setlerinde I. Tip hata yapma oranı daha düşüktür.
- Sınıflar arası dengesizlik söz konusu olduğunda kesinlik parametresi kategori sayısı az olan veri setinde daha anlamlı olduğu görülmüştür. Dengesizliği giderilen veri setlerinde sınıf örnek sayısı ortalama örnek hacmine yakın olan veri setinde daha yüksek kesinlik değeri elde edilmiştir.
- Dengesiz veri setine ait sonuçlarda "mutsuz" sınıfının en düşük TP oranlarına sahip olduğu görülmüştür. Veri seti dengeli hale dönüştürüldüğünde TP oranlarının "mutsuz" sınıfı lehine arttığı gözlemlenmiştir.

Ek olarak gerçekleştirilen deneyler sayesinde Türk aile yapısının demografik çerçevesini ve evli bireylerin mutluluk düzeylerine katkı sağlayan alanların incelenmesi sağlanmıştır. Elde edilen rank değerleri incelendiğinde evli çiftlerin kişisel mutluluk düzeylerini etkileyen faktörlerin başında aile içi mutluluk düzeyi ve kumar alışkanlığından dolayı sorun yaşanması olduğu görülmüştür. Ailesi mutlu olan bireylerin çoğunluğunun kişisel olarak mutlu olduğu gözlemlenmiştir. Ailesi mutsuz olan bireyler de ise eşler arasında kültürel farklılıklar, siyasi görüş farklılıkları, cinsel uyumsuzluk ve alkol alışkanlığı durumları bireyin kişisel mutluluğunu olumsuz yönde etkileyen faktörler olarak belirlenmiştir.

Çalışma sayesinde veri madenciliği teknikleri kullanılarak dengeli/dengesiz Likert ölçekli yapıya sahip verileri incelemenin mümkün olduğu gösterilmiştir. Türkiye Aile Yapısı Araştırması verilerinin dengesiz veri setleri için geliştirilen diğer algoritmalarla incelenmesi hususu başka çalışmalara konu olarak önerilmektedir. Ayrıca edinilen bilgilere dayalı olarak anket sorularında yapılabilecek iyileştirmeler veya eklenebilecek yeni sorular ile sınıflandırma başarısının artırılmasına olanak sağlayacağı düşünülmektedir.

KAYNAKÇA

[1] Salzberg, S. L., Searls, D. B., Kasif, S. (1998). Computational Methods in Molecular Biology. Amsterdam: Elsevier Sciences B.V.

[2] Gundecha, P., Liu, H. (2012). Mining Social Media: A Brief Introduction.

[3] Tan, P. N., Steinbach, M., Kumar, V. (2006). Introduction to Data Mining. Pearson: Addison Wesley, Boston.

[4] Zaki, M. J., Meira, Jr. W. (2014). Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge: Cambridge University Press.

[5] Ozer, P., Sprinkhuizen-Kuyper, I. G. (2008). Data algorithms for classification, BSc Thesis Artificial Intelligence, Radbound University Nijmegen.

[6] García, E., Romero, C., Ventura, S., Calders, T. (2007). Drawbacks and solutions of applying association rule mining in learning management systems. CEUR Workshop Proceedings. 305. 13-22.

[7] Han, J., Kamber, M., Pei, J. (2012). Data mining: Concepts and techniques, third edition (3rd ed.). Morgan Kaufmann Publishers.

[8] Srivastava, A., Han, E. H., Kumar, V., Singh, V. (1999). Parallel Formulations of Decision-Tree Classification Algorithms. Data Mining and Knowledge Discovery 3, 237–261.

[9] Sohn, S., Moon, T. (2004). Decision Tree based on data envelopment analysis for effective technology commercialization. Expert Systems with Applications. 26. 279-284.

[10] Koufakou, A., Gosselin, J., Guo, D. (2005). "Using data mining to extract knowledge from student evaluation comments in undergraduate courses", International *Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada, pp. 3138-3142.

[11] Bresfelean, V. (2007). Analysis and Predictions on Students' Behavior Using Decision Trees in Weka Environment. 51 - 56.

[12] Mardikyan, S., Badur, B. (2011). "Analyzing Teaching Performance of Instructors Using Data Mining Techniques", Informatics in Education, Vol. 10, No. 2, pp. 245–257.

[13] 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). İÜ, Sosyal Bilimler Enstitüsü, İstanbul.

[14] Fokoue, E., Gündüz, N. (2013). "Data Mining and Machine Learning Techniques for Extracting Patterns in Students' Evaluations of Instructors", Rochester Institute of Technology RIT Scholar Works, pp.1-26.

[15] Alhendawi, K. M., Baharudin, A. S. (2014). A Classification Model for Predicting Web Users Satisfaction with Information Systems Success using Data Mining Techniques. Journal of Software Engineering, 8: 265-277.

[16] 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.

[17] Şehribanoğlu, S., Diler, S. (2016). Veri Madenciliği Süreçleri ve Karar Ağaçları Algoritmaları ile Bir Uygulama (yüksek lisans tezi). Van Yüzüncü Yıl Üniversitesi, Fen Bilimleri Enstitüsü, Van.

[18] Sánchez-Maroño, N., Alonso-Betanzos, A., Fontenla-Romero, O., Polhill, J. G., Craig, T. (2017). Empirically Derived Behavioral Rules in Agent-Based Models Using Decision Trees Learned from Questionnaire Data.

[19] Bajdora, P., Pawełoszek, I. (2020). "Data Mining Approach in Evaluation of Sustainable Entrepreneurship", Procedia Computer Science, 176, pp.2725–2735.

[20] Koçak, H. (2020). Çalışanların Örgütsel Bağlılıklarının Cart Karar Ağacı Algoritması ile Belirlenmesi. Uluslararası İktisadi ve İdari Bilimler Dergisi.

[21] Beernaert, B. (2021). "Using Machine Learning Techniques for Analyzing Survey Data", Ghent University, Faculty of Science, Master Thesis.

[22] Aile, Çalışma ve Sosyal Hizmetler Bakanlığı, Aile ve Toplum Hizmetleri Genel Müdürlüğü Türkiye Aile Yapısı İleri İstatistik Analizi, 2018.

[23] Özkan, Y. (2008). Veri Madenciliği Yöntemleri, Papatya Yayınevi.

[24] Dangare, C. S., & Apte, S. S., 2012, Improved study of heart disease prediction, system using data mining classification techniques, International Journal of Computer Applications, 47(10), 44-48

[25] Goyal, A., & Mehta, R., 2012, Performance comparison of Naïve Bayes and J48 classification algorithms, International Journal of Applied Engineering Research, 7(11), 2012.

[26] Marmanis, H., & Babenko, D., 2009, Algorithms of the intelligent web, Manning, Greenwich, ISBN 978-1-933988-66-5.

[27] Liu, C., Berry, P. M., Dawson, T. P., & Pearson, R. G., 2005, Selecting thresholds of occurrence in the prediction of species distributions, Ecography, 28(3), 385-393

[28] Azari, A., Janeja, V. P., & Mohseni, A., 2012, Predicting hospital length of stay (PHLOS): A multi-tiered data mining approach, In Data Mining Workshops (ICDMW), pp. 17-24

[29] Dirican, A. (2001). Tanı testi performansının değerlendirilmesi ve kıyaslanması. Cerrahpaşa Tıp Dergisi. 32, 25-30

[30] Silahtaroğlu, G. (2013). Veri Madenciliği Kavram ve Algoritmaları, Papatya Yayınevi.

[31] Aher, S. B. (2011). Data Mining in Educational System using WEKA. International Conference on Emerging Technology Trends (ICETT), Proceedings published by International Journal of Computer Applications.

[32] El-Halees, A. M. (2009). Mining Students Data to Analyze e-Learning Behavior: A Case Study.

[33] Özdemir, O., Kaya, A. (2018). Bulanık Kümeleme Analizinde Bulanık Kümeleme Algoritmalarının Karşılaştırılması. (Yüksek Lisans Tezi) Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir.

[34] Timor, M., Şimşek, U. T. (2008). Veri madenciliği sepet analizi ile tüketici davranışı modellemesi. İstanbul Üniversitesi İşletme Fakültesi İşletme İktisadı Enstitüsü Dergisi, 59: 3-10.

[35] Sullivan, W. (2017). Machine Learning for Beginners Guide Algorithms: Supervised & Unsupervsied Learning. Decision Tree & Random Forest Introduction. CreateSpace Independent Publishing Platform.

[36] Mitra, S., Acharya T. (2003). Data Mining Multimedia, Soft Computing and Bioinformatics, Wiley Publication.

[37] Murthy, K. S. (1998). Automatic Construction of Decision Tree from Data: A Multi-Disciplinary Survey. Kluver Academic Publishers. Siemens Corporate Research.

[38] Myatt, J. G. (2007). Making Sense of Data**.** Wiley Publication.

[39] Öğüdücü, G. Ş. (2008). Veri madenciliği temel sınıflandırma yöntemleri. İstanbul Teknik Üniversitesi, İstanbul. Erişim Tarihi: 14.12.2015.

[40] Damanik, I. S., Windarto, A. P., Wanto, A., Andani, S. R., Saputra, W. (2019). Decision Tree Optimization in C4. 5 Algorithm Using Genetic Algorithm. Journal of Physics: Conference Series, vol. 1255, no. 1, p. 012012.

[41] Gupta, G. (2014). A self-explanatory review of decision tree classifiers. International conference on recent advances and innovations in engineering (ICRAIE-2014), pp. 1–7.

[42] Gavankar, S. S., Sawarkar, S. D. (2017). Eager decision tree. 2nd International Conference for Convergence in Technology (I2CT), Mumbai, pp. 837–840.

[43] Quinlan, J.R. (1987). Simplifying decision tree. International Journal of Machine Studies, vol.27, pp.221-234.

[44] Breiman, L., Friedman J., Olshen R., Stone C. (1984). Classification and Regression Trees, Wadsworth Int. Group.

[45] Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Los Altos, Morgan Kaufmann Series in Machine Learning, (1st edition).

[46] Quinlan, J. R. (1986). Introduction of decison trees. Machine Learning, vol.1, pp.81-106.

[47] Rokach, L., Maimon, O. (2005). Decision Trees. Data Mining and Knowledge Discovery Handbook, Ed. by Oded Maimon, Lior Rokach, Springer Science+Business Media Inc., pp.165-192.

[48] Olaru, C., Wehenkel L. (2003). A complete fuzzy decision tree technique, Fuzzy Sets and Systems, vol.138(2), pp.221-254.

[49] Cichosz, P. (2015). Data Mining Algorithms Explained Using R. John Wiley & Sons.

[50] Kotsiantis, S. B. (2007). Supervised Machine Learning: A Review of Classification Techniques, Infomatica 31, 249 – 268.

[51] Pandya, R., Pandya, J. (2015). C5.0 Algorithm to Improve Decision Tree with Feature Selection and Reduced Error Pruning. Int'l Journal of Computer Applications, Vol. 117, No 16, pp. 18 – 21.

[52] Le, Q. V., Ranzato, M. A., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., Ng, A.Y. (2012). Building High – Level Features Using Large Scale Unsupervised Learning. In Proceedings of the 29th Int'l Conference on Machine Learning, Edinbargh, Scotland, UK.

[53] Xiaohu, W. Lele, W., Nianfang, L. (2012). An Application of Decision Tree Based on ID3. In Proceedings of 2012 Int'l Conference on Solid State Devices and Material Science. Physics Procedia, Vol. 25, pp. 1017-1021.

[54] Ben-Gal, I., Trister, C. (2014). Parallel Construction od Decision Trees with Consistently Non-Increasing Expected Number of Tests. Applied Stochastic Models in Business and Industry.

[55] Lu, Z., Wu, X., Bongard, J. C. (2015). Active learning through adaptive heterogeneous ensembling. IEEE Transactions on Knowledge and Data Engineering 27 (2):368–81.

[56] Lim, T. S., Loh, W. Y., Shih, Y. S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning 40 (3):203–28.

[57] Hssina, B., Merbouha, A., Ezzikouri, H., Erritali, M. (2014). A comparative study of decision tree ID3 and C4. 5. International Journal of Advanced Computer Science and Applications 4 (2).

[58] Garca Laencina, P. J., Abreu, P. H., Abreu, M. H., Afonoso, N. (2015). Missing data imputation on the 5-year survival prediction of breast cancer patients with unknown discrete values. Computers in Biology and Medicine 59:125–33.

[59] Behera, H. S., Mohapatra, D. P. (2015). Computational Intelligence in Data Mining Volume 1: Proceedings of the International Conference on CIDM. 5-6 December 2015 Vol. 410. Springer.

[60] Steinberg, C. (2009). CART: Classification and Regression Trees. The Top Ten Algorithms in Data Mining, Ed. by, Xindong Wu, Vipin Kumar, Boca Raton FL, Chapman & Hall/CRC Taylor and Francis Group LLC, p.181.

[61] Köktürk, F. (2012). K-En Yakın Komşuluk, Yapay Sinir Ağları ve Karar Ağaçları Yöntemlerinin Sınıflandırma Başarılarının Karşılaştırılması (doktora tezi). BEÜ, Sağlık Bilimleri Enstitüsü, Zonguldak.

[62] Singh, S., Gupta, P. (2014). Comparative Study ID3, CART and C4.5 Decision Tree Algorithm: A survey, International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682 Vol.27, No.27.

[63] Xiaowei, L. (2014). Application of decision tree classification method based on information entropy to web marketing. In 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation, 121–127, IEEE.

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

[65] De Ville, B. (2006). Decision Trees for Business Intelligence and Data Mining: Using SAS Enterprise Miner. Cary, NC: SAS Institute Inc.

[66] Mistikoğlu, G., Gerek, I. H., Erdis, E., Usmen, P. E. M., Cakan, H., Kazan, E. E. (2015). Decision Tree Analysis of Construction Fall Accidents Involving Roofers. Expert Systems with Applications42 (4): 2256–2263.

[67] Kohavi, R. (1996). Scaling up the accuracy of naïve bayes classifiers: A decision-tree hybrid," ser. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. AAAI Press, pp. 202–207.

[68] Nor, W. M. H., Salleh, M., Omar, A. H. (2013). A Comparative Study of Reduced Error Pruning Method in Decision Tree Algorithms. IEEE International Conference on Control System, Computing and Engineering (ICCSCE). IEEE; p. 392–397.

[69] Pfahringer, B. (2011). Random model trees: an effective and scalable regression method. University of Waikato, New Zealand.

ÖZGEÇMİŞ

ORCID NO: 0000-0001-9505-7891

Ad Soyad : Ferdi KARAKÜTÜK

Yabancı Dil : İngilizce

Eğitim ve Mesleki Geçmişi:

- 2014, Anadolu Üniversitesi, Fen Fakültesi, İstatistik Bölümü
- 2020, İstatistikçi, Türkiye İstatistik Kurumu, Sosyal Sektör İstatistikleri Takımı

Yayınları ve/veya Bilimsel/Sanatsal Faaliyetleri:

- Kaya, A., Özdemir, Ö., Karakütük, F. (2021). Investigation of Life Satisfaction Data by Data Mining and Machine Learning Techniques, International Journal of Culture Heritage, Volume 6 (1), 63-68.