T.C.
MARMARA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
EKONOMETRİ ANABİLİM DALI
İSTATİSTİK BİLİM DALI

DİJİTAL REKLAM VERİLERİNDEN YARARLANARAK POTANSİYEL KONUT ALICILARININ RASTGELE ORMAN YÖNTEMİYLE SINIFLANDIRILMASI

YÜKSEK LİSANS TEZİ

HAYDAR EKELİK

İSTANBUL,2019

T.C.
MARMARA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
EKONOMETRİ ANABİLİM DALI
İSTATİSTİK BİLİM DALI

DİJİTAL REKLAM VERİLERİNDEN YARARLANARAK POTANSİYEL KONUT ALICILARININ RASTGELE ORMAN YÖNTEMİYLE SINIFLANDIRILMASI

YÜKSEK LİSANS TEZİ

HAYDAR EKELİK

Danışman: PROF.DR. DİLEK ALTAŞ

İSTANBUL,2019

T.C.
MARMARA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜ

TEZ ONAY BELGESİ

EKONOMETRİ Anabilim Dalı İSTATİSTİK Bilim Dalı TEZLİ YÜKSEK LİSANS öğrencisi HAYDAR EKELİK'nın DİJİTAL REKLAM VERİLERİNDEN YARARLANARAK POTANSİYEL KONUT ALICILARININ RASTGELE ORMAN YÖNTEMİYLE SINIFLANDIRILMASI adlı tez çalışması, Enstitümüz Yönetim Kurulunun 6.12.2018 tarih ve 2018-34 sayılı kararıyla oluşturulan jüri tarafından oy birliği / oy çokluğu ile Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Savunma Tarihi 03/01/2019

Öğretim Üyesi Adı Soyadı			**İmzası**
1.	Tez Danışmanı	Prof. Dr. DİLEK ALTAŞ	
2.	Jüri Üyesi	Prof. Dr. AHMET METE ÇİLİNGİRTÜRK	
3.	Jüri Üyesi	Prof. Dr. HAKAN SATMAN	

ÖZ

Günümüzde internet ağlarının yaygınlaşması ve internete erişimin bir ihtiyaç haline gelmesi internet sitelerinde ve diğer dijital platformlardaki reklamların kullanılmasını yaygınlaştırmıştır. Dijital reklamcılık olarak adlandırılan bu süreç firmalar, markalar ve diğer kuruluşlar için insanlara ulaşma ve reklam amaçları doğrultusunda hedeflerini gerçekleştirmelerinde vazgeçilmez bir reklam aracı olmuştur. En önemli özelliği ölçülebilir olan dijital reklamcılık, firmalara çok geniş veriler(istatistikler) vermektedir. Firmalar bu verileri kullanıp dijital reklamların değerlendirmesini yaparak gelecek reklam planları için ön görüye sahip olurlar. Dijital reklam yayınlarından elde edilen veriler çeşitli istatistiksel yöntemlerle analiz elde edilebilir. Bu tezin amacı bir inşaat firmasının dijital reklam kampanyasından elde edilen kullanıcı verilerini kullanarak bir sınıflandırma yapmaktır. Kullanıcıların satış ofisine gelip gelmediklerinin kaydının tutulduğu veriler analiz edilerek bir sınıflandırıcı oluşturulmuştur. Bundan sonraki reklamlarla elde edilen kullanıcı verileri bu sınıflandırıcı kullanılarak sınıflandırılabilir. Böylece kullanıcıların satış ofisine gelip gelmemeleriyle ilgili bir ön bilgi elde edilir. Firma bu ön bilgi sayesinde satış ve pazarlama doğrultusundaki hedeflerini daha doğru bir şekilde belirleyebilir.

Tezin amacı doğrultusunda bağımlı değişken olarak kullanıcıların satış ofisine gelip gelmemesi, bağımsız değişken olarak ise dijital reklamlar sayesinde kullanıcın iletişim bilgilerini ***hangi gün*** *firma çalışanlarına gönderdiği, kullanıcının cinsiyeti, reklamı* ***hangi sitede*** *görüp siteye geldiği, reklamı* ***hangi reklam alanında*** *(doğal, 300*250 görsel boyutlu vb.) gördüğü,* ***hangi cihazdan*** *(bilgisayar veya telefon) gördüğü, kullanıcının daha önce ilgili firmada kayıtlı olup olmaması ve bu formu hangi amaçla doldurduğu (yatırım, ev sahibi olma vb.) olmak üzere toplamda 7 adet bağımsız değişken kullanılmıştır.*

Tez kapsamında, karar ağaçları algoritmaları ile birlikte topluluk öğrenme algoritmaları Rastgele Orman (Random Forset), Hızlandırma (Boosting) ve Torbalama (Bagging) algoritmaları tanıtılmış olup aralarındaki farklardan söz edilmiştir.

Uygulamada R programı kullanılmış ve veriyi analiz etmek için bir topluluk öğrenme algoritması olan Rastgele Ormanlar Yöntemi kullanılmıştır. Temelinde karar ağaçları olan bu yöntem diğer sınıflandırma algoritmalarına göre daha iyi sonuçlar vermiştir.

ABSTRACT

Classification of Potential Residential Buyers by Using Random Forest Method taking advantage Digital Advertising Data

The widespread use of internet networks and the need to access the internet has become widespread in the use of internet sites and other digital platforms. This process, which is called digital advertising, has become an indispensable advertising tool for companies, brands and other organizations to reach their goals and to realize their goals in accordance with advertising purposes. Digital advertising, the most important feature of which is measurable, gives companies very large data (statistics). Digital advertising, the most important feature of which is measurable, gives companies very large data (statistics). Firms use this data to evaluate the digital advertising and have a look to the future advertising plans. Data obtained from digital advertising publications can be analyzed by various statistical methods. The purpose of this thesis is to make a classification by using user data from a construction company's digital advertising campaign. A classifier is created by analyzing the data that the users are kept in the sales office and a classifier is created and the user data obtained with the subsequent ads can be classified using this classifier. Thus, a preliminary information can be obtained about whether the users come to the sales office. Through this preliminary information, the company will determine its targets in sales and marketing more accurately.

*For the purpose of the thesis, as a dependent variable, whether the users come to the sales office, as the independent variable, thanks to digital ads, the user sends which days the contact information to the employees of the company, the gender of the user, the site in which the advertisement is seen and the site in which it is advertised, in which advertisement area (native, 300*250 visual size etc.). a total of 7 independent variables were used, which were seen by the user (computer or telephone), whether the user had previously been registered in the relevant company and for which purpose he filled in this form (investment, host, etc.)*

Within the scope of the thesis, Random Forset, Boosting and Bagging algorithms have been introduced together with decision trees algorithms and differences between them have been mentioned

In practice the R program was used and the Random Forests Method, a community learning algorithm, was used to analyze the data. This method, which is based on decision trees, yields better results than other classification algorithms.

TEŞEKKÜR

Tez çalışmam boyunca beni yönlendiren ve destek veren değerli danışman hocam Prof. Dr. Dilek Altaş'a, eleştiri, fikir ve yardımlarıyla çalışmamda katkı sağlayan değerli hocalarım Prof. Dr. Mustafa Tekin, Prof. Dr. Mehmet Hakan Satman, Dr. Öğr. Üyesi Seda Karakaş Geyik ve Dr. Öğr. Üyesi Şenol Emir'e çok teşekkür ederim.

Ayrıca, verilerin elde edilmesi sürecinde yardımlarını esirgemeyen ve gerekli iletişimi sağlayan Güçlü Bestan ve Pelin Acar'a ve verilerin hazırlanmasında büyük emeği olan Osman Cankut Yağcılar ve Yağmur Yalçın'a teşekkürlerimi sunarım**.**

Haydar Ekelik

İstanbul, 2019

İÇİNDEKİLER

ŞEKİL LİSTESİ

..

TABLO LİSTESİ

GİRİŞ

Günümüzde internet ağlarının yaygınlaşması ve internete erişimin bir ihtiyaç haline gelmesi internet sitelerinde ve diğer dijital platformlardaki reklamların kullanılmasını yaygınlaştırmıştır. Dijital reklamcılık olarak adlandırılan bu süreç, internet teknolojilerinin geldiği son noktada, insanların bilgiye en hızlı ulaştığı, ilk aramayı yaptığı internete bağlı cihazlar üzerinde, markaların ya da ürünlerin tanıtımının yapılmasıdır. Markalar ya da ürünler bu teknolojiyi kullanarak reklam kampanyalarında daha ulaşılabilir ve daha görünür olmaktadır. Günümüzde şirketler, web sitelerini ziyaret ettikten sonra tüketicilerin davranışlarındaki değişikliklere bağlı olarak yapılan araştırmaların sonucunda web sitelerine trafik sağlamak için çeşitli reklam türleri kullanarak dijital reklamcılık yatırımlarını genişletmektedir. Çoğu şirket, günümüzde çevrimdışı reklamlara (televizyon, gazete, radyo vb.) ek olarak farklı türde çevrimiçi reklamlara yatırım yapmaktadır. Şirketler, müşterilerine dijital reklamlar sayesinde kişisel veya belirli bir hedef kitleye özgü reklamları gösterebilirler.

Dijital pazarlama, geleneksel pazarlamayla aynı amaca ulaşmak için dijital kanalları kullanan geleneksel pazarlamanın alt dalı olarak tanımlanabilir. Dijital pazarlama, müşterilere sosyal medya, arama motorları (google vb.) bloglar, forumlar, e-posta pazarlama, mobil uygulamalar ve çevrimiçi görüntülü pazarlama kanallarının bir araya getirilmesiyle ulaşır.

Bu çalışmada bir inşaat firmasının, belirli bir dönemde yapmış olduğu dijital reklam yayınlarından elde edilen veriler (kullanıcı verileri) analiz edilmiştir. İnşaat firmasının reklamlarını dijital platformlarda görmüş olan kullanıcılar bu reklamlar sayesinde firma sitesine gelmiş ve burada konut alımıyla ilgili olarak iletişim bilgilerini firmaya bir form aracılığıyla göndermişlerdir. Bu kullanıcıların (müşteriler) inşaat firması satış ofisine gelip gelmedikleri kayıt edilmiştir. Elde edilen bu bilgiler sayesinde kullanıcılar satış ofisine gelip gelmeme durumuna bağlı olarak veri madenciliğinde çokça kullanılan ve karar ağaçları için bir topluluk öğrenme algoritması olan Rastgele Orman (Random Forest) yöntemiyle sınıflandırılmıştır. Böylelikle elde edilen yeni bir kullanıcı bilgisi, geliştirilen model sayesinde bu kullanıcının satış ofisine gelip

gelmemesi tahmin edilecektir. Böylelikle firma elindeki müşteri portföyünü daha etkin bir şekilde kullanacaktır.

Bu amaç doğrultusunda tezin Birinci bölümünde veri madenciliği tanımı, amaçları ve kullanım alanları hakkında bilgi verilmiştir.

İkinci bölümde, karar ağaçları hakkında genel bilgilere, karar ağaçlarında kullanılan yeniden örnekleme teknikleri, model değerlendirme yaklaşımları, bölünme ve budanma yöntemlerine yer verilmiştir.

Üçüncü bölümde, popüler karar ağacı algoritmaları hakkında bilgiler verilmiştir.

Dördüncü bölümde, karar ormanları algoritmaları (topluluk öğrenme algoritmaları) olan Torbalama (Bagging), Hızlandırma (Boosting) ve Rastgele Orman (Random Forest) algoritmaları hakkında bilgi verilmiştir.

Beşinci bölümde, dijital reklam yayınları sonucunda elde edilen verilerle Rastgele Orman algoritmasıyla yapılan uygulama sonuçları yer almaktadır.

Sonuç bölümünde ise, uygulama sonucunda elde edilen bulgulara dayanarak çalışmanın genel bir değerlendirilmesi yapılmış ve çalışmanın gerçek hayatta nasıl kullanılacağı hakkında bilgilere yer verilmiştir.

1.Bölüm

Veri Madenciliği

1960'lardan beri, veri tabanı ve bilgi teknolojisi, ilkel dosya işleme sistemlerinden çok gelişmiş ve güçlü veri tabanlarına kadar sistematik olarak gelişmektedir. 1970'li yıllardan beri veri tabanı sistemlerinde araştırma ve geliştirme, ilişkisel veri tabanı sistemlerinin, veri modelleme araçlarının, veri indekslemenin ve veri organizasyon tekniklerinin geliştirilmesine yol açmıştır. Kullanıcılar sorgulama dilleri, sorgulama işlemleri ve kullanıcı arabirimleri aracılığıyla uygun ve ulaşılabilir veri erişimi elde etmektedir. Sorgunun salt okunur bir işlem olarak görüldüğü çevrimiçi hareket işleme (on-line transaction processing) (OLTP) için etkin yöntemler, büyük veri setlerini, büyük ölçüde etkin depolama, alma ve yönetimi için ilişkisel teknolojinin önemli bir araç olarak gelişimine ve geniş çapta kabulüne katkıda bulunmuştur. 1980'lerin ortalarından beri veri tabanı teknolojisi, ilişkisel teknolojinin popüler olarak benimsenmesi, yeni ve güçlü veri tabanı sistemleri üzerinde araştırma ve geliştirme faaliyetlerinin yükselmesi ile karakterize edilmiştir.[1]

Günümüzde veri birçok farklı veri tabanında saklanabilir haldedir. Yakın zamanda ortaya çıkan bir veri tabanı mimarisi, yönetim kararlarını vermeyi kolaylaştırmak amacıyla tek bir alanda tek bir şema altında organize edilen çoklu heterojen veri kaynaklarının bulunduğu veri ambarıdır. Veri ambarı teknolojisi, veri temizleme, veri entegrasyonu ve On-Line Analytical Processing (OLAP), yani özetleme, konsolidasyon[2] ve toplanma gibi işlevselliklerin yanı sıra farklı açılardan bilgi görüntüleme yeteneğine sahip analiz teknikleri içerir.[3]

Veri madenciliğinin son yıllarda bilgi endüstrisine büyük ilgi göstermesinin başlıca sebebi, büyük miktarlardaki verinin geniş çapta kullanılabilir olması ve bu tür bilgilerin yararlı bilgi ve bilgiye dönüştürülmesi için gerekli olan ihtiyaçtan kaynaklanmaktadır. Kazanılan bilgi, işletme yönetimi, üretim kontrolü ve pazar

[1] Jiawei Han ve Micheline Kamber, Data Mining:Concepts and Techniques ,Simon Fraser University: Morgan Kaufman Publıshers, 2000,s.3

[2] Verileri, farklı noktalarda konumlandırılmış yerel veri tabanlarından merkezi veri tabana toplama işlemine verilen

[3] Han ve Kamber, s.5

analizinden mühendislik tasarımına ve bilim araştırmasına kadar çeşitli uygulamalar için kullanılabilir.[4]

1.1.Veri Madenciliği Tanımı ve Veri Tabanlarında Bilgi Keşfi (KDD)

Veri tabanı sistemlerinin artan kullanımı ve hacimlerindeki bu olağanüstü artış, kuruluşları elde toplanan bu verilerden nasıl faydalanılabileceği problemi ile karşı karşıya bırakmıştır. Geleneksel sorgu (query) veya raporlama araçlarının veri yığınları karşısında yetersiz kalması, veri tabanlarında bilgi keşfi (VTBK), (Knowledge Discovery in Databases) adı altında, sürekli ve yeni arayışlara neden olmaktadır. VTBK, süreci içerisinde, modelin kurulması ve değerlendirilmesi aşamalarından meydana gelen veri madenciliği (data mining) en önemli kesimi oluşturmaktadır. İnsanların hızla büyüyen sayısal veri hacimlerinden faydalı bilgi elde etmelerine yardımcı olan yeni nesil hesaplama teorilerine ve araçlarına ihtiyaç duyulmuştur. Bu teoriler ve araçlar, veri tabanlarında ortaya çıkan bilgi keşfinin (KDD) konusudur[5]. Konunun önde gelen uzmanlarından Piatetsky-Shapiro veri madenciliğini, verilerden daha önceden bilinmeyen, zımnî[6], muhtemelen faydalı enformasyonun monoton olmayan bir süreçte çıkartılması işlemi olarak tanımlamaktadır.[7]

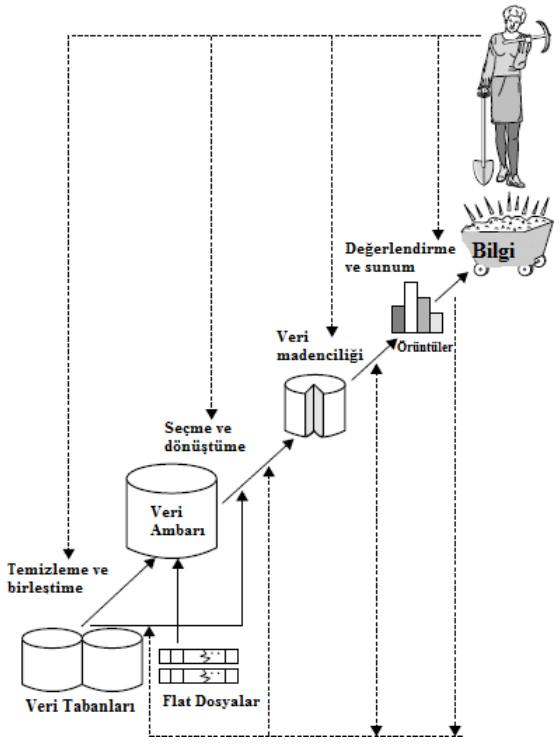
Bir başka görüşe göre; veri madenciliği, veri tabanlarında, veri ambarlarında veya diğer bilgi depolarında saklanan büyük miktardaki verilerden ilginç bilgileri keşfetme sürecidir.[8]

Özetle veri madenciliği, büyük miktarda veriden elde edilen bilgiyi çıkarma (değerli olan bilgiyi elde etme) olarak ifade edilir. Terim aslında yanlış ifade edilmektedir. Kayalardan veya kumdan altın madenciliğine atıfta bulunmak için, kaya veya kum madenciliği yerine altın madenciliği denilmektedir. Benzer şekilde, veri madenciliği, maalesef biraz uzun olan "veriden bilgi madenciliği" olarak adlandırılmış olmalıdır. Bununla birlikte, daha kısa vadeli olan bilgi madenciliği, büyük miktarda

[4] Han ve Kamber, s.3
[5] Usama Fayyad, Gregory Piatetsky-shapiro, ve Padhraic Smyth, From Data Mining to Knowledge Discovery in Databases, AI Magazine Volume 17 Number 3,1996, s.37–54.
[6] Kapalı bir biçimde söylenen ya da anlatılan, sezdirilen, kapalı.
[7] Haldun Akpınar, 'Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği, İ.Ü. İşletme Fakültesi Dergisi,C:29, S: 1/Nisan 2000, s.1-22.
[8] Han ve Kamber, s.6

veriden madenciliğe verilen önemi yansıtmayabilir. Madencilik, büyük miktarda ham maddeden oluşan küçük bir değerli altın (külçe) setini tanımlayan süreci karakterize eden canlı bir terimdir. Böylece hem "veri" hem de "madenciliği" taşıyan böyle yanlış bir isim popüler hale gelmiştir. Buna ek olarak, veri madenciliğine benzer veya biraz farklı bir anlam taşıyan, veri tabanlarından bilgi madenciliği, bilgi çıkarma, veri / örüntü analizi, veri arkeolojisi ve veri taraması gibi başka birçok terim vardır.[9]

Birçok kişi veri madenciliğini, popüler olarak kullanılan başka bir terim olan "Veri tabanlarında Bilgi keşfi" (KDD) ile eşanlamlı olarak ele almaktadır. Alternatif olarak, diğerleri veri madenciliğini, veri tabanlarındaki bilgi keşfi sürecinde sadece temel bir adım olarak görür. Bir süreç olarak bilgi keşfi, aşağıdaki şekilde betimlenmiştir ve aşağıdaki adımların yinelemeli bir dizisinden oluşmaktadır.[10]



Şekil 1.1. Bilgi keşfi sürecinde bir adım olarak veri madenciliği.[11]

- Veri temizleme; gürültü ve tutarsız verileri kaldırdığı adımdır.

[9] Han ve Kamber, s.6
[10] Han ve Kamber, s.6
[11] Jiawei Han , Micheline Kamber ve Jian Pei, Data Mining Concepts and Techniques, Third Edit, Waltham, Morgan Kaufman Publıshers,2012, s.7

- Veri entegrasyonu (birleştirme); birden fazla veri kaynağının birleştirilebileceği adımdır.

- Veri seçimi; Analiz ile ilgili verilerin veri tabanından alındığı yer.

- Veri dönüşümü; örneğin, özet veya toplama işlemlerini gerçekleştirerek verilerin analiz için uygun biçimlere dönüştürülmesi veya birleştirilmesi adımıdır. Veri birleştirme olarak da bilinen bu adım, seçilen verilerin madencilik prosedürüne uygun formlara dönüştürüldüğü bir aşamadır.[12]

- Veri madenciliği; veri örüntülerini çıkarmak için algoritmaların uygulandığı önemli bir adımdır.

- Örüntü değerlendirmesi; bazı ilginçlik ölçümlere dayanan bilgiyi temsil eden ilginç kalıpları tanımlama adımıdır.

- Bilgi sunumu; madencilik bilgisini kullanıcıya sunmak için görselleştirme ve bilgi temsili tekniklerinin kullanıldığı adımdır[13].

İlginç örüntüler kullanıcıya sunulur ve bilgi tabanında yeni bilgi olarak saklanabilir. Bu görüşe göre, veri madenciliğinde, değerlendirme için gizli örüntüler açığa çıkarıldığı için, önemli olsa da, tüm süreçte sadece bir adımdır.[14]

Bu adımlardan bazılarını birleştirmek yaygındır. Örneğin, veri temizleme ve veri entegrasyonu bir veri ambarı oluşturmak için ön-işleme aşaması olarak birlikte gerçekleştirilebilir. Veri seçimi ve veri dönüşümü, verilerin birleştirilmesinin sonucu olduğu veya veri ambarlarında veri seçiminin dönüştürülmüş veriler üzerinde yapıldığı durumlarda da birleştirilebilir.[15]

[12] Osmar R. Zaïane, Principles of Knowledge Discovery in Databases.(webdocs.cs.ualberta.ca/~zaiane/courses/cmput690/notes/Chapter1/ch1.pdf, 25Ağustos 2018'de erişildi)
[13] Han ve Kamber, s.6
[14] Han ve Kamber, s.6
[15] Zaïane, s.4

1.2. Veri Madenciliğinin Uygulandığı Veri Tabanları

Veri madenciliği her türlü bilgi havuzuna uygulanabilir. Bununla birlikte, farklı veri tiplerine uygulandığında algoritmalar ve yaklaşımlar farklılık gösterebilir.[16]

1.2.1. İlişkisel Veri Tabanları (Relational Databases)

Veri tabanı yönetim sistemi (DBMS) olarak da adlandırılan bir veri tabanı sistemi, veri tabanı olarak bilinen birbiriyle ilişkili veri topluluğundan veriyi yönetmek ve bunlara erişmek için bir dizi yazılım programından oluşur.[17]

İlişkisel veri tabanı, her birine benzersiz bir ad atanan bir tablo koleksiyonudur. Her tablo bir dizi öznitelikten (sütun veya alan) oluşur ve genellikle çok sayıda tüple (veri grubu) (kayıtlar veya satırlar) depolanır. İlişkisel bir tabloda bulunan her bir tüple (veri grubu), benzersiz bir anahtar tarafından tanımlanan ve özellik değerleri kümesi tarafından belirtilen bir nesneyi temsil eder.

İlişkisel verilere SQL gibi ilişkisel bir sorgu dilinde yazılmış veri tabanı sorguları veya grafik kullanıcı ara yüzleri yardımıyla erişilebilir. Sonra kullanıcı, örneğin, sorguya dahil edilecek nitelikleri ve bu özellikler üzerindeki kısıtlamaları belirtmek için bir menü kullanabilir. Belirli bir sorgu, birleştirme, seçim ve projeksiyon gibi bir dizi ilişkisel işleme dönüştürülür ve daha sonra verimli işlem için optimize edilir. Bir sorguda verilerin belirtilen alt kümelerinin alınmasına izin verir.

Veri madenciliği ilişkisel veri tabanlarına uygulandığında, trendler veya veri kalıpları aranarak daha da ileri gidebilir. Örneğin, veri madenciliği sistemleri müşteri verilerini, yeni müşterilerin kredi riskini, gelirlerine, yaşlarına ve önceki kredi bilgilerine göre tahmin etmek için analiz edebilir. İlişkisel veri tabanları, veri madenciliği için en popüler ve en zengin bilgi havuzlarından biridir ve bu nedenle veri madenciliği çalışmalarında önemli bir veri formudur.[18]

[16] Zaïane, s.5
[17] Han ve Kamber, s.9
[18] Han ve Kamber, s.9

1.2.2. Veri Ambarı (Data Warehouses)

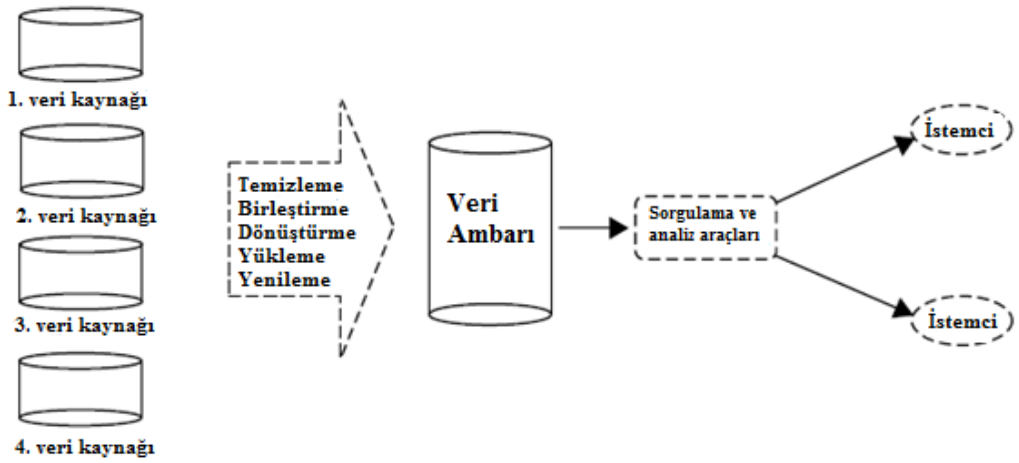
Bir veri ambarı, genellikle, her bir boyutun şemadaki bir öznitelik veya bir dizi özdeşliğe karşılık geldiği veri küpü adı verilen çok boyutlu bir veri yapısı tarafından modellenir ve her bir hücre, sayım veya toplam gibi birtakım ölçümün değerini depolar. Basitçe veri ambarı bir işletmenin sahip olduğu verinin, eskileri de dâhil olmak üzere karar destek amacıyla kullanılmasına olanak sağlar.[19]

Veri ambarı, birbiriyle bütünleşik olmayan uygulamaların bütünleştirilmese olanak sağlar.[20] Bir veri küpü çok boyutlu bir veri görünümü sağlar ve özetlenmiş verilerin önceden hesaplanmasını ve hızlı bir şekilde erişilmesini sağlar. Veri ambarı, KDD için iki önemli aşamayı ayarlamaya yardımcı olur: (1) veri temizleme ve (2) veri erişimi.[21]

Veri temizliği: Kuruluşlar sahip oldukları çok çeşitli veri ve veri tabanlarının birleşik ve mantıksal görünümü hakkında düşündükçe, verileri tek bir adlandırma kuralıyla eşleştirmek, eksik verileri tekdüze olarak temsil etmek, gürültü ve hata gibi veri işleme sorunları ele almalıdırlar.

Veri erişimi: Verilere erişmek ve veriye erişim yolları sağlamak için geçmişe dönük ve iyi tanımlanmış yöntemler oluşturulmalıdır. (örneğin, çevrimdışı saklanan veriler).

[19] Han ve Kamber, s.11
[20] Yalçın Özkan, Veri Madenciliği Yöntemleri, 1.Basım, İstanbul: Papatya Yayınları, 2008, s.22
[21] Usama Fayyad, Gregory Piatetsky-shapiro, ve Padhraic Smyth, From Data Mining to Knowledge Discovery in Databases, AI Magazine Volume 17 Number 3,1996, s.37–54.



Şekil 1.2. Veri ambarı için tipik çerçeve[22]

Veri ambarlarının analizi için popüler bir yaklaşım, Codd (1993) tarafından önerilen çevrimiçi analitik işlem (On Line Transactional Processing) (OLAP) olarak adlandırılmaktadır. OLAP araçları, çok boyutlu veri analizi sağlamaya odaklanır; bu, birçok boyutta bilgi işlem özetleri ve dökümlerinde SQL'den daha üstündür. OLAP araçları, etkileşimli veri analizini basitleştirmeye ve desteklemeye yöneliktir, ancak KDD araçlarının amacı, mümkün olduğunca fazla işlemi otomatik hale getirmektir.[23]

1.2.3. İşlemsel Veri Tabanları (Transactional Databases)

Genel olarak, bir işlemsel veri tabanı, her kaydın bir işlemi temsil ettiği dosyadan oluşur. Bir işlem genel olarak benzersiz kimlik numarası (işlem ID) ve işlemi oluşturan öğelerin listesini (bir mağazada satın alınan öğeler gibi) içerir. İşlem veri tabanı, satışla ilgili, işlem tarihi, müşteri kimlik numarası, satış yetkilisinin kimlik numarası ve satışın gerçekleştiği şube gibi diğer bilgileri içeren ek tablolara sahip olabilir.[24]

[22] Han, Kamber ve Pei, s.11
[23] Usama Fayyad, Gregory Piatetsky-shapiro, ve Padhraic Smyth, From Data Mining to Knowledge Discovery in Databases, AI Magazine Volume 17 Number 3,1996, s.37–54.
[24] Han ve Kamber, s.12

1.2.4. Gelişmiş Veri Tabanı Sistemleri ve Gelişmiş Veri Tabanı Uygulamaları (Advanced Database Systems And Advanced Database Applications)

İlişkisel veri tabanı sistemleri iş uygulamalarında yaygın olarak kullanılmaktadır. Veri tabanı teknolojisinin ilerlemesiyle birlikte, yeni veri tabanı uygulamalarının gereksinimlerini karşılamak için çeşitli türlerde gelişmiş veri tabanı sistemleri ortaya çıkmıştır.

Gelişmiş veri tabanı uygulamaları arasında mekansal veri (haritalar gibi), mühendislik tasarım verileri (binaların tasarımı, sistem bileşenleri veya entegre devreler gibi), hiper metin ve multimedya verilerinin (metin, resim, video ve ses verileri dahil) ele alınması, zamanla ilgili veriler (tarihsel kayıtlar veya borsa verileri gibi) ve dünya çapında ağ (www) (internet'in sağladığı geniş, yaygın olarak dağıtılmış bilgi deposu) yer almaktadır. Bu uygulamalar, karmaşık nesne yapıları, değişken uzunluklu kayıtlar, yarı yapılandırılmış veya yapılandırılmamış veriler, metin ve multimedya verileri ile karmaşık yapıları ve dinamik değişimleri olan veri tabanı şemalarını işlemek için verimli veri yapıları ve ölçeklenebilir yöntemleri gerektirir.

Bu ihtiyaçlara cevap olarak, gelişmiş veri tabanı sistemleri ve özel uygulama odaklı veri tabanı sistemleri geliştirilmiştir. Bunlar, nesne yönelimli ve nesne-ilişkisel veri tabanı sistemleri, mekânsal veri tabanı sistemleri, zamansal ve zaman serisi veri tabanı sistemleri, metin ve çoklu ortam veri tabanı sistemleri, heterojen ve eski veri tabanı sistemleri ve Web tabanlı küresel bilgi sistemleridir.[25]

1.3. Veri Madenciliği Modelleri

Veri madenciliğinde kullanılan modeller, tahmin edici (predictive) ve tanımlayıcı (descriptive) olmak üzere iki ana başlık altında incelenmektedir.

[25] Han ve Kamber, s.12.

Tahmin edici modellerde, sonuçları bilinen verilerden hareket edilerek bir model geliştirilmesi ve kurulan bu modelden yararlanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesi amaçlanmaktadır.[26]

Tanımlayıcı madencilik görevleri, veri tabanındaki verilerin genel özelliklerini karakterize etmektedir.[27]

Bazı durumlarda, kullanıcıların verilerindeki hangi tür kalıpların ilginç olabileceği konusunda bir fikri olmayabilir ve bu nedenle paralel olarak birkaç farklı desen çeşidi aramak isteyebilir. Bu nedenle, farklı kullanıcı beklentilerini veya uygulamalarını karşılamak için çok sayıda desen türünü oluşturabilen bir veri madenciliği sistemine sahip olmak önemlidir. Ayrıca, veri madenciliği sistemleri, çeşitli taneciklerde (yani, farklı soyutlama seviyelerinde) kalıpları bulabilmelidir.

Veri madenciliği modellerini gördükleri işlevlere göre,

- Sınıflama (Classificatiori) ve Regresyon (Regression),
- Kümeleme (Clustering),
- Birliktelik Kuralları (Association Rules) ve Ardışık Zamanlı Örüntüler (Sequentiai Patterns),

olmak üzere üç ana başlık altında incelemek mümkündür.[28]

1.3.1. Sınıflama ve Regresyon Modelleri

Mevcut verilerden hareket ederek geleceğin tahmin edilmesinde faydalanılan ve veri madenciliği teknikleri içerisinde en yaygın kullanıma sahip olan sınıflama ve regresyon modelleri arasındaki temel fark, tahmin edilen bağımlı değişkenin kategorik veya süreklilik gösteren bir değere sahip olmasıdır.

[26] Haldun Akpınar, 'Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği, İ.Ü. İşletme Fakültesi Dergisi,C:29, S: 1/Nisan 2000, s.1-22.
[27] Han ve Kamber, s.13.
[28] Haldun Akpınar, 'Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği, İ.Ü. İşletme Fakültesi Dergisi,C:29, S: 1/Nisan 2000, s.1-22.

Makine öğrenimi terminolojisinde denetimli öğrenme olarak da bilinen sınıflandırma, veri koleksiyonundaki nesneleri sınıflamak için verilen sınıf etiketlerini kullanır. Sınıflandırma yaklaşımları normalde, tüm nesnelerin bilinen sınıf etiketleri ile ilişkilendirildiği bir eğitim seti kullanır. Sınıflandırma algoritması eğitim setinden öğrenir ve bir model oluşturur. Model yeni nesneleri sınıflandırmak için kullanılır.[29]

Sınıflama ve regresyon modellerinde kullanılan başlıca teknikler,

- Karar Ağaçları (Decision Trees),
- Yapay Sinir Ağları (Artificial Neural Networks),
- Genetik Algoritmalar (Genetic Algorithms),
- K-En Yakın Komşu (K-Nearest Neighbor),
- Bellek Temelli Akıl Yürütme (Memory Based Reasoning),
- Naive-Bayes,
- Destek vektör makinaları
- Lojistik Regresyondur (Logistic Regression).

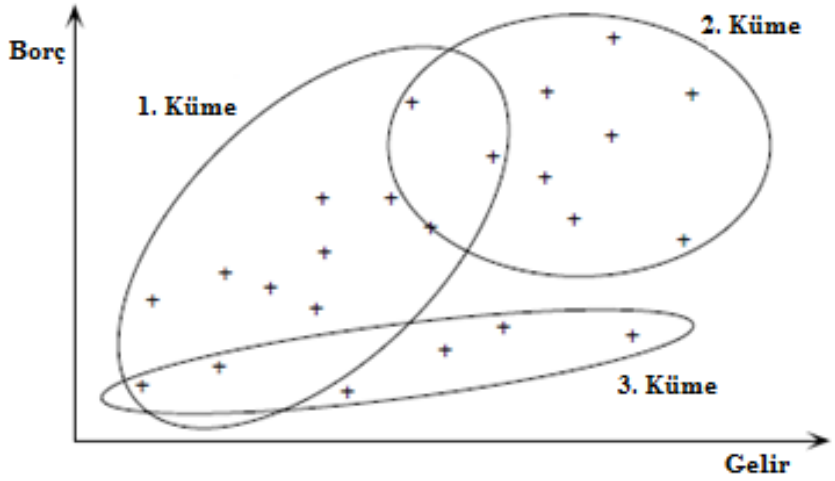
1.3.2. Kümeleme Modelleri

Kümeleme verilerin kendi aralarındaki benzerlikleri göz önüne alınarak gruplandırılması işlemidir.[30] Kümeleme, bilinen bir sınıf etiketine başvurmadan veri nesnelerini analiz eder. Genel olarak, sınıf etiketleri, başlangıçta bilinmedikleri için eğitim verilerinde mevcut değildir.[31] Başlangıç aşamasında veri tabanındaki kayıtların hangi kümelere ayrılacağı veya kümelemenin hangi değişken özelliklerine göre yapılacağı bilinmemekte, konunun uzmanı olan bir kişi tarafından kümelerin neler

[29] Zaïane,s.11
[30] Özkan,s.47
[31] Han ve Kamber,s.16

olacağı tahmin edilmektedir.[32] Nesneler, sınıf içi benzerliği en üst düzeye çıkarmak ve sınıflar arası benzerliği en aza indirgeme ilkesine dayalı olarak kümelenir veya gruplandırılırlar.[33]

Şekil 1.3, kredi verilerinin üç kümeye ayrılmasının olası bir kümelenmesini göstermektedir; Kümelerin üst üste geldiği, veri noktalarının birden fazla kümeye ait olmasına izin verdiğine dikkat ediniz.[34]



Şekil 1.3. Kredi veri setinin 3 kümeye ayrılması.

[32] Haldun Akpınar, 'Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği, İ.Ü. İşletme Fakültesi Dergisi,C:29, S: 1/Nisan 2000, s.1-22.
[33] Han ve Kamber,s.16
[34] Usama Fayyad, Gregory Piatetsky-shapiro, ve Padhraic Smyth, From Data Mining to Knowledge Discovery in Databases, AI Magazine Volume 17 Number 3,1996, s.37–54.

1.3.3. Birliktelik Kuralları ve Ardışık Zamanlı Örüntüler (İlişkilendirme Analizi)

Bir alışveriş sırasında veya birbirini izleyen alışverişlerde müşterinin hangi mal veya hizmetleri satın almaya eğilimli olduğunun belirlenmesi, müşteriye daha fazla ürünün satılmasını sağlama yollarından biridir. Satın alma eğilimlerinin tanımlanmasını sağlayan birliktelik kuralları ve ardışık zamanlı örüntüler, pazarlama amaçlı olarak pazar sepeti analizi (Market Basket Analysis) adı altında veri madenciliğinde yaygın olarak kullanılmaktadır.[35]

İlişkilendirme analizi, belirli bir veri kümesinde sık sık birlikte ortaya çıkan özellik-değer koşullarını gösteren ilişkilendirme kurallarının keşfeder. Analiz, pazar sepeti veya işlem verileri analizi için yaygın olarak kullanılmaktadır.

[35] Haldun Akpınar, 'Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği, İ.Ü. İşletme Fakültesi Dergisi,C:29, S: 1/Nisan 2000, s.1-22.

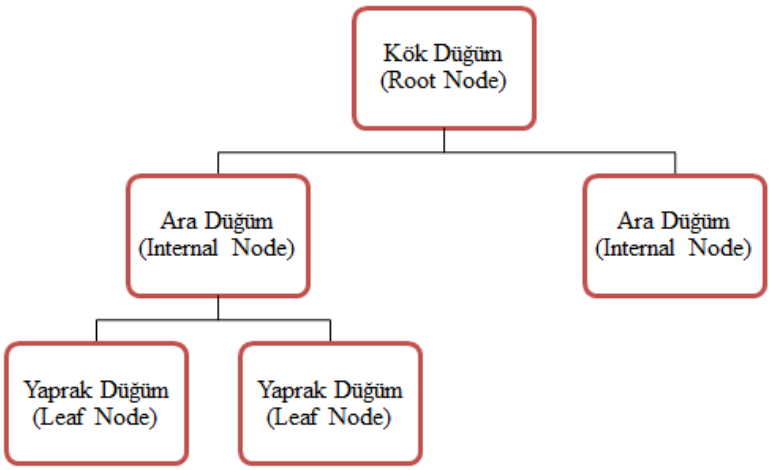
2.BÖLÜM

KARAR AĞAÇLARI

Karar ağaçlar akış şemasına benzeyen yapılardır. Karar ağaçları, istatistiksel olarak anlamlı grupları bulan ve cevapları açık bir şekilde, kolay okunabilir ağaç diyagramları ile veren, gözlemleri sınıflayan ya da tahmin eden kurallar grubudur.[36] Karar ağaçları, sınıflandırma ve regresyon için güçlü ve popüler bir araçtır. Karar ağaçları kuralları temsil eder. Kurallar kolayca ifade edilebilir, böylece insanlar bunları anlayabilir veya doğrudan veri tabanı erişim dilinde kullanılabilir.[37]

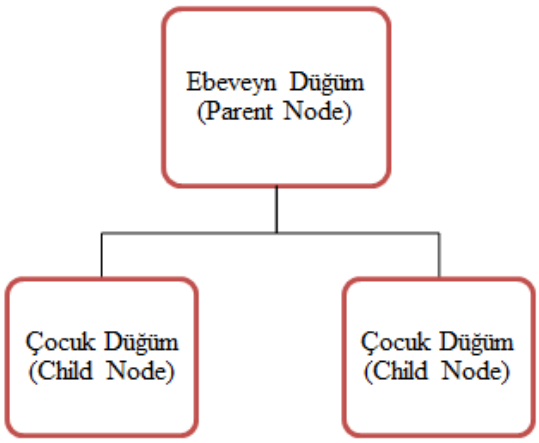
Her bir nitelik bir düğüm tarafından temsil edilir. Dallar ve yapraklar ağaç yapısının elemanlarıdır.[38] Bir karar ağacı, doğal bir ağaçta olduğu gibi kök (*root)*, dal (*branch*) ve yapraklardan (*leaf*) meydana gelmektedir.[39] Bir girdi verildiğinde, her düğümde, bir algoritma uygulanır ve sonuca bağlı olarak dallardan biri alınır. Bu işlem kökte başlar ve bir yaprak düğüme ulaşana kadar ardı ardına tekrarlanır, bu noktada yaprakta yazılan değer, çıkışı oluşturur.[40]

[36] Nurhan Doğan ve Kazım Özdamar, 'CHAID Analizi ve Aile Planlaması Le Lgili Bir Uygulama', T Klin Tıp Bilimleri 2003, 23.1 (2003), 392–398.
[37] Karar Ağacı Metedolojisi,(http://dms.irb.hr/tutorial/tut_dtrees.php, 25 Ağustos 2018'de erişildi.)
[38] Özkan, s.52
[39] Haldun Akpınar, Data: Veri Madenciliği Veri Analizi, Papatya Yayıncılık, 2014, s.204.
[40] Ethem Alpaydin, Introduction to Machine Learning, 2.Edition, London,The MIT Press, 2010,s.186 <https://doi.org/10.1016/j.neuroimage.2010.11.004>.



Şekil 2.1. Örnek bir karar ağacı yapısı

Karar ağacı içerisinde bulunan düğümler, ağaç içerisinde birbirleri ile olan seviyelerine göre ebeveyn (parent) ve çocuk (child) düğümler olarak isimlendirilir.[41] Her aşamada gerçekleştirilen bölme işlemi bir öğrenme süreci olarak ifade edilir. Bu süreç kök düğümden yapraklara doğru gerçekleştirildiği için tümden gelim söz konusudur.(Top-down inducation of decision tree) (TDIDT)



Şekil 2.2. Bir karar ağacında ebeveyn ve çocuk düğümler

[41] Akpınar, Data: Veri Madenciliği Veri Analizi, s.205

Düğümlerin yapıları bakımından karar ağaçları üçe ayrılır.

1) Tek değişkenli karar ağaçları

2) Çok değişkenli karar ağaçları

3) Melez karar ağaçları

Tek değişkenli karar ağaçlarında düğümlerde sorulan sorular ilgili olayın tek bir değişkenine bakılarak oluşturulur ve bu da aslında uzayı dikine bölmektir. Çok değişkenli karar ağaçlarında, düğümlerdeki sorular birçok değişkenin oluşturduğu uzayda ifade edilir. Melez ağaçlarda ise bu iki test biçimi birlikte kullanılır.[42]

Karar ağaçları öğreniminde iki ana hedef söz konusudur. Sınıflandırma ağaçları (classification tree) olarak isimlendirilen birinci grupta, veri dizisinin olabildiğince homojen olarak sınıflandırılması; regresyon ağaçları (regression tree) olarak isimlendirilen ikinci grupta ise, tahmin modellerinin kurulması hedeflenmektedir.[43]

2.1. Yeniden Örnekleme Metotları

Yeniden örnekleme yöntemleri modern istatistikte vazgeçilmez bir araçtır. Bu yöntemler uygun model hakkında ek bilgi elde etmek için, bir eğitim setinden defalarca örnek çekmeyi ve her bir örnekle alakalı bir model oluşturmayı içerirler. Örneğin, doğrusal regresyon modelinin değişkenliğini tahmin etmek için, eğitim verilerinden sürekli olarak farklı örnekler çekilebilir, her yeni örneğe doğrusal bir regresyon uydurulabilir ve daha sonra ortaya çıkan modellerin farklılık derecesi incelenebilir. Böyle bir yaklaşım, orijinal eğitim gözlemleri kullanılarak yalnızca bir kez modelden yararlanılamayacak bilgiler elde edilmesine izin verebilir.[44]

Yeniden örnekleme yaklaşımları hesaplama açısından zahmetlidir, çünkü eğitim verilerinin farklı alt kümelerini kullanarak aynı istatistiksel yöntemi birden çok kez uygulamak gerekir. Ancak, hesaplama gücündeki son gelişmeler, yeniden

[42] Vildan Gülpınar, 'Avrupa Birliği Ülkeleri Ile Türkiye'nin Ekonomik Göstergelerinin Karar Ağacı Yöntemi Ile Karşılaştırılması',İstanbul,(Yüksek Lisans Tezi,2008), s.66..
[43] Akpınar, Data: Veri Madenciliği Veri Analizi, s.205
[44] Gareth James ve diğerleri, An Introduction to Statistical Learning with Applications in R, Performance Evaluation, 2014, s.176, LXİV <https://doi.org/10.1016/j.peva.2007.06.006>.

örnekleme yöntemlerinin hesaplamalarına engel teşkil etmemektedir. Örneğin, çapraz doğrulama, performansını değerlendirmek veya uygun esneklik seviyesini seçmek için belirli bir istatistiksel öğrenme yöntemiyle ilişkili test hatasını tahmin etmek için kullanılabilir. Bir modelin performansını değerlendirme süreci model değerlendirmesi olarak bilinirken, bir model için uygun esneklik seviyesinin seçilmesi süreci model seçimi olarak bilinir.[45]Ayrıca bu yöntemler model değerlendirme yöntemleri olarak da adlandırılmaktadır.

2.1.1. Çapraz Doğrulama (Cross-Validation)

Test hatası, yeni bir gözlemin tahmini değerini elde etmek için istatistiksel bir öğrenme yönteminin ortalama hatasıdır - yani, yöntemin eğitiminde kullanılmayan bir ölçümdür. Bir veri seti verildiğinde, düşük bir test hatasıyla sonuçlanırsa, belirli bir istatistiksel öğrenme yönteminin kullanılması garanti edilebilir. Belirlenen bir test seti mevcutsa test hatası kolayca hesaplanabilir. Ancak genellikle böyle değildir. Aksine, eğitim hatası, eğitiminde kullanılan gözlemlere istatistiksel öğrenme yöntemi uygulanarak kolayca hesaplanabilir. Ancak, eğitim hata oranı genellikle test hata oranından oldukça farklıdır ve eğitim hata oranı test hata oranından daha küçük çıkma eğilimindedir.

Test hata oranını doğrudan tahmin etmek için kullanılabilecek belirlenmiş test kümesinin yokluğunda, mevcut eğitim verilerini kullanarak bu miktarı tahmin etmek için bir takım teknikler kullanılmaktadır. Bazı yöntemler, test hata oranını tahmin etmek için eğitim hata oranına matematiksel bir yaklaşım sunar. Burada test hata oranı, model oluşturma sürecinden gelen eğitim gözlemlerinin bir alt kümesini kullanarak ve daha sonra bu alt kümenin gözlemlerine istatistiksel öğrenme yöntemini uygulayarak tahmin eden yöntemlerdir.[46]

[45] James ve diğerleri, s.176
[46] James ve diğerleri, s.176.

2.1.1.1 Doğrulama Seti Yaklaşımı (Hold-out)

Bir dizi gözleme belirli bir istatistiksel öğrenme yönteminin uygulanmasıyla ile ilişkili test hatasını tahmin etmek istendiği varsayılsın. Mevcut gözlem kümesi eğitim seti ve doğrulama seti veya tutma seti (hold-out set) olarak rastgele iki bölüme ayrılır. Model eğitim setiyle tahmin edilir ve uygun model, doğrulama (test) setindeki gözlemlerin tahmini değerini elde etmek için kullanılır. Elde edilen doğrulama (validasyon) set hata oranı - tipik olarak nicel bir tahmin durumunda ortalama hata kare (MSE) kullanılarak değerlendirilir - test hata oranının bir tahmini elde edilir.[47]

Şekil 2.3. Doğrulama seti yaklaşımının şematik görüntüsü[48]

Özet olarak, n gözlem rastgele bir eğitim setine (diğerleri arasında mavi, 7, 22 ve 13 gözlemleri içeren) ve bir doğrulama setine (diğerlerinin yanı sıra bej olarak gösterilen ve gözlem 91'i içeren) ayrılmıştır. İstatistiksel öğrenme metodu eğitim setine uyar ve performansı doğrulama (validasyon) setinde değerlendirilir.

Doğrulama seti yaklaşımı, kavramsal olarak basittir ve uygulanması kolaydır. Fakat iki potansiyel dezavantajı vardır:[49]

1. Test hata oranının tahmini, hangi gözlemlerin eğitim setine dahil edildiğine ve hangi gözlemlerin doğrulama (validasyon) setine dahil edildiğine bağlı olarak oldukça değişken olabilir.

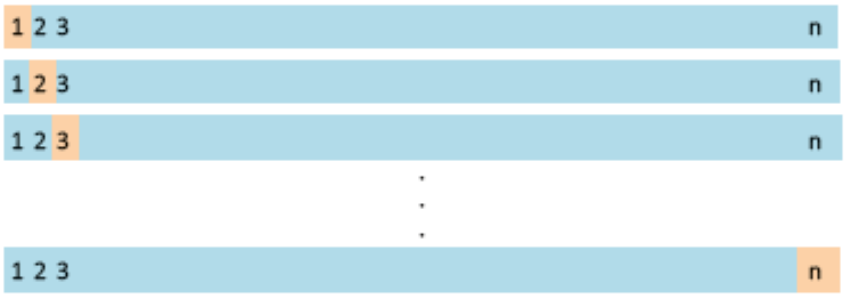
2. Doğrulama yaklaşımında, modele uymak için gözlemlerin sadece bir alt kümesi (doğrulama setinden ziyade eğitim setine dahil olanlar) kullanılır. İstatistiksel yöntemler daha az gözlemde eğitildiğinde daha

[47] James ve diğerleri, s.176.
[48] James ve diğerleri, s.177.
[49] James ve diğerleri, s.178.

kötü performans gösterdiğinden, bu doğrulama seti hata oranının, tüm veri kümesine uyan model için test hata oranını arttırma eğiliminde olabilir.

2.1.1.2 Tek-Çıkışlı Çapraz Doğrulama (Leave-One-Out Cross-Validation)

Tek çıkışlı çapraz doğrulama, doğrulama seti yaklaşımı ile ilişkilidir ama bu yöntemin dezavantajlarını kapatmaya çalışır. Doğrulama seti yaklaşımı gibi, tek çıkışlı çapraz doğrulama gözlem kümesini iki parçaya ayırır. Ancak, karşılaştırılabilir boyutta iki alt küme oluşturmak yerine, doğrulama kümesi için tek bir gözlem (x_1, y_1) kullanılır ve kalan gözlemler $\{(x_2, y_2),...,(x_n, y_n)\}$ eğitim setini oluştur. İstatistiksel öğrenme yöntemi n-1 tane eğitim gözlemiyle oluşturulur ve x_1 değeri kullanılarak dışlanan gözlem için bir tahmin $\hat{y}_1$ elde edilir. Model oluşturma işleminde (x_1, y_1) kullanılmadığından $MSE_1 = (y_1 - \hat{y}_1)^2$ test hatası için yaklaşık yansız bir tahmin sağlar. Ancak, MSE_1 test hatası için yansız olsa da, zayıf bir tahmindir çünkü çok değişkendir, tek bir gözlem (x_1, y_1) üzerine kurulmuştur.[50]



Şekil 2.4. Tek çıkışlı çapraz doğrulamanın şematik görüntüsü.[51]

[50] James ve diğerleri, s.179.
[51] James ve diğerleri, s.179.

Şekilde *n* tane veri noktaları kümesi, bir gözlem hariç diğer gözlemleri içeren bir eğitim setine (maviyle gösterilen) ve yalnızca bir gözlemi içeren bir doğrulama setine (bej olarak gösterilmiştir) tekrar tekrar bölünür.

Doğrulama verileri için (x_2, y_2) öğesini seçerek işlemleri tekrarlayabiliriz, n -1 gözlem için eğitim gözlemleri $\{(x_1, y_1), (x_3, y_3)..., (x_n, y_n)\}$'dir ve $MSE_2 = (y_2 - \hat{y}_2)^2$ olarak hesaplanır. Bu yaklaşımı n kere tekrarlayarak, $MSE_1, ...MSE_n$ olacak şekilde n kare hatalar üretilir. Test MSE için tek çıkışlı çapraz doğrulama seti tahmini, bu n test hatası tahminlerinin ortalamasıdır:

$$CV_n = \frac{1}{n} \sum_{i=1}^{n} MSE_i \qquad (2.1)$$

Tek çıkışlı çapraz doğrulama seti, doğrulama seti yaklaşımına göre birkaç avantajı vardır. İlk olarak, daha az yansızdır. Tek çıkışlı çapraz doğrulama setinde, tüm veri kümesinde olduğu gibi, *n-1* gözlem içeren eğitim seti kullanılarak istatistiksel öğrenme yöntemi tekrar tekrar uygulanır. Bu, eğitim setinin tipik olarak orijinal veri kümesinin boyutunun yaklaşık yarısı olduğu doğrulama seti yaklaşımının tersidir. İkincisi, tek çıkışlı çapraz doğrulama seti yaklaşımında, doğrulama hatası yaklaşımı kadar test hata oranı büyümemektedir. İkinci olarak, doğrulama setindeki rastlantısallık tekrar tekrar uygulandığında farklı sonuçlar veren doğrulama yaklaşımının tersine, tek çıkışlı çapraz doğrulama setinin birden çok kez uygulanması yakın sonuçlar verecektir: doğrulama seti bölünmelerindeki rastgelelik yoktur.[52]

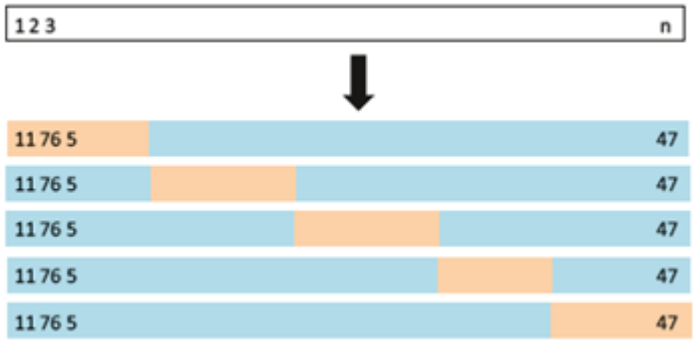
2.1.1.3 K-Katmanlı Çapraz Doğrulama (K-Fold Cross-Validation)

Tek çıkışlı çapraz doğrulamaya bir alternatif k-katmanlı çapraz doğrulamadır. Bu yaklaşım, gözlem kümesini rastgele yaklaşık olarak eşit büyüklükteki k gruplarına veya katlara ayırır. İlk katman bir doğrulama (test) seti olarak kabul edilir ve kalan $k - 1$ katmanla model oluşturulur. Ortalama kare hata MSE_1 dışarıda kalan kattaki gözlemler üzerinden hesaplanır. Bu işlem, k kez tekrarlanır; her defasında farklı bir

[52] James ve diğerleri, s.180.

gözlem grubu bir doğrulama (test) seti olarak kullanılır. Bu süreç, test hatasının $MSE_1,...MSE_k$ olacak biçimde k tane tahminle sonuçlanır. K- kat çapraz doğrulama tahmini bu değerlerin ortalaması alınarak hesaplanır.[53]

$$CV_k = \frac{1}{k}\sum_{i=1}^{k} MSE_i \qquad (2.2)$$



Şekil 2.5. K-katmanlı doğrulama seti yaklaşımının şematik görüntüsü[54]

Şekilde bir dizi n gözlem, rastgele birbiriyle çakışmayan beş gruba ayrılır. Bu beşliklerin her birinden bej renkli olanlar doğrulama (test) seti ve geri kalan mavi renkli olanlar da eğitim setleridir. Test hatası, ortaya çıkan beş MSE tahmininin ortalaması alınarak tahmin edilir.

Tek çıkışlı çapraz doğrulama da *k'nın n'ye* eşit olduğu özel bir k-katmanlı çapraz doğrulamadır. Uygulamada genelde k = 5 veya k = 10 kullanılarak k-katmanlı çapraz doğrulama gerçekleştirir. Tek çıkışlı çapraz doğrulama, istatistiksel öğrenme yöntemini *n* kere uygulanmasını gerektirir. Bu hesaplama açısından özellikle de *n*'nin çok büyük olduğu durumlarda zaman alan bir yöntemdir. Buna karşılık, 10 katmanlı çapraz doğrulama yapmak öğrenme yöntemine yalnızca on kez uygulanmasını gerektirir ki bu daha mümkün hesaplama yöntemidir.[55]

[53] James ve diğerleri, s.181.
[54] James ve diğerleri, s.181.
[55] James ve diğerleri, s.183.

2.1.1.4 K-Katmanlı Çapraz Doğrulama İçin Yanlı Varyans Ödünleşimi

K-katmanlı çapraz doğrulamanın az görülen fakat potansiyel olarak daha önemli bir avantajı test hata oranında, tek çıkışlı çapraz doğrulamadan çok daha doğru tahminler vermesidir. Bu yaklaşımda, istatistiksel öğrenme yöntemini uygulamak için kullanılan eğitim seti, tüm veri kümesinin yalnızca yarısını içerir. Bu mantığı kullanarak, tek çıkışlı çapraz doğrulamanın test hatasının yaklaşık olarak yansız tahmini vereceğini görmek zor değil, çünkü her bir eğitim seti *n-1* gözlemleri içerir, bu da neredeyse tam veri setindeki gözlem sayısı kadardır. $k = 5$ veya $k = 10$ için k-katmanlı çapraz doğrulama yapılması, her bir eğitim seti $(k-1)n/k$ gözlem içerir ve bu gözlem sayısı tek çıkışlı çapraz doğrulama yaklaşımından daha az, fakat büyük ölçüde doğrulama seti yaklaşımından daha fazla olduğundan orta düzeyde yanlılığa yol açacaktır. Bu nedenle, yanlılığı azaltmadan dolayı, tek çıkışlı çapraz doğrulamanın k-katmanlı çapraz doğrulamaya tercih edilmesinin gerektiği açıktır. Tek çıkışlı çapraz doğrulama gerçekleştirildiğinde, her biri hemen hemen aynı gözlem kümesi üzerinde eğitilmiş modellerin çıktıları ortalanmaktadır; bu nedenle, bu çıktılar birbirleriyle yüksek derecede(pozitif) ilişkilidir. Aksine $k < n$ ile k-katmanlı çapraz doğrulama yapıldığında, her modeldeki eğitim setleri arasındaki örtüşme daha küçük olduğundan, birbiriyle daha az ilişkili olan *k* uyumlu modellerin çıktıları ortalanır. Tek çıkışlı çapraz doğrulamadan kaynaklanan test hatası tahmini, k-katmanlı çapraz doğrulamadan kaynaklanan test hata tahmininden daha az bir varyansa sahip olma eğilimindedir.

k-katmanlı çapraz doğrulamada k seçimi ile ilişkili bir yanlı varyans vardır. Bu hususlar göz önüne alındığında $k = 5$ veya $k = 10$ kullanılarak k-katmanlı çapraz doğrulama gerçekleştirilir, çünkü bu değerler ampirik olarak ne yüksek yanlı ne de yüksek varyansa sahip olmayan tahminleri elde etmek için kullanılabilir.[56]

[56] James ve diğerleri, s.184.

2.1.1.5 Sınıflandırma Problemlerinde Çapraz Doğrulama

Sınıflandırma problemlerinde MSE yerine yanlış sınıflandırılmış gözlemlerin sayısı kullanılır. Örneğin, sınıflandırma probleminde, tek çıkışlı çapraz doğrulamadan hata oranı

$$CV_n = \frac{1}{n}\sum_{i=1}^{n} Err_i \qquad (2.3)$$

formunu alır. Burada $Err_i = I\left(y_i \neq \hat{y}_i\right)$ yanlış sınıflandırılmış gözlem sayısına eşittir. K-katmanlı çapraz doğrulama hata oranı ve doğrulama seti hata oranları sınıflandırma problemleri için benzer şekilde tanımlanmıştır.[57]

2.1.2. Bootstrap

Efron yeniden örneklemeyi, ana kütleden elde edilen örneklemden alt örneklemlerin seçilmesi olarak tanımlamaktadır.[58] Bootsrap, belirli bir tahmin ediciyle ya da istatistiksel öğrenme yöntemiyle ilişkili belirsizliği ölçmek için kullanılabilecek yaygın ve uygulanabilir bir istatistik araçtır. Basit bir örnek olarak, bootstrap, lineer regresyon uyumundan katsayıların standart hatalarını tahmin etmek için kullanılabilir. Bununla birlikte, Bootsrap'nin gücü, bir takım değişkenlik ölçümlerinin elde edilmesi zor olan ve istatistiksel yazılım tarafından otomatik olarak çıkarılamayan, çok çeşitli istatistiksel öğrenme yöntemlerine kolaylıkla uygulanabilmesidir.[59]

Açıklayıcı birörnek vermek gerekirse, X ve Y'nin rasgele değişkenler olduğu durumda, sırasıyla X ve Y finansal varlıklarına sabit bir para yatırmak istediği varsayılsın. Paranın bir kısmını α kadar X'e ve geriye kalan $1-\alpha$ Y'ye yatırılsın. Bu iki varlık üzerindeki getirilerle ilgili değişkenlik olduğundan, yatırımın toplam riskini veya varyansını en aza indirmek için α'nın ne olacağı belirlenmek istenir. Diğer bir

[57] James ve diğerleri, s.184.
[58] Dilek Altaş and Murat Çinko, 'Bootstrap Yönteminin Rıdge Regresyonda Uygulanması', Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi Cilt XXII, Sayı 1, 2003, s.281–292.
[59] James ve diğerleri, s.187

deyişle $Var\left(\alpha X+(1-\alpha)Y\right)$ minimize etmek istenir. Riskleri en aza indiren değer α değeri aşağıdaki denklemle gösterilir.[60]

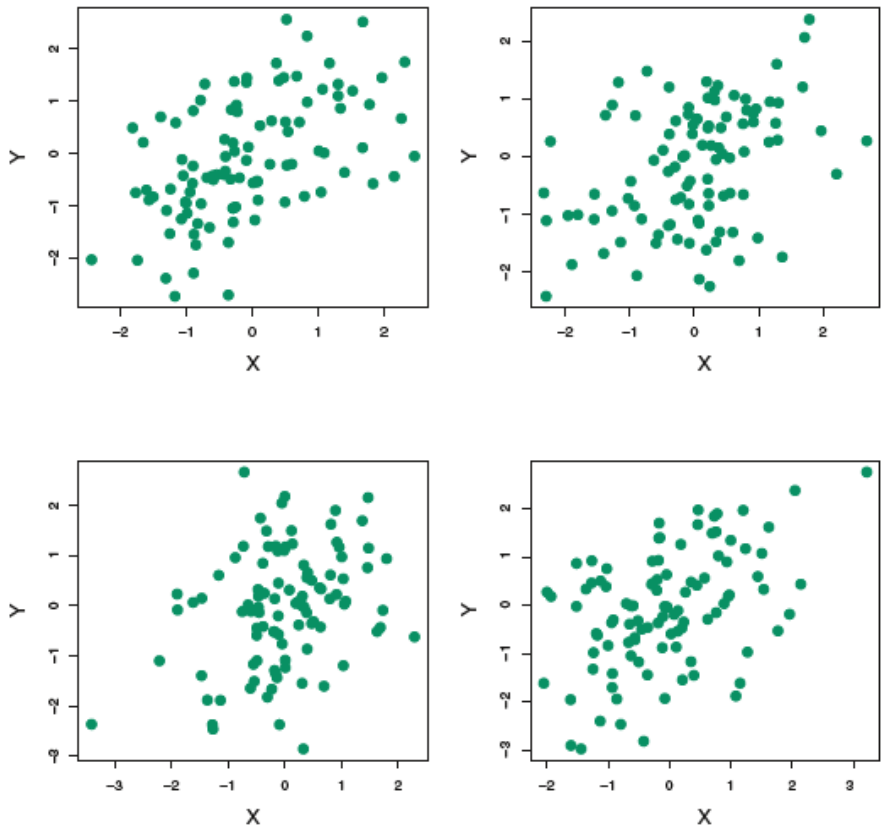
$$\alpha=\frac{\sigma_Y^2-\sigma_{XY}}{\sigma_X^2+\sigma_Y^2-2\sigma_{XY}} \qquad (2.4)$$

Burada $\sigma_X^2=Var(X)$, $\sigma_Y^2=Var(Y)$ *ve* $\sigma_{XY}=Cov(X,Y)$, i göstermektedir.

Gerçekte, σ_X^2,σ_Y^2 *ve* σ_{XY} değerleri bilinmez ancak *X* ve *Y* için geçmiş ölçümleri içeren bir veri kümesi kullanılarak bunların tahminleri hesaplanır. Üstü şapkalı olanlar daha sonra, yatırımın varyansını en aza indiren α değeriyle tahmin edilebilir.

$$\hat{\alpha}=\frac{\hat{\sigma}_Y^2-\hat{\sigma}_{XY}}{\hat{\sigma}_X^2+\hat{\sigma}_Y^2-2\hat{\sigma}_{XY}} \qquad (2.5)$$

[60] James ve diğerleri, s.187



Şekil 2.6. Her bir grafik, X ve Y yatırımları için 100 tane gözlem içeren simüle edilmiş getiriyi gösterir.[61]

Şekil 2.6'da Soldan sağa ve yukarıdan aşağıya, α için elde edilen tahminler 0.576, 0.532, 0.657 ve 0.651'dir.

α tahmininin doğruluğu ölçülmek istenir. α'nın standart sapmasını tahmin etmek için *X* ve *Y* yatırımlarından 100 tane gözlem içeren simüle etme işlemini tekrar edilir ve α'yı denklem (2.5) 1.000 kere kullanılarak tahmin edilir. Bu simülasyonlar için parametreler $\sigma_x^2 = 1$, $\sigma_y^2 = 1.25$ ve $\sigma_{XY} = 0.5$ olarak ayarlanır ve böylece α'nın gerçek değerinin 0.6 olduğu bilinir.

α için tüm tahmini değerlerin ortalaması

$$\bar{\alpha} = \frac{1}{1.000} \sum_{r=1}^{1.000} \hat{\alpha}_r = 0,5996$$

[61] James ve diğerleri, s.188

$\alpha = 0,6$ 'ya çok yakındır. Tahminlerin standart hatası da

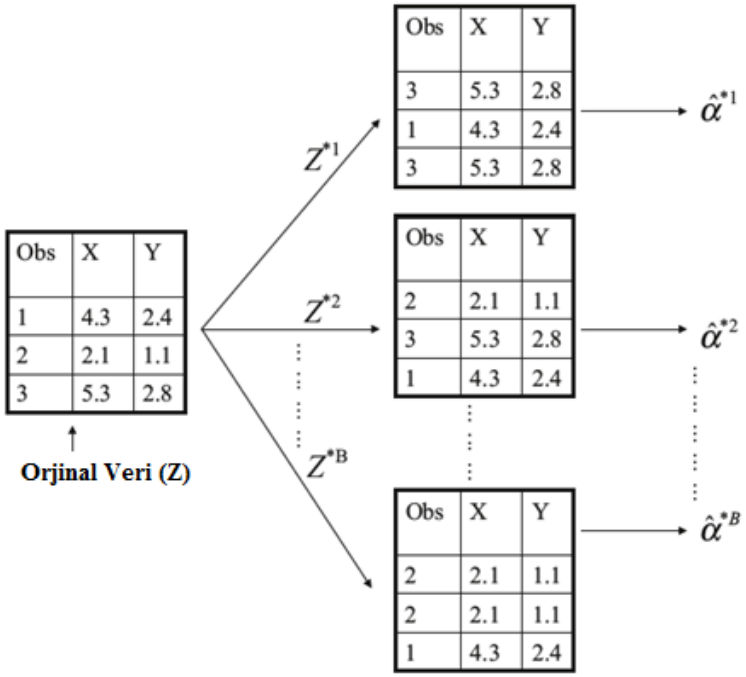
$\sqrt{\frac{1}{1.000-1}\sum_{r=1}^{1.000}\left(\hat{\alpha}_r - \bar{\alpha}\right)^2} = 0,083$ olarak bulunur.

Bir bootstrap veri seti oluşturmak için veri kümesinden rastgele bir şekilde n gözlemli Z^{*1} veri setini seçilir. Örnekleme, iadeli olarak gerçekleştirilir, yani aynı gözlem, bootstrap veri seti bir kereden fazla yer alabilirr. $\hat{\alpha}^{*1}$ olarak adlandırılan α 'nın yeni bir bootstrap tahmini oluşturmak için Z^{*1}'i kullanılabilir. Bu süreç, B farklı bootstrap veri seti üretmek için B kez tekrarlanır.(burada B büyük bir değeri ifade etmektedir) $Z^{*1}, Z^{*2}, ..., Z^{*B}$ 'ye karşılık gelen α tahminleri $\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, ..., \hat{\alpha}^{*B}$ 'dir. Böylece elde edilen bootstrap tahminlerinin standart hatası aşağıdaki formül kullanılarak hesaplanır.[62]

$$SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}^{*r} - \frac{1}{B}\sum_{r'=1}^{B}\hat{\alpha}^{*r'}\right)^2} \qquad (2.6)$$

Bu, orijinal veri kümesinden tahmin edilen $\hat{\alpha}$ 'nın standart hata tahminidir.

[62] James ve diğerleri, s.189



Şekil 2.7. n= 3 gözlem içeren küçük bir örnek üzerinde bootstrap yaklaşımının grafiksel bir gösterimi.

Yukarıdaki şekilde boostrap örneklemesi üç gözlemli veri setinde örneklendirilmiştir.

2.2. Karar Ağaçlarında Değerlendirme

Bu bölümde karar ağaçları değerlendirmesinde ana kavram ve kalite kriterleri tanıtılacaktır. Daha çok sınıflandırma ağaçlarındaki değerlendirme kriterlerinden bahsedilecektir.

2.2.1 Genelleme Hatası

KA(S), *S* veri seti üzerinde eğitilmiş bir sınıflandırma ağacını temsil etsin. *KA(S)*'nin genelleme hatası, örnek uzayının *D* dağılımına göre seçilen bir örneği yanlış sınıflandırma olasılığıdır. Bir sınıflandırma ağacının sınıflandırma doğruluğu, 1 (bir)

eksi genelleme hatasıdır. Eğitim hatası, sınıflandırma ağacı tarafından yanlış şekilde sınıflandırılmış eğitim kümesindeki örneklerin yüzdesi olarak tanımlanır:[63]

$$\varepsilon\left(KA(S),S\right)=\sum_{\langle x,y\rangle\in S} L\left(y,KA(S)(x)\right) \qquad (2.7)$$

Burada $L\left(y,KA(S)(x)\right)=\begin{cases}1 \ eğer & y\neq KA(S)(x)\\ 0 \ eğer & y=KA(S)(x)\end{cases}$ olarak tanımlanan 0-1 kayıp fonksiyonudur.

Eğitim hatası, genelleme hatasının bir tahmini olarak alınabilir. Bununla birlikte, eğitim hatasının kullanılması, özellikle ağacın eğitim verisini aşırı öğrenmesi durumunda düşük seviyelerde olma eğilimindedir.[64]

2.2.1.1 Doğruluk Ölçüsüne Alternatifler

Doğruluk, sınıfların dengesiz dağıldığı bir modeli değerlendirmek için yeterli bir ölçüm değildir. Veri kümesinde, azınlık sınıf örneklerinden çok daha fazla çoğunluk sınıfı içerdiği durumlarda, her zaman çoğunluk sınıfı seçilebilir ve iyi bir doğruluk performansı elde edebilir. Bu nedenle, bu durumlarda, hassasiyet ve özgüllük ölçümleri doğruluk ölçümlerine alternatif olarak kullanılabilir.

Sınıflandırma problemlerinde 2 sınıflı bir bağımlı değişkenin aldığı değerler pozitif negatif olarak kaydedilirse;

Duyarlılık (sensitivity) sınıflandırıcının pozitif örnekleri ne kadar iyi tanıdığını ölçer.

$$Duyarlılık=\frac{doğru\_pozitif}{pozitif} \qquad (2.8)$$

Burada doğru_pozitif, gerçekte pozitif olan ve model tarafından tahmin edilen pozitif örneklerin sayısına karşılık gelir ve pozitif, gerçekte olan pozitif örnek sayısıdır.[65]

[63] Lior Rokach ve Oded Maimon, Data Mining With Decision Tree Theory and Applications,2.Edition,World Scientific Publishing,2015, s.31
[64] Rokach ve Maimon, s.32

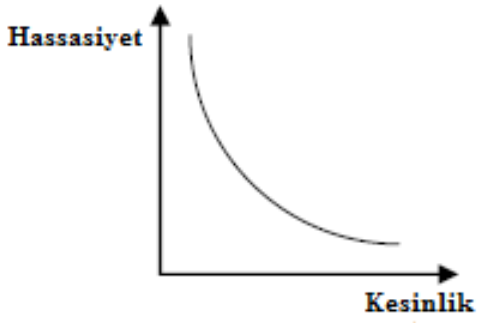
Özgüllük(specificity), sınıflandırıcının negatif örnekleri ne kadar iyi tanıdığını ölçer.

$$Özgüllük = \frac{doğru\_negatif}{negatif} \qquad (2.9)$$

Burada doğru_negatif, gerçekte negatif olan model tarafından negatif olarak tahmin edilen örneklerin sayısına karşılık gelir ve denklemdeki negatif, gerçek negatif örnek sayısıdır.[66]

İyi bilinen bir başka performans ölçüsü de kesinliktir (precision). Kesinlik "pozitif" sınıf olarak sınıflandırılan örneklerin gerçekten "pozitif" olduğunu ölçer. Bu ölçü, tüm bir veri kümesini sınıflandırmak için kullanılan sınıflandırıcıları değerlendirmek için yararlıdır.

$$Kesinlik = \frac{doğru\_pozitif}{doğru\_pozitif + yanlış\_pozitif} \qquad (2.10)$$



Şekil 2.8. Duyarlılık-Kesinlik diyagramı[67]

Yukarı tanımlardan hareketle doğruluk, hassasiyet ve özgüllüğün fonksiyonu olarak tanımlanabilir.

[65] Rokach ve Maimon, s.34
[66] Rokach ve Maimon, s.34
[67] Rokach ve Maimon, s.35

$$Doğruluk = Duyarlılık \cdot \frac{pozitif}{pozitif + negatif} + Özgüllük \cdot \frac{negatif}{pozitif + negatif} \quad (3.11)$$

2.2.1.2 Karmaşıklık Matrisi (Confusion Matrix)

Karmaşıklık matrisi, sınıflandırma kuralının özelliklerinin bir göstergesi olarak kullanılır. Her sınıf için doğru veya yanlış sınıflandırılmış öğelerin sayısını içerir. Ana diyagonal üzerinde, her sınıf için doğru şekilde sınıflandırılmış gözlemlerin sayısını görülebilir; diyagonal olmayan elemanlar, yanlış sınıflandırılmış gözlemlerin sayısını gösterir. Bu matrisinin bir yararı, sistemin iki sınıfı karıştırıp karıştırmadığını görmenin anlaşılır olmasıdır.

Test kümesindeki her örnek için, gerçek sınıf, tahmin edilen sınıflarla karşılaştırılır. Sınıflandırıcı tarafından doğru bir şekilde sınıflandırılan pozitif (negatif) bir örnek, bir DP (doğru negatif) olarak adlandırılır; yanlış sınıflandırılan pozitif (negatif) bir örnek, yanlış negatif (yanlış pozitif) olarak adlandırılır.[68]

Tablo 2.1. Hata matrisi gösterimi[69]

Hata Matrisi		
	Tahmin edilen negatif	Tahmin edilen pozitif
Negatif gözlemler	a	b
Pozitif gözlemler	c	d

Tablo 2.1'den yola çıkarak yukarıda bahsedilen formülleri aşağıdaki gibi yazabiliriz.[70]

- $Doğruluk = \frac{(a+d)}{(a+b+c+d)}$

[68] Rokach ve Maimon, s.36
[69] Rokach ve Maimon, s.37
[70] Rokach ve Maimon, s.37

- $Yanlış\ Sınıflandırma\ Oranı = \frac{(b+c)}{(a+b+c+d)}$
- $Kesinlik = \frac{d}{(b+d)}$
- $Doğru\ Pozitif\ Oran\ (Duyarlılık) = \frac{d}{(c+d)}$
- $Yanlış\ Pozitif\ Oran = \frac{b}{(a+b)}$
- $Doğru\ Negatif\ Oran\ (Özgüllük) = \frac{a}{(a+b)}$
- $Yanlış\ Negatif\ Oran = \frac{c}{(c+d)}$

2.2.1.3 Sınırlı Kaynaklar Altında Sınıflandırıcı Değerlendirme

Yukarıda belirtilen değerlendirme ölçütleri, olasılıklı sınıflandırıcılar sınırlı bir kotaya dâhil edilecek nesneleri seçmek için kullanıldığında yetersiz kalmaktadır. Bu, fayda-maliyet unsurları gerektiren kaynak kısıtlamaları nedeniyle gerçek yaşam uygulamalarında ortaya çıkan yaygın bir durumdur. Kaynak sınırlamaları, sistemin tüm örneklerinin seçilmesini engeller. Örneğin, doğrudan pazarlama uygulamalarında, listedeki herkese mail göndermek yerine, pazarlama çalışmalarında sınırlı bir kota uygulanması, yani pazarlama bütçesini aşmadan olumlu cevap verme olasılığı en yüksek olan kitlenin hedeflenmesi gerekir.[71]

Karar verici, sınıflandırıcının beklenen performansını değerlendirmek ister. Örneğin, bazı ülkelerde, bir devlet üniversitesinde belirli bir bölüm için kabul edilebilecek lisans öğrencilerinin sayısı sınırlıdır. Belirli bir yıl için gerçek kota, hükümet bütçesi de dahil olmak üzere farklı parametrelere göre belirlenir. Bu durumda karar verici, gerçek kota boyutunu bilmemekle birlikte başvuru sahiplerini seçmek için

[71] Rokach ve Maimon, s.37

birkaç sınıflandırıcı ile değerlendirmek ister. En uygun sınıflandırıcıyı önceden bulmak önemlidir çünkü seçilen sınıflandırıcı, önemli özelliklerin ne olduğunu, yani başvuru sahibinin kayıt ve kabul biriminin sağlaması gereken bilgileri belirleyebilir.[72]

Olasılıksal sınıflandırıcılarda, yukarıda bahsedilen hassasiyet ve kesinlik tanımları genişletilebilir ve bir olasılık eşiği olan τ'nın bir fonksiyonu olarak tanımlanabilir. Eğer bir sınıflayıcıyı $\{\langle x_1, y_1\rangle,...,\langle x_n, y_n\rangle\}$ olan gözlemden oluşan belirli bir test setine göre değerlendirirsek, burada x_i örnek *i*'nin giriş özelliklerini temsil eder ve y_i gerçek sınıfını(pozitif veya negatif) temsil eder.

$$Kesinlik(\tau) = \frac{\left|\{\langle x_i, y_i\rangle : \hat{P}_{KA}(pos|x_i) > \tau, y_i = pos\}\right|}{\left|\{\langle x_i, y_i\rangle : \hat{P}_{KA}(pos|x_i) > \tau\ \}\right|} \qquad (2.12)$$

$$Hassasiyet(\tau) = \frac{\left|\{\langle x_i, y_i\rangle : \hat{P}_{KA}(pos|x_i) > \tau, y_i = pos\}\right|}{\left|\{\langle x_i, y_i\rangle : y_i = pos\ \}\right|} \qquad (2.13)$$

Burada *KA*, x_i gözleminin koşullu olasılığını $\hat{P}_{KA}(pos|x_i)$, "pozitif" olarak tahmin etmek için kullanılan olasılıksal bir sınıflandırıcıyı temsil eder.

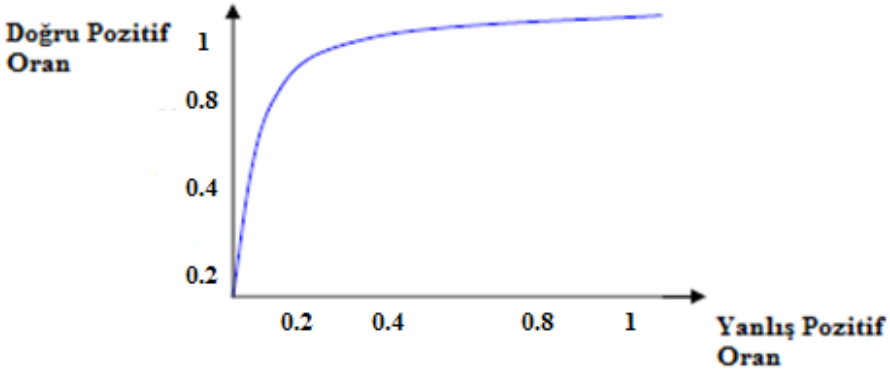
Genelde eşik değer 0,5 olarak alınır, bu da bir gözlemin "pozitif" olarak tahmin edilmesi için 0,5'ten yüksek olması gerektiği anlamına gelir. τ değerinin değiştirilmesiyle, pozitif olarak sınıflandırılan örneklerin sayısı kontrol edilebilir. Böylece, τ istenen bir kota büyüklüğüne ayarlanabilir.[73]

Yukarıdaki tartışma, sınıflandırma probleminin ikili olduğu varsayımına dayanmaktadır. İkiden fazla sınıfın olduğu durumlarda, bir sınıfın diğerleriyle karşılaştırılmasıyla analizin kolaylıkla yapılabilmesi mümkündür.

[72] Rokach ve Maimon, s.38
[73] Rokach ve Maimon, s.39

2.2.1.3.1 ROC Eğrisi (Receiver Operating Characteristic Curve)

ROC eğrileri, doğru pozitif ile yanlış pozitif oranlar arasındaki ilişkiyi göstermektedir.[74] X ekseni yanlış pozitif bir oranı temsil eder ve Y-ekseni doğru pozitif oranı temsil eder. ROC eğrisindeki ideal nokta (0,100), yani tüm pozitif örnekler doğru olarak sınıflandırılır ve negatif örnekler pozitif olarak yanlış sınıflandırılmaz.



Şekil 2.9. ROC eğrisi diyagramı[75]

ROC dışbükey gövdesi, optimal sınıflandırıcıları tanımlamak için güçlü bir yöntem olarak da kullanılabilir.[76] ROC eğrilerinin bir ailesi verildiğinde, ROC dışbükey gövdesi, ROC alanının kuzey-batı sınırına doğru olarak daha fazla olan noktaları içerebilir. Böylelikle iyi sınıflandırma yapılmış olduğu söylenebilir.[77]

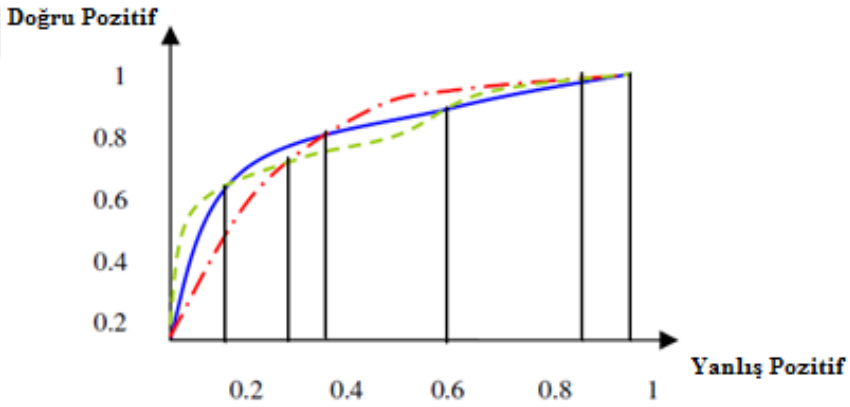
2.2.1.3.2 Eğri Altında Kalan Alan (Area Under Curve) (AUC)

Belirli bir sabit kota kullanmadan olasılıksal bir modelin değerlendirilmesi önemlidir. Daha önce sözü edilen ROC eğrilerinin kullanılması sorunlu olabilir. Bu tür ölçütler, "en iyi model hangisi?" sorusuna kesin bir cevap verebilir, ancak sadece bir model eğri uzayına hâkim olursa, diğer tüm modellerin eğrilerinin altında ya da tüm grafik alanı boyunca bu eğriye eşit olduğu anlamına gelir. Eğer hakim bir model yoksa,

[74] Foster Provost and Tom Fawcett, 'Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions', KDD-97 Proceedings, (1997), s.43–48.
[75] Rokach and Maimon,s.39
[76] Foster Provost and Tom Fawcett, 'Robust Classification for Imprecise Environments', Machine Learning, 42.3 (2001), s.203–231 <https://doi.org/10.1023/A:1007601015854>.
[77] Rokach ve Maimon, s.39

o zaman ROC eğrisini kullanarak bu soruya cevap verilemez. Elbette, pratikte tam anlamıyla en iyi model yoktur. Verilen en iyi cevap, bir modelin diğerlerinden daha iyi performans gösterdiği alanlarla ilgilidir. Şekil 2.10'da gösterildiği gibi, her model farklı alanlarda farklı değerler almaktadır. Model performansının tam bir düzenine ihtiyaç varsa, başka bir ölçüm kullanılmalıdır.[78]

ROC eğrisinin altındaki alan (AUC), seçilen karar kriterinden ve önceki olasılıklardan bağımsız olduğu için, sınıflandırıcı performansı açısından yararlı bir metriktir. AUC karşılaştırması, sınıflandırıcılar arasında bir hakimiyet ilişkisi kurabilir. ROC eğrileri kesişiyorsa, toplam AUC, modeller arasındaki ortalama bir karşılaştırma yapabilir.[79] Diğer ölçümlerin aksine, ROC eğrisinin altındaki alan (AUC) eğitim setinin dengesizliğine bağlı değildir.[80] Dolayısıyla, iki sınıflandırıcının AUC değerlerinin karşılaştırılması, yanlış sınıflandırma oranlarının karşılaştırılmasından daha adil ve bilgilendirici niteliktedir.[81]



Şekil 2.10. Eğrilerin üstünlük alanları.

ROC eğrisi, modellerin tam bir sırasını değil, hakimiyet alanlarını veren bir ölçü örneğidir. Bu örnekte, eşit olarak kesikli çizgili yeşil eğri yanlış pozitif (YP) < 0,2

[78] Rokach ve Maimon, s.43
[79] Sauchi Stephen Lee, 'Noisy Replication in Skewed Binary Classification', Computational Statistics and Data Analysis, 34.2 (2000), s.165–191 <https://doi.org/10.1016/S0167-9473(99)00095-X>.
[80] Aleksander Kolcz, Abdur Chowdhury, and Joshua Alspector, 'Data Duplication: An İmbalance Problem?', 2003 <https://doi.org/10.1.1.72.8356-1>.
[81] Rokach ve Maimon, s.43

için en iyisidir. Mavi renkli eğri 0,2<YP<0,4 için en iyisidir. Noktalı çizgi kırmızı eğri 0,4 < YP < 0,9 için en iyisidir ve 0.9'dan 1'e kadar kesikli çizgi modeli en iyisidir.

2.3. Bölünme Kriterleri

Çoğu karar ağacında ayırma işlevleri tek değişkenlidir, bir iç düğüm tek bir özelliğin değerine göre bölünür. Sonuç olarak, karar ağacı bölünmeyi gerçekleştirecek en iyi özniteliği (bağımsız değişkeni) arar.[82]

2.3.1 Katışıklık (Safsızlık) Tabanlı Kriter

K tane ayrık değere sahip bir rastgele değişken X verildiğinde, $P=(p_1, p_2,..., p_k)$, ye göre dağılım gösteren bir safsızlık (katışıklık) ölçüsü olan $\phi:[0,1]^k \rightarrow R(\square)$ aşağıdaki koşulları sağlayan bir fonksiyondur.

- $\phi(P) \geq 0$.
- Eğer $p_i = 1$ yapacak şekilde bir *i* bileşeni varsa $\phi(P)$ minimumdur.
- Eğer $\forall i, \quad 1 \leq i \leq k, \quad p_i = 1/k$ ise $\phi(P)$ maksimumdur.
- $\phi(P)$ P'nin bileşenlerine göre simetriktir.
- $\phi(P)$ tanımlandığı aralıkta sürekli ve türevlidir.

Olasılık vektörünün 1 bileşeni varsa (değişken x yalnızca bir değer alır), değişken saf olarak tanımlanır. Öte yandan, tüm bileşenler eşitse, katışıklık seviyesi maksimuma ulaşır.

Bir eğitim seti *S* verildiğinde, bağımlı değişken *y*'nin olasılık vektörü şu şekilde tanımlanır:[83]

[82] Rokach ve Maimon, s.61
[83] Rokach ve Maimon, s.62

$$P_y(S) = \left(\frac{\left|\sigma_{y=c_1} S\right|}{|S|}, \ldots, \frac{\left|\sigma_{y=c_{|dom(y)|}} S\right|}{|S|} \right) \qquad (2.14)$$

Ayrık öznitelik a_i'den dolayı bölünmenin iyiliği, $v_{i,j} \in dom(a_i)$ değerlerine göre *S*'nin bölümlenmesinden sonra hedef özniteliğin (bağımlı değişkenin) safsızlığında (katışıklığında) bir azalma olarak tanımlanır:

$$\Delta\Phi(a_i, S) = \phi\left(P_y(S)\right) - \sum_{j=1}^{|dom(a_i)|} \frac{\left|\sigma_{a_i=v_{i,j}} S\right|}{|S|} \cdot \phi\left(P_y\left(\sigma_{a_i=v_{i,j}} S\right)\right) \qquad (2.15)$$

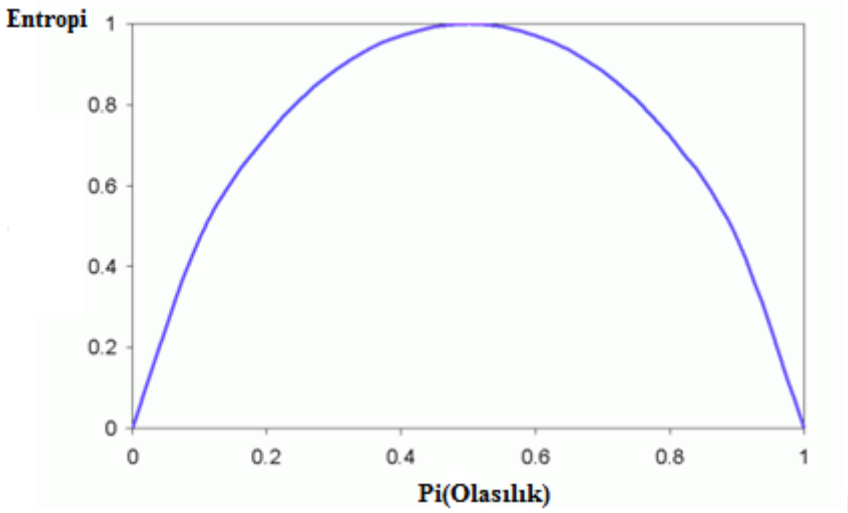
2.3.2 Entropi

Bir sistemdeki belirsizliğin ölçüsüne entropi denir. Enformasyon teorisinde genellikle *Shannon Entropisi* olarak da ifade edilen entropi, tesadüfi bir değişkendeki belirsizliğin ölçüsüdür. Diğer bir deyişle öngörülemezliğin (rastlantısallığın) bir ölçüsüdür. Entropi tipik olarak *bit, nat* veya *ban* gibi birimlerle ölçülür. *Nit* veya *nepit* olarak da isimlendirilen *nat,* enformasyonun veya entropinin *2* tabanlı logaritma yerine, *e* tabanlı doğal logaritma ile hesaplanması ile elde edilen sonucun birimidir. Benzer şekilde hesaplamalar *10* tabanlı logaritma ile yapıldığında elde edilen sonucun birimi ban olarak isimlendirilir. *S* bir kaynak olsun. Bu kaynağın $\{m_1, m_2, \ldots, m_n\}$ olmak üzere *n* tane mesaj üretebildiği varsayılsın. Tüm mesajlar birbirinden bağımsız olarak üretilir ve m_i mesajlarının üretilme olasılıkları p_i olsun. $P = \{p_1, p_2, \ldots, p_n\}$ dağılımına sahip çıktılar üreten *S* kaynağının entropisi *H(S)* aşağıdaki gibi hesaplanır.[84]

$$H(S) = -\sum_{i=1}^{n} p_i \log_2^{(p_i)} \qquad (2.16)$$

Eğer veri serisi tamamen homojen ise entropi sıfırdır ve örneklem eşit olarak bölünürse entropisi birdir.

[84] Claude E Shannon, 'A Mathematical Theory of Communication', The Bell System Technical Journal, 27.July 1928 (1948), s.379–423 <https://doi.org/10.1145/584091.584093>.



Şekil 2.11. Olasılık değerlerine göre entropi eğrisi

Bir değişkenin entropi ölçüsü ne kadar fazla ise, o alan için elde edilen sonuçlar da o derece belirsizdir. Bu nedenle karar ağaçlarını oluşturmaya başlarken kök düğüm olarak entropi ölçüsü en az olan değişkenler kullanılır.[85]

2.3.3 Bilgi Kazancı (İnformation Gain)

Bilgi Kazancı, entropi ölçümünü katışıklık ölçümü olarak kullanan (bilgi teorisinden kaynaklanan) katışıklık (safsızlık) tabanlı bir kriterdir.[86] Hedef niteliğini (bağımlı değişken) ifade eden T, hedef niteliği olmayan (bağımsız değişken) bir X değişkenin değerine bağlı olarak $T_1, T_2, \ldots, T_n$ alt kümelerine ayrılırsa, T'nin bir elemanının sınıfını belirlemek için gerekli bilgi, T_i 'nin bir elemanının sınıfının belirlenmesinde gerekli olan bilginin ağırlıklı ortalaması kabul edilir. Bu tanıma bağlı olarak T'nin bir elemanının sınıfını belirlemek için gerekli bilgi aşağıdaki şekilde hesaplanır:[87]

$$H(X,T) = \sum_{i=1}^{n} \frac{T_i}{T} H(T_i) \qquad (2.17)$$

[85] Gülpınar,s.75
[86] Rokach and Maimon, s.62
[87] Özkan, s.58

Burada $H(T_i)$, T_i değişkenine bağlı olarak hesaplanan entropi değeridir. T veri tabanını X testine göre bölmekle elde edilen bilgileri ölçmek için 'kazanç ölçütü (Bilgi Kazancı)' adı verilen bir ölçüte başvurulur. Bu ölçüt aşağıdaki şekilde tanımlanır:

$$Kazanç(X,T) = H(T) - H(X,T) \qquad (2.18)$$

Burada ayırma işlemi yapılırken *Kazanç (X,T)* değeri en yüksek olan değişken seçilerek kazancı maksimize edebilecek X testinin seçilmesi amaçlanır.[88]

2.3.4 Gini İndeksi

Karar ağaçlarında kullanılan Gini indeksinin İtalyan istatistikçi ve sosyolog Corrado Gini tarafından 1912 yılında yayımlanan Variabilitia' e mutabilitia' isimli makalesinde açıklanan ve bulucusuna atfen Gini Katsayısı, Gini Endeksi olarak isimlendirilen ve milli gelirin ülke yaşayanları arasındaki istatistiki paylaşımını ölçen endeks ile karşılaştırılmaması gerekir.[89]

Gini endeksi, hedef özellik (bağımlı değişken) değerlerinin olasılık dağılımları arasındaki ayrışmaları ölçen katışıklık (safsızlık) temelli bir ölçüttür. Gini indeksi, aşağıda verilen adımlara sahiptir.[90]

$$Gini(y,S) = 1 - \sum_{c_j \in dom(y)} \left(\frac{\left| \sigma_{y=c_j} S \right|}{|S|} \right)^2 \qquad (2.19)$$

Sonuç olarak, a_i özniteliğini seçmek için değerlendirme kriteri şu şekilde tanımlanır:

$$GiniGain(a_i,S) = Gini(y,S) - \sum_{v_{i,j} \in dom(a_i)} \frac{\left| \sigma_{a_i=v_{i,j}} S \right|}{|S|} \cdot Gini\left(y, \sigma_{a_i=v_{i,j}} S\right) \qquad (2.20)$$

Bu algoritma, nitelik değerlerinin sağ ve solda olarak olmak üzere 2 bölüme ayrılması esasına dayanmaktadır.[91]

---

[88] Özkan, s.58
[89] Akpınar, Data: Veri Madenciliği Veri Analizi, s.212
[90] Rokach ve Maimon, s.63

Nitelikler için hesaplanan Gini değerleri içinden en küçük olan seçilir ve bölünme bu değişken üzerinden yapılır.

2.3.5 Kazanç Oranı

Hedef niteliğini (bağımlı değişken) ifade eden *T*, hedef niteliği olmayan (bağımsız değişken) bir *X* değişkenin değerine bağlı olarak $T_1, T_2, \ldots, T_n$ alt kümelerine ayrılırsa kazanç oranı, bilgi kazancını aşağıdaki gibi normalleştirir.[92]

$$Kazanç\,Oranı\left(X_i, T\right) = \frac{Kazanç\left(X_i, T\right)}{Entropi\left(X_i, T\right)} \qquad (2.21)$$

Payda sıfır olduğunda bu oran tanımlanmaz. Ayrıca, oran paydanın çok küçük olduğu nitelikleri destekleme eğiliminde olabilir. Buna göre, oranın iki aşamada yapılması önerilmektedir. İlk olarak, tüm özellikler için bilgi kazancı hesaplanır. En azından ortalama bilgi kazanımının yanı sıra gerçekleştirilmiş öznitelikleri dikkate almanın bir sonucu olarak, en iyi oran kazanımını elde eden öznitelik seçilir. Quinlan (1988) kazanım oranının, hem doğrulukta hem de sınıflandırıcı karmaşıklığı açısından basit bilgi kazancı kriterlerinden daha iyi performans gösterdiğini belirtmiştir.

2.3.6 İkili Kriter – Binary Criteria

İkili kriter, ikili karar ağaçları oluşturmak için kullanılır. Bu ölçüm, girdi özniteliğinin (bağımsız değişkenin) alanının iki alt alana bölünmesine dayanır. $\beta\left(a_i, dom_1\left(a_i\right), dom_2\left(a_i\right), S\right)$, $dom_1\left(a_i\right)$ ve $dom_2\left(a_i\right)$'nin karşılık gelen alt bölgeleri olduğunda, *S* örneği üzerinde öznitelik (değişken) a_i için ikili kriter değerini gösterir. Öznitelik alanının iki karşılıklı ayrık ve kapsamlı alt alana en uygun şekilde ayrılması için elde edilen değer, öznitelikleri karşılaştırmak için kullanılır.[93]

$$\beta^*\left(a_i, S\right) = \max_{\forall dom_1\left(a_i\right); dom_2\left(a_i\right)} \beta\left(a_i, dom_1\left(a_i\right), dom_2\left(a_i\right), S\right) \qquad (2.22)$$

---

[91] Özkan, s.106
[92] J.Ross Quinlan, C4.5 Programs For Machine Learning, California, Morrgan Kaufman Publisher, 1993, s.23
[93] Rokach ve Maimon, s.65

2.3.7 Twoing Kriteri

Hedef özniteliğin (bağımlı değişken) etki alanı geniş olduğunda Gini indeksi sorunlarla karşılaşabilir.[94]Bu gibi durumlarda, twoing kriteri adı verilen ikili kriter kullanılabilir.

Algoritma şu şekilde çalışmaktadır.[95]

Adım 1

a) Niteliklerin içerdiği değerler göz önüne alınarak eğitim kümesi iki ayrı dala ayrılır. Bunlara aday bölünme adı verilir. Bir *t* düğümünde sol ve sağ olmak üzere iki ayrı dal bulunur. Bu bölümlenen kümeler t_{sol} ve $t_{sağ}$ biçimindedir.

b) Aday bölümlerin her biri için P_{sol} ve p(j|t_{sol}) olasılıkları hesaplanır. Söz konusu olasılıklar aşağıda verilmektedir. Burada $p(j|t_{sol})$ ifadesi bir *j* sınıf değerinin sol tarafdaki bölünmede olma olasılığını verir. Söz konusu *j* değerleri sınıf değerlerinin yer aldığı nitelik olarak göz önüne alınır.

$$p_{sol} = \frac{t_{sol}\text{'daki herbir nitelik değerinin ilgili nitelik sütunundaki tekrar sayısı}}{\text{Eğitim kümesindeki kayıtların sayısı}} \quad (2.23)$$

$$P(j \mid t_{sol}) = \frac{t_{sol}\text{'daki kayıtların } j \text{ sınıfları sayısı}}{t_{sol}\text{'daki herbir nitelik değerinin ilgili nitelik sütunundaki tekrar sayısı}} \quad (2.24)$$

c) Aday bölümlerin her biri için $P_{sağ}$ ve p(j|$t_{sağ}$) olasılıkları hesaplanır. Söz konusu olasılıklar aşağıda verilmektedir. Burada $p(j|t_{sağ})$ ifadesi bir *j* sınıf değerinin sağ tarafdaki bölünmede olma olasılığını verir.

$$p_{sağ} = \frac{t_{sağ}\text{'daki herbir nitelik değerinin ilgili nitelik sütunundaki tekrar sayısı}}{\text{Eğitim kümesindeki kayıtların sayısı}} \quad (2.25)$$

[94] Leo Breiman and others, Classification And Regression Trees, Chapman & Hall/CRC Texts in Statistical Science Series, 1984, s.105 i <https://doi.org/10.1002/widm.8>.
[95] Özkan, s.89

$$P(j \mid t_{sağ}) = \frac{t_{sağ}\text{'daki kayıtların } j \text{ sınıfları sayısı}}{t_{sağ}\text{'daki herbir nitelik değerinin ilgili nitelik sütunundaki tekrar sayısı}} \qquad (2.26)$$

a. $\Phi(s \mid t)$, t düğümündeki s aday bölünmelerinin uygunluk ölçüsü olsun. Söz konusu uygunluk ölçüsü şu şekilde hesaplanır.

$$\Phi(s \mid t) = 2 p_{sol} p_{sağ} \sum_{j=1}^{n} \left| P(j \mid t_{sol}) - P(j \mid t_{sağ}) \right| \qquad (2.27)$$

b. $\Phi(s \mid t)$ değerleri hesaplandıktan sonra içinde en büyük olan seçilir. Bu değerin ilgili olduğu aday bölünme satırı bize dallanmanın yapılacağı satırı bildirecektir.

c. Dallanma bu şekilde yapıldıktan sonra bu adıma ilişkin olan karar ağacı çizilir.

Adım 2

Algoritmanın birinci adımına dönülerek ağacın alt kümesine aynı işlemler uygulanır.

2.4. Budama Kriterleri

Karar ağaçları öğreniminde budama (pruning) süreci, bir bakıma karar ağaçlarının optimize edilme sürecidir. Budama süreci, doğal ağaçlarda olduğu gibi ağacın daha sağlıklı olabilmesi için, giderek incelen dalların kesilmesi işlemidir. Doğal süreçten farkı, uzaklaştırılan yaprak veya dallarda bulunan nesnelerin, hiyerarşik yapı içerisinde kendilerinden daha üstte yer alan düğümlere ilave edilmesidir. Bir karar ağacının büyümesi önceden tanımlanan kriterlere uygun bir biçimde sona erdirilebilir. Diğer bir stratejide bu kriterler olabildiğince esnek tutularak ağacın karar ağaçlarının olabildiğince büyümesi ve daha sonra optimize edilmesi sağlanır.[96]

[96] Akpınar, Data: Veri Madenciliği Veri Analizi, s.217

2.4.1 Durdurma Kriteri

Bir durdurma kriteri tetiklenene kadar ağaç büyüme devam eder. Bu durum regresyon analizindeki gibidir yani adım adım regresyon simülasyonları belirli bir noktayı geçtikten sonra, ek değişkenlerin eklenmesi R^2'nin (determinasyon katsayısı) çok az bir değişim meydana getirebilir. Bu durum ağaçlarda, bir ağaç için, doğru büyüklükteki ağaçtan daha yüksek bir yanlış sınıflandırma oranına sahip olması olarak ifade edilir.[97]

Aşağıdaki koşullar ortak durdurma kurallarıdır:[98]

- Eğitim kümesindeki tüm örnekler hedef değişkenin tek bir değerine ait olduğunda,
- Maksimum ağaç derinliğine ulaşıldığında,
- Terminal düğümündeki vaka sayısı, üst düğümler için minimum vaka sayısından daha az olduğunda,
- Düğüm bölünmüşse, bir veya daha fazla alt düğümdeki vaka sayısı, alt düğümler için minimum vaka sayısından daha az olduğunda,
- En iyi bölme kriteri belirli bir eşikten daha büyük olmadığında,

Ağaç büyümeyi durdurur. Durma kriterleri ağaç büyümesini durduran nispeten kaba bir yöntem iken, yapılan çalışmalar ağacın performansını düşürmeye eğilimli olduklarını göstermiştir. Büyümeyi durdurmak için alternatif bir yaklaşım, ağacın büyümesine izin vermek ve daha sonra onu en uygun büyüklüğe geri getirmektir.[99]

2.4.2 Sezgisel Budama (Heuristic Pruning)

Dar (kritik) durma kriterlerini kullanmak, küçük ve eksik öğrenmiş karar ağaçları yaratma eğilimindedir. Öte yandan, gevşek (hafif) durdurma kriterleri

[97] Breiman ve diğerleri, s.60
[98] Rokach ve Maimon, s.69
[99] Rokach ve Maimon, s.69

kullanmak, eğitim setini aşırı öğrenmiş büyük karar ağaçları üretme eğilimindedir. Bu ikilemi çözmek için Breiman ve arkadaşları (1984), gevşek bir durdurma ölçütüne dayanan ve karar ağacının eğitim setine aşırı öğrenmesine izin veren bir budama metodolojisi geliştirmiştir. Daha sonra, aşırı öğrenmiş ağaç, genelleme doğruluğuna katkıda bulunmayan alt dalları kaldırarak daha küçük bir ağaca doğru budanır. [100]

Budamanın diğer bir önemli özelliği Bohanec ve Btratko (1994) tarafından sunulan "basitlik için ticaret doğruluğu" dır. Amaç, yeterince hassas, kompakt bir konsept açıklaması yapmak ise, budama oldukça faydalıdır. Bu süreç içinde ilk karar ağacı tam olarak doğru olduğu için, budamış bir karar ağacının doğruluğu, ilk ağaca ne kadar yakın olduğunu gösterir.[101]

Karar ağaçlarının budaması için çeşitli teknikler vardır. Çoğu, düğümlerin yukarıdan aşağıya veya aşağıdan yukarıya doğru hareket etmesini sağlar. Bu işlem belirli bir kriteri iyileştirirse bir düğüm kesilir.[102]

2.4.2.1 Maliyet-Karmaşıklık Budama (Cost Complexity Pruning)

Maliyet karmaşıklığı budama (zayıf bağlantı budama veya hata karmaşıklığı budama olarak da bilinir) iki aşamada ilerler.. İlk aşamada bir dizi ağaç $T_0, T_1, \ldots, T_k$ eğitim verileri üzerinde inşa edilir burada T_0 budama öncesi orijinal ağaç ve T_k kök ağacıdır.[103]

İkinci aşamada, bu ağaçlardan biri, genelleme hatası tahminine dayalı olarak budanmış ağaç olarak seçilir.

T_{i+1} ağacı, bir önceki ağaçta T_i bulunan alt ağaçların bir veya daha fazlasının uygun yapraklarla değiştirilmesiyle elde edilir. Budanmış olan alt ağaçlar, budanmış yaprak başına görünen hata oranındaki en düşük artışı sağlayanlardır:

---

[100] Rokach ve Maimon, s.69
[101] Marko Bohanec and Ivan Bratko, 'Trading Accuracy for Simplicity in Decision Trees', Machine Learning, 15.3 (1994), s.223–250 , <https://doi.org/10.1023/A:1022685808937>.
[102] Rokach ve Maimon, s.70
[103] Rokach ve Maimon, s.70

$$\alpha = \frac{\varepsilon\left(pruned\left(T,t\right),S\right) - \varepsilon\left(T,S\right)}{\left|leaves\left(T\right)\right| - \left|leaves\left(pruned\left(T,t\right)\right)\right|} \qquad (2.28)$$

Burada $\varepsilon(T,S)$ S veri seti üzerinde T ağacının hata oranını gösterir ve $\left|leaves(T)\right|$ T'deki yaprak sayısını gösterir. $pruned(T,t)$ T içindeki düğüm t'nin uygun bir yaprak ile değiştirilmesiyle elde edilen ağacı gösterir.[104]

İkinci aşamada, her budanmış ağacın genelleme hatası tahmin edilir. En iyi budanmış ağaç daha sonra seçilir. Verilen veri kümesi yeterince büyükse, veri setinin bir eğitim setine ve bir budama setine ayırması önerilir. Ağaçlar eğitim seti kullanılarak inşa edilir ve budama setinde değerlendirilir. Diğer yandan, eğer verilen veri kümesi yeterince büyük değilse, hesaplama karmaşıklığı ile ilgili söylemlere rağmen çapraz doğrulama metodolojisini kullanılması önerilir.

2.4.2.2 Azaltılmış Hata Budama (Reduced Error Pruning)

Azaltılmış hata budama olarak bilinen karar ağaçları için basit bir tekniktir, Quinlan (1987) tarafından önerilmiştir. İç düğümler alttan üste doğru geçiş yaparken, algoritma en sık sınıfla değiştirmenin, ağaçların doğruluğunu azaltmamasını belirlemek için her bir iç düğümü kontrol eder. Doğruluk azalmazsa düğüm budanır. Prosedür daha fazla budamanın doğruluğu azaltana kadar devam eder.[105]

Doğruluğu tahmin etmek için, Quinlan (1987) budama setinin kullanımını önermektedir. Bu prosedürün belirli bir budama setine göre en küçük doğru alt ağaç ile bittiği gösterilebilir.

2.4.2.3 Minimum Hata Budama (Minimum Error Pruning) (MEP)

Niblett ve Bratko (1986) tarafından önerilen MEP, iç düğümlerin aşağıdan yukarıya geçişini içerir. Bu teknik, her düğümde, budama ile ve bulamayan *1-olasılık* hata oranı tahminini karşılaştırmaktadır. *1-olasılık* hata oranı tahmininin frekanslarını

[104] Rokach ve Maimon, s.70
[105] Rokach ve Maimon, s.70

kullanarak basit olasılık tahminini düzenler. S_t, yaprak t'ye ulaşan örnekleri gösterirse, bu yapraktaki beklenen hata oranı şöyledir:[106]

$$\varepsilon(t) = 1 - \max_{c_i \in dom(y)} \frac{\left|\sigma_{y=c_i} S_t\right| + l \cdot Papr(y = c_i)}{\left|S_t\right| + 1} \qquad (2.29)$$

Burada $Papr(y = c_i)$ c_i değeri için y'nin önsel olasılığıdır ve l önsel olasılığa verilen ağırlığı gösterir.

İç düğümün hata oranı, alt dallarının hata oranının ağırlıklı ortalamasıdır. Ağırlık, her daldaki örneklerin oranına göre belirlenir.

Hesaplamalar yapraklara kadar tekrarlı olarak yapılır. Bir iç düğüm budanırsa, bu bir yaprak olur ve hata oranı doğrudan son denklem kullanılarak hesaplanır. Sonuç olarak, belirli bir iç düğümü budamadan ve sonra hata oranını karşılaştırılabilir. Bu düğümün budaması hata oranını artırmazsa, budama kabul edilir.

2.4.2.4 Kötümser Budama (Pessimistic Pruning)

Kötümser budama, budama setinin veya çapraz doğrulamanın gerekliliğini ortadan kaldırır ve bunun yerine kötümser istatistiksel korelasyon testini kullanır.[107]

Temel fikir, eğitim seti kullanılarak tahmin edilen hata oranının yeterince güvenilir olmamasıdır. Bunun yerine, binom dağılımı için devamlılık düzeltmesi olarak bilinen daha gerçekçi bir ölçüm kullanılmalıdır:[108]

$$\varepsilon'(T,S) = \varepsilon(T,S) + \frac{\left|leaves(T)\right|}{2 \cdot \left|S\right|} \qquad (2.30)$$

Ancak, bu düzeltme hala iyimser bir hata oranı üretiyor. Sonuç olarak, Quinlan hata oranı bir referans ağaç kabul edilebilir bir standart hatada ise, dahili bir düğüm t'nin budanmasını önermektedir.

---

[106] Rokach ve Maimon, s.71
[107] J. R. Quinlan, 'Simplifying Decision Trees', International Journal of Man-Machine Studies, 27.3 (1987), s.221–234, <https://doi.org/10.1016/S0020-7373(87)80053-6>.
[108] Rokach ve Maimon, s.71

$$\varepsilon'\left(\,pruned\left(T,t\right),S\right)\leq\varepsilon'\left(T,S\right)+\sqrt{\frac{\varepsilon'\left(T,S\right)\cdot\left(1-\varepsilon'\left(T,S\right)\right)}{|S|}} \qquad (2.31)$$

Son durum oranlar için istatistiksel güven aralığına dayanır. Genellikle, son koşul, *T* kökleri iç düğüm olan *t* bir alt ağacı ifade eder ve S, eğitim setinin *t* düğümüne karşılık gelen kısmını belirtir.

Kötümser budama işleyişi, iç düğümler üzerinde yukarıdan aşağıya doğru gerçekleşir. Bir iç düğüm budanırsa, tüm alt düğümleri budama işleminden çıkarılır ve bu da nispeten hızlı bir budama ile sonuçlanır.[109]

2.4.2.5. Hata Tabanlı Budama (Error-Based Pruning) (EBP)

EBP, kötümser budamanın gelişmiş halidir. Kötümser budamada olduğu gibi, hata oranı, oranlar için istatistiksel güven aralığının üst sınırı kullanılarak tahmin edilir.[110]

$$\varepsilon_{UB}\left(T,S\right)=\varepsilon\left(T,S\right)+Z_{\alpha}\cdot\sqrt{\frac{\varepsilon'\left(T,S\right)\cdot\left(1-\varepsilon'\left(T,S\right)\right)}{|S|}} \qquad (2.32)$$

Burada $\varepsilon\left(T,S\right)$ S eğitim seti üzerindeki T ağacının yanlış sınıflandırma hatasını gösterir. Z standart normal kümülatif dağılım fonksiyonunun tersidir ve α istenen önem seviyesidir.

Alt ağaç *(T, t)[subtree(T,t)],* düğüm *t* tarafından oluşturulan alt ağacı belirtir. *Maxchild(T, t) t*'nin en sık görülen çocuk düğümünü gösterir (yani *S* deki örneklerin çoğu bu çocuğa ulaşır.) S_t, *S*'deki tüm düğümleri *t* noktasına ulaştıran tüm örnekleri göstersin. Prosedür aşağıdan yukarıya tüm düğümleri çaprazlar ve aşağıdaki değerleri karşılaştırır:[111]

[109] Rokach ve Maimon, s.72
[110] Rokach ve Maimon, s.72
[111] Rokach ve Maimon, s.72

1) $\varepsilon_{UB}\left(subtree\left(T,t\right),S_t\right)$

2) $\varepsilon_{UB}\left(pruned\left(subtree\left(T,t\right),t\right),S_t\right)$

3) $\varepsilon_{UB}\left(subtree\left(T,maxchild\left(T,t\right)\right),S_{maxchild\left(T,t\right)}\right)$

En düşük değere göre, algoritma ya ağacın tepesini düğümsüz olarak bırakılır ya da düğüm *t* alt köküyle birlikte *maxchild(T, t)* ile değiştirir.

2.4.2.6. Budama Yöntemlerinin Karşılaştırılması (Comparison of Pruning Methods)

Çeşitli araştırmalar, farklı budama tekniklerinin performansını karşılaştırmaktadır. Sonuçlar, bazı yöntemlerin (maliyet karmaşıklık budama, azaltılmış hata budama gibi) aşırı-budama eğiliminde olduğunu, yani daha küçük fakat daha az doğru karar ağaçları oluşturduğunu göstermektedir. Diğer yöntemler (hataya dayalı budama, kötümser hata budaması ve minimum hata budaması gibi) az budama eğilimindedir. Karşılaştırmaların çoğu, diğer budama yöntemlerinden daha iyi performans gösteren bir budama yönteminin olmadığını ortaya çıkarmıştır.[112] Bir veri seti üzerinde iyi çalışan budama yöntemi başka bir veri seti üzerinde kötü sonuçlar verebilir.

[112] Rokach ve Maimon, s.72

3.BÖLÜM

KARAR AĞACI ALGORİTMALARI

Karar ağaçlarını öğrenmek için geliştirilmiş olan algoritmaların çoğu, olası karar ağaçlarının uzandığı yerde yukarıdan aşağı, açgözlü bir arama yapan bir çekirdek algoritmasında varyasyonlardır.[113] Bu bölümde, ID3, C4.5 ve CART dahil olmak üzere, popüler karar ağacı indüksiyon algoritmalarından bazılarını kısaca gözden geçirilecektir. Bahsedilecek algoritmaların tümü, önceki bölümlerde daha önce tarif edilmiş olan bölünme kriteri ve budama yöntemlerini kullanmaktadır. Bu nedenle, bu bölümün amacı, her algoritmanın hangi kriteri kullandığını ve her bir algoritmanın avantaj ve dezavantajlarının neler olduğunu belirtmektir.

3.1. Iterative Dichotomiser 3 *(*ID3)

ID3 algoritması çok basit bir karar ağacı algoritması olarak kabul edilir.[114] Bilgi kazanımını bir bölme kriteri olarak kullanılarak, tüm örnekler tek bir hedef özelliğe ait olduğunda veya en iyi bilgi kazanımı sıfırdan büyük olmadığında ID3 algoritması büyümeyi keser. ID3 herhangi bir budama prosedürü uygulamaz, sayısal nitelikler veya eksik değerleri işlemez. ID3'ün ana avantajı sadeliğidir. Bu sebepten dolayı ID3 algoritması sıklıkla öğretim amaçlı kullanılmaktadır. Bununla birlikte, ID3'ün bazı dezavantajları vardır.[115]

1) ID3, optimal bir çözümü garanti etmez, açgözlü bir strateji kullandığı için yerel optimumlarda sıkışabilir. Lokal optimumdan kaçınmak için, arama sırasında geri izleme kullanılabilir.

2) ID3, eğitim verilerini ezberleyebilir. Aşırı öğrenmeyi (ezberlemeyi) önlemek için daha büyük olanlara göre daha küçük karar ağaçları tercih edilmelidir. Bu algoritma genellikle küçük ağaçlar üretir, ancak her zaman mümkün olan en küçük ağacı üretmez.

---

[113] Alpaydin, s.55
[114] J. R. Quinlan, 'Induction of Decision Trees', Machine Learning, 1.1 (1986), 81–106 <https://doi.org/10.1023/A:1022643204877>.
[115] Rokach ve Maimon, s.77

3) ID3 nominal nitelikler için tasarlanmıştır. Bu nedenle, sürekli veriler nominal (ikili) verilere dönüştürüldükten sonra kullanılabilir.

Yukarıdaki dezavantajlar nedeniyle, uygulamacıların çoğu ID3 yerine C4.5 algoritmasını tercih etmektedir. Çünkü C4.5, ID3 algoritmasının dezavantajlarının kapatmaya çalışan bir yöntemdir.

3.2. C4.5

Quinlan tarafından sunulan ID3'ün gelişmiş hali C4.5, bölünme kriteri olarak kazanç oranını kullanır. Bölünme, bölünecek örneklerin sayısı belirli bir eşiğin altında olduğunda durur. Hataya dayalı budama, büyüme aşamasından sonra gerçekleştirilir.

C4.5 sayısal(nümerik) özellikleri işleyebilir. Ayrıca, düzeltilmiş kazanç oranı ölçütlerini kullanarak eksik değerleri içeren bir eğitim kümesinden de sonuç çıkarılabilir.[116]

C4.5 algoritması ID3'e birkaç iyileştirme sağlar. En önemlileri şunlardır:

1) C4.5, doğruluğa katkıda bulunmayan dalları çıkartan ve bunları yaprak düğümleriyle değiştiren bir budama prosedürü (kriteri) kullanır. Budama prosedürü olarak EBP'yi kulanmaktadır.

2) C4.5, değişken değerlerinin eksik olmasına izin verir

3) C4.5, özniteliğin değer aralığını iki alt kümeye bölerek sürekli öznitelikleri işler(İkili bölünme). Özellikle, kazanç oranı ölçütünü en üst düzeye çıkaran en iyi eşiği arar. Eşiğin üzerindeki tüm değerler birinci alt kümeyi oluşturur ve diğer tüm değerler ikinci alt kümeyi oluşturur.

C5.0, C4.5'e bir dizi iyileştirme sunan güncellenmiş, ticari bir sürümüdür: C5.0'ın bellek ve hesaplama zamanı açısından C4.5'ten çok daha verimli olduğu iddia edilmektedir. Bazı durumlarda, saatten bir buçuk saniyeye kadar sadece 3,5 saniyede etkileyici bir hızlanma sağlar.

[116] Rokach ve Maimon, s.78

J48, weka veri madenciliği programında C4.5 algoritmasının açık kaynaklı bir Java uygulamasıdır. J48 algoritması sadece C4.5'in yeniden yapılandırılması olduğundan, C4.5'e benzer şekilde performans göstermesi beklenir. Yine de, C4.5'i J48 ve C5.0 ile karşılaştıran yeni bir karşılaştırmalı çalışma Moore ve arkadaşları tarafından yapılmıştır. C4.5'in C5.0 ve J48'den daha iyi (doğruluk anlamında), özellikle küçük veri kümelerinde daha iyi performans göstermektedir.[117]

3.3. Sınıflama ve Regresyon Ağaçları (Clasification anda Regression Tree) (CART)

CART, Sınıflandırma ve Regresyon Ağaçları anlamına gelir. Breiman ve arkadaşları tarafından geliştirilmiştir ve ikili ağaç yapıları, yani her iç düğümün tam olarak dışarı giden iki dalı olduğu durumudur. Bölünmeler twoing kriteri ve gini kullanılarak seçilir ve elde edilen ağaç maliyet karmaşıklığı budama ile budanır. CART'da daha fazla kazanç elde edilemeyeceği düşünüldüğünde veya bazı durdurma kurallarıyla karşılaşıldığında dallanma durur.[118]

CART ağaç indüksiyonunda yanlış sınıflandırma maliyetlerini dikkate alabilir. Ayrıca olasılık dağılımlarını sağlamadan önce kullanıcılara olanak tanır.

CART'ın önemli bir özelliği, regresyon ağaçları oluşturma yeteneğidir. Regresyon durumunda, CART tahmini karesel hatayı en aza indiren bölünmeler arar (en küçük kareler sapması). Her bir yapraktaki tahmin, o düğüm için ağırlıklı ortalamaya dayanmaktadır.

3.3.1. Regresyon Ağaçları

Bir regresyon ağacı inşa etme sürecini ele alalım. Genel olarak iki adımdan bahsedilebilir. Tahmin uzayı –yani $X_1,X_2,\ldots,X_p$ için olası değerler kümesini J farklı ve birbiriyle örtüşmeyen $R_1,R_2,\ldots R_j$ farklı bölgeye ayırlır R_j bölgesine düşen her gözlem için, aynı öngörü yapılır, bu da Rj'deki eğitim gözlemleri için tahmin değerlerinin ortalamasıdır.

[117] Samuel A Moore, James Kurinskas, and Gary M Weiss, 'Are Decision Trees Always Greener on the Open (Source) Side of the Fence?', 2009, s.185–188.
[118] Rokach ve Maimon, s.79

Örneğin, 1. Adımda, iki bölge, R_1 ve R_2'nin elde edildiği ve birinci bölgedeki eğitim gözlemlerinin yanıt ortalamasının 10 olduğunu, ikinci bölgedeki eğitim gözlemlerinin yanıt ortalamasının ise 20 olduğu varsayılsın. Daha sonra, $X = x$ belirli bir gözlem için, eğer $x \in R_1$ ise 10 değerini veya $x \in R_2$ ise 20 değeri tahmin edilir.

$R_1,R_2,...R_j$ bölgelerini nasıl inşa edilir? Teoride, bölgeler herhangi bir şekle sahip olabilir. Bununla birlikte, basitlik açısından ve modelinin yorumlanmasını kolaylaştırmak için, veri kümesini yüksek boyutlu dikdörtgenlere veya kutulara ayırma seçilebilir. Amaç *RSS* (Residual Sum of Square) (Hata kareler toplamı)'yi minimize eden $R_1,R_2,...,R_j$ bölgelerini bulmaktır.[119]

$$\sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j} \right)^2 \qquad (3.1)$$

Burada $\hat{y}_{R_j}$, j bölgesindeki eğitim gözlemleri için ortalama (cevaptır) tahmin değeridir.

Ne yazık ki, özellik uzayının (veri kümesini) olası tüm bölümlerini J bölgelerine ayırmak hesaplama açısından mümkün değildir. Bu nedenle, öz yineli (öz çağrılı) ikili bölünme olarak bilinen yukarıdan aşağı, açgözlü bir yaklaşım ele alınır. Bu yaklaşım yukarıdan aşağıya doğrudur çünkü ağaç tepesinde başlar (bu noktada tüm gözlemler tek bir bölgeye aittir) ve daha sonra veri kümesi (öznitelik alanı) birbirinden ayırlır. Her bölünme, ağaç üzerinde aşağıya doğru iki yeni dal ile belirtilir. Bu yaklaşım açgözlüdür, çünkü ağaç oluşturma sürecinin her aşamasında, en iyi bölünme, ileriye bakmak ve gelecekteki bazı adımlarda daha iyi bir ağaca yol açacak bir bölünme seçmek yerine, o aşamada yapılır.[120]

İkili bölme yapmak için, önce tahminci X_j'yi ve kesme noktasının s değerini seçerek, tahmin edilecek uzay alanını $\{X \mid X_j < s\}$ ve $\{X \mid X_j \geq s\}$ bölgelerine bölerek, RSS'de mümkün olan en fazla azalma bulunmaya çalışılır. Yani, tüm tahmin edicileri $X_1,X_2,...,X_p$ ve öngörücülerin her biri için kesme noktasının s değerlerini göz önünde

[119] James ve diğerleri, s.306
[120] James ve diğerleri, s.306

bulundurulur ve daha sonra ortaya çıkan ağacın en düşük RSS'ye sahip olacak şekilde öngörücü ve kesme noktasını seçilir. Daha ayrıntılı olarak, herhangi bir *j* ve *s* için, yarı düzlem çiftleri tanımlanır;

$$R_1(j,s) = \{X \mid X_j < s\} \text{ ve } R_2(j,s) = \{X \mid X_j \geq s\} \qquad (3.2)$$

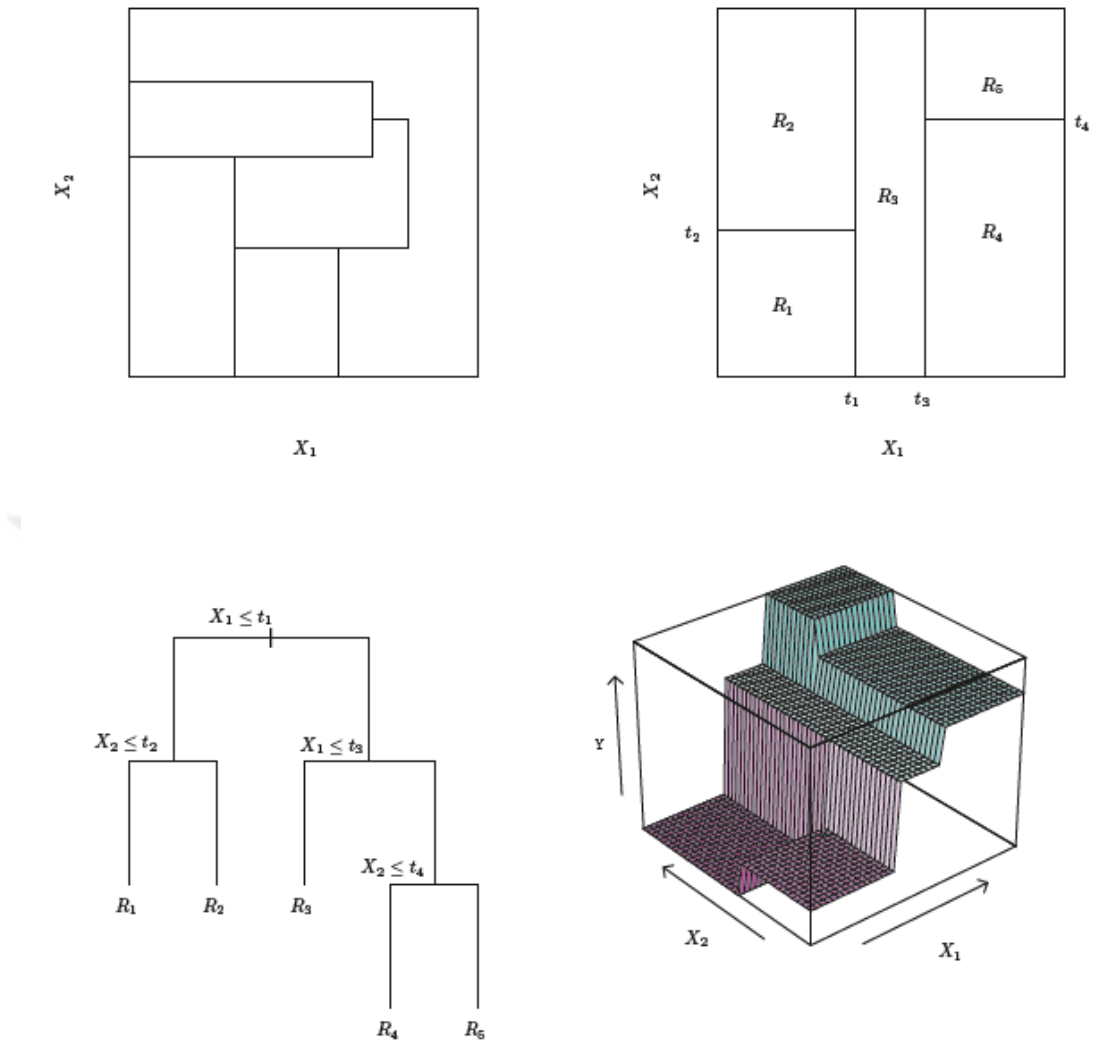
ve denklemi minimize eden *j* ve *s* değerini aranır.

$$\sum_{i:x_1 \in R_1(j,s)} \left(y_i - \hat{y}_{R_1}\right)^2 + \sum_{i:x_1 \in R_2(j,s)} \left(y_i - \hat{y}_{R_2}\right)^2 \qquad (3.3)$$

Burada $\hat{y}_{R_1}$ ve $\hat{y}_{R_2}$ sırasıyla $R_1(j,s)$ ve $R_2(j,s)$'deki eğitim gözlemleri tahminlerinin ortalamasıdır. Minimize edilen *j* ve *s* değerlerinin (3.3)'de bulunması, özellikle değişken sayısı çok büyük olmadığında oldukça hızlı bir şekilde yapılabilir.[121]

Daha sonra, elde edilen bölgelerin her birinde *RSS* en aza indirmek ve verileri daha fazla bölmek için en iyi öngörücü (tahmin ediciyi) ve en iyi kesme noktası arayan işlemler tekrar edilir. Bununla birlikte, bu sefer, tüm özellik alanını bölmek yerine, önceden tanımlanmış iki bölgeden biri bölünür. Böylelikle üç bölge elde edilir. Yine, bu üç bölgeden biri, *RSS*'yi en aza indirgemek için daha da bölünür. Süreç, durdurma kriterine ulaşılana kadar devam eder; Örneğin, hiçbir bölge beşten fazla gözlem içermeyene kadar devam edilebilir. $R_1, R_2, \ldots R_j$ bölgeleri oluşturulduktan sonra, bu test gözleminin ait olduğu bölgedeki eğitim gözlemlerinin ortalamasını kullanarak belirli bir test gözleminin cevabını tahmin edilir.[122]

[121] James ve diğerleri, s.307
[122] James ve diğerleri, s.307



Şekil 3.1. Sol üst: İkili bölünmeden kaynaklanamayan iki boyutlu özellik alanının bir bölümüdür. Sağ üst Sağ: İki boyutlu bir örnek üzerinde öz ikili bölünme çıktısı. Sol alt: Sağ üst paneldeki bölüme karşılık gelen bir ağaç. Alt Sağ: Bu ağaca karşılık gelen tahmin yüzeyinin perspektif çizimi.[123]

Bölüm 2.4.2.1'de anlatılan maliyet karmaşıklığı budamasına ek olarak; Her olası alt ağacı dikkate almak yerine negatif olmayan α parametresiyle indekslenen ağaç dizilerini dikkate alınır. α'nın her bir değeri için:

$$\sum_{m=1}^{|T|} \sum_{x_i \in R_m} \left(y_i - \hat{y}_{R_m} \right)^2 + \alpha |T| \qquad (3.4)$$

[123] James ve diğerleri, s.308

denklem (3.4)'deki ifade olabildiğince küçük olacak biçimde $T \subset T_0$ alt ağacına karşılık gelir. Burada $|T|$, T ağacının terminal düğümlerinin (yaprak) sayısıdır. R_m m. terminal düğüme karşılık gelen dikdörtgendir. (tahmin uzayının alt kümesi) ve $\hat{y}_{R_m}$, R_m ile ilişkili tahmini değerlerdir. – yani R_m'deki eğitim gözlemlerinin ortalamasıdır.

Ayarlama parametresi α alt ağacın karmaşıklığı ile eğitim verilerine uyumu arasındaki dengeyi kontrol eder. $\alpha = 0$ olduğunda alt ağaç T, T_0'a eşit olur, çünkü o zaman denklem (3.4) eğitim hatasını ölçer. Ancak α arttıkça bir çok terminal düğüme sahip olan ağaca karşılık gelen bir değer elde edilir. Böylece denklem (3.4) daha küçük bir alt ağaç için minimize edilmeye çalışılır.

Denklem (3.4) 'de α sıfırdan arttıkça, dallar ağaçtan budanmış olur ve öngörülebilir bir tarzda budanırlar, böylece α'nın bir fonksiyonu olarak tüm alt ağaçları elde etmek kolaydır. Doğrulama seti veya çapraz doğrulama kullanarak alfanın değeri seçilebilir. Daha sonra tüm veri setine dönüp α'ya karşılık gelen alt ağaç elde edilir.[124]

Özet olarak bir regresyon ağacı oluşturmak için aşağıdaki adımlar izlenir. [125]

1) Eğitim verileri üzerinde büyük bir ağaç oluşturmak için, her bir terminal düğümünün minimum sayıda gözlemden daha az olması durumunda durmak için tekrarlı ikili bölünme kullanılır.

2) α'nın bir fonksiyonu olarak, en iyi alt sıraların bir dizisini elde etmek için büyük ağaçlara maliyet karmaşıklığı budamasını uygulanır.

3) α 'yı seçmek için k-kat çapraz doğrulama kullanılır.

a) Eğitim setinin k'ıncı katı hariç adım 1 ve 2'yi tekrarlanır.

b) İhmal edilen k.kat'da, α'nın bir fonksiyonu olarak ortalama kare tahmini hatası değerlendirilir.

[124] James ve diğerleri, s.308
[125] James ve diğerleri, s.309

Her bir α değeri için sonuçları ortalanır ve ortalama hatayı minimize etmek için α seçilir.

4) Seçilen α değerine karşılık gelen 2. adımdan alt ağaç oluşturulur.

3.3.1 Sınıflandırma Ağaçları

Bir sınıflandırma ağacı regresyon ağacına çok benzerdir, ancak burada niceliksel verilerden ziyade niteliksel veriler tahmin edilir. Sınıflandırma ağacında, her bir gözlemin ait olduğu bölgedeki, eğitim gözlemleri en yaygın şekilde hangi sınıfa ait ise gözlemin o sınıfa olduğu tahmin edilir. Bir sınıflandırma ağacının sonuçlarının yorumlanmasında sadece belirli bir terminal düğüm bölgesine karşılık gelen sınıf tahmini değil, aynı zamanda bu bölgeye giren eğitim gözlemleri arasındaki sınıf oranlarıyla da ilgilenilir.[126]

Bir sınıflandırma ağacı oluşturma süreci, regresyon ağacı oluşturma sürecine oldukça benzerdir. Regresyon ağacında olduğu gibi bir sınıflandırma ağacı yetiştirmek için ikili bölünme kullanılır. Ancak sınıflandırma ağacında RSS ikili bölmeleri ayırmak için bir ölçüt olarak kullanılmaz. *RSS*'ye alternatif olarak sınıflandırma hata oranı kullanılmaktadır. Belirli bir bölgede, o bölgedeki en yaygın eğitim gözlemleri sınıfına karşılık gelen bir gözlemi atanmak istendiğinde, sınıflandırma hata oranı en yaygın sınıfa ait olmayan eğitim gözlemlerinin sayısına bölümüdür.

$$E = 1 - \max_{k} \left(\hat{p}_{mk} \right) \qquad (3.5)$$

Burada $\hat{p}_{mk}$ k'ıncı sınıfa ait olan m'inci bölgedeki eğitim gözlemlerinin oranını gösterir. Bununla birlikte, sınıflandırma hatasının, ağacın oluşması için yeterince hassas olmadığını ve pratikte diğer iki yöntemin daha tercih edilebilir olduğu ortaya çıkmaktadır. Bu yöntemler Gini indeksi ve entropi yöntemleridir. Bu yöntemler $\hat{p}_{mk}$ için yeniden yazılırsa;[127]

[126] James ve diğerleri, s.311
[127] James ve diğerleri, s.312

Gini indeksi;

$$G = \sum_{k=1}^{K} \hat{p}_{mk} \left(1 - \hat{p}_{mk}\right) \qquad (3.6)$$

K sınıfları boyunca toplam değişimin bir ölçüsü olarak tanımlanır. tüm $\hat{p}_{mk}$, ların sıfıra eşit veya sıfıra yakın olması durumunda Gini değerinin küçük bir değer aldığı görülür. Bu nedenle Gini indeksi düğüm saflığının bir ölçüsü olarak adlandırılır. küçük bir değer, bir düğümün tek bir sınıftan ağırlıklı olarak gözlemler içerdiğini gösterir.

Entropi;

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log_2^{(\hat{p}_{mk})} \qquad (3.7)$$

3.7'deki gibi tanımlanır. Eğer $\hat{p}_{mk}$'lerin hepsi sıfır veya sıfıra yakınsa entropinin sıfıra yakın bir değer alır. Bu nedenle, Gini indeksi gibi eğer *m*. düğüm saf ise entropi küçük bir değer alacaktır.

Bu iki yaklaşım sınıflandırma hata oranından çok düğüm saflığına daha duyarlı olduğu için, bir sınıflandırma ağacı oluşturulurken ya Gini indeksi yada entropi belirli bir bölünmenin kalitesini değerlendirmek için kullanılır.

Açıklanan bu üç yaklaşımdan herhangi biri ağaç budandığında kullanılabilir, ama amaç son budanmış ağacın tahmin doğruluğu olduğunda sınıflandırma hata oranı tercih edilebilir.

Karar ağaçları nitel bağımsız değişkenlerin varlığında da oluşturulabilir. Bu değişkenlerdeki ayırma, nitel değerlerin bir kısmının bir dala ve kalan diğer kısımların başka bir dala ayrılarak yapılır.

3.4. CHAID (Chi-Kare-Otomatik-Etkileşim Algılama)

Yetmişli yıllardan başlayarak, uygulamalı istatistik araştırmacıları karar ağaçları oluşturmak için yöntemler geliştirmişlerdir.[128] CHAID, bir popülasyonu, bağımlı değişkendeki varyasyonu gruplar içi minimum ve gruplar arası maksimum olacak şekilde farklı alt gruplara veya bölümlere tekrarlı olarak ayıran bir tekniktir.[129] Bu alt kümeler küçük tahmin edici alt gruplardan oluşur. En iyi tahmin sonucunu elde edebilmek için başlangıç değişkenleri bağımsız olarak yeniden kategorileştirilir. Adımsal olarak uygulanan benzer kategorileri birleştirme işlemi değişkenler arasında daha fazla birleştirme sağlanamayacağına istatistiksel olarak karar verilinceye kadar devam eder. Değişkenlerin bölünmeye uygun olup olmadığına Bonferroni düzeltilmiş *p* değeri kullanılarak karar verilir. Chi-kare-Otomatik-Etkileşim Algılama (CHIAD) ilk olarak sadece nominal değişkenleri işlemek için tasarlanmıştır. Her bir giriş özniteliği (bağımısz değiken) a_i için, CHAID hedef özelliğine (bağımlı değişken) göre en az önemli farkı olan V_i deki değer çiftlerini bulur. Anlamlı fark, istatistiksel bir testten elde edilen *p* değeriyle ölçülür. Kullanılan istatistiksel test, hedef özellik(bağımlı değişken) türüne göre değişir. Hedef özniteliği sürekli ise bir F testi kullanılır. Nominal ise bir Pearson ki-kare testi; ve ordinal ise bir olasılık oranı testi kullanılır. Seçilen her bir değer çifti için CHAID, elde edilen *p* değerinin belirli bir birleştirme eşiğinden daha büyük olup olmadığını kontrol eder. Cevap olumluysa, değerler birleştirir ve birleştirilecek ek bir potansiyel çifti aranır. Önemli (anlamlı) bir çift bulunana kadar işlem tekrarlanır.[130]

Bu süreç,

- Maksimum ağaç derinliğine ulaşıldığında,
- Ebeveyn (düğüm) için düğümdeki minimum vaka sayısına ulaşıldığında ve bu yüzden daha fazla bölünemediğinde,

[128] G. V. Kass, 'An Exploratory Technique for Investigating Large Quantities of Categorical Data', Applied Statistics, 29.2 (1980), s.119-127 <https://doi.org/10.2307/2986296>.
[129] Nurhan Doğan ve Kazım Özdamar, 'CHAID Analizi ve Aile Planlaması Le Lgili Bir Uygulama', T Klin Tıp Bilimleri 2003, 23.1 (2003), s.392–398.
[130] G. V. Kass, 'An Exploratory Technique for Investigating Large Quantities of Categorical Data', Applied Statistics, 29.2 (1980), s.119-127 <https://doi.org/10.2307/2986296>.

- Çocuk düğümü için düğümdeki minimum vaka sayısına ulaşıldığında,

yukarıdaki koşullardan biri yerine getirildiğinde de durur. CHAID, hepsini tek geçerli bir kategori olarak değerlendirerek eksik değerleri ele alır. CHAID analizinde budama yapılmaz.

CHAID Analizinin pratikte en çok tercih edilen ağaç diyagramı olmasının nedenleri arasında;[131]

- Geniş örneklemlerden yararlanma yeteneğinden dolayı potansiyel olarak çok güvenilir tahminler sunması,

- Bağımsız değişkenlerdeki kayıp gözlemleri tahmin edebilmesi, modelin gerçek yapısal formunda belirlenen varsayımları dikkate almadığı için ikili (binary) ve multi nominal lojistik regresyon modellerine alternatif bir parametrik olmayan ağaç diyagramı olarak kullanılabilmesi sayılabilir.

Genel olarak CHAID yönteminin algoritması şu şekildedir.[132]

Bağımlı değişken d ≥ 2 kategoriye, analizde kullanılan belirli bir bağımsız değişken de c ≥ 2 kategoriye sahip olsun. Analizdeki bir alt problem, bağımsız değişkenin uygun kategorileri birleştirilerek verilen *cxd* boyutlu olumsallık tablosunun en anlamlı *jxd* boyutlu tablo durumuna indirgenebilme problemi olsun. Kavramsal olarak ilk önce $T_i^{(j)}$istatistiği hesaplanır. Bu, *cxd* tablosu için $\left(j = 2,3,4,\ldots,c \right)$ bilinen χ^2 istatistiğidir. Eğer $T_j^{(*)} = \max T_i^{(j)}$ ise en iyi *jxd* tablosu için χ^2 değeri elde edilmiş demektir. Bu durumda $T_i^{(*)}$ en anlamlı olarak seçilir.

Algoritmanın tamamı şu şekildedir;[133]

[131]Nurhan Doğan ve Kazım Özdamar, 'CHAID Analizi ve Aile Planlaması Le Lgili Bir Uygulama', T Klin Tıp Bilimleri 2003, 23.1 (2003), s.392–398.
[132]G. V. Kass, 'An Exploratory Technique for Investigating Large Quantities of Categorical Data', Applied Statistics, 29.2 (1980), s.119-127 <https://doi.org/10.2307/2986296>.
[133] Nurhan Doğan ve Kazım Özdamar, 'CHAID Analizi ve Aile Planlaması Le Lgili Bir Uygulama', T Klin Tıp Bilimleri 2003, 23.1 (2003), s.392–398.

Adım 1. her bir bağımsız değişken için bağımlı değişkenin kategorileri ile bağımsız değişkenin kategorileri arasında çapraz tablo oluşturulur.

Adım 2. 2xd alt tablosunda bağımsız değişkene ait anlamlılığı en düşük olan kategori çiftleri bulunur. Birleşmeleri anlamlı bulunan iki kategori birleştirilir. Bu birleşme bileşik bir kategori olarak düşünülür ve bu adım bağımsız değişkenin kendi içindeki birleşmeleri anlamsız oluncaya kadar devam eder.

Adım 3. üç ya da daha çok sayıda orijinal kategori içeren bileşik kategorilerin her biri için birleşmenin tekrar çözümlendiği en önemli iki bölünme bulunur. Eğer anlamlılık bir kritik değerin altındaysa bölünme tamamlanarak ikinci adıma dönülür.

Adım 4. Optimum düzeyde birleştirilen bağımsız değişkenlerin her birinin anlamlılığı hesaplanır, en çok anlamlı olan ayrılır. Eğer bu anlamlılık kritik bir değerden büyükse seçilen bağımsız değişkenin birleştirilen kategorilerine göre veri alt gruplara bölünür.

Adım 5. Henüz analiz edilmemiş veri için birinci adıma gidilir. Her bir bağımsız değişken için, kendi içinde kategorileri en anlamlı bir şekilde birleştirilip en iyi bölünme bulunduktan sonra, bağımlı değişkene göre olumsallık tablosu oluşturulur. Daha sonra χ^2 ve Bonferroni p değeri hesaplanır. Bağımsız değişkenler birbiri ile karşılaştırılıp en küçük p değerine sahip olan bağımsız değişkenin kategorilerine göre veriler alt gruplara ayrılır.

Genel olarak bir CHAID analizi yapmak için büyük bir örnek büyüklüğüne ihtiyaç vardır. Her bir dalda, toplam popülasyon bölüşüldüğünde, mevcut gözlemlerin sayısı azaltılır ve toplam örneklem büyüklüğü ile bireysel gruplar güvenilir analiz için çok küçük olabilirler.[134]

3.5. Ayrıntılı CHAID

Ayrıntılı CHAID, *Biggs*, de *Ville* ve *Suen* (1991) tarafından, orijinal CHAID yönteminin bazı zayıflıklarını telafi etmek için geliştirilmiştir. Ayrıntılı CHAID,

[134] Sarah Littler, CHAID, (https://select-statistics.co.uk/blog/chaid-chi-square-automatic-interaction-detector, 05.08.2018'de erişildi.)

CHAID'in değiştirilmiş bir halidir ve her bir tahmin değişkeni için mümkün olan tüm dağılımları daha ayrıntılı bir şekilde incelemektedir. Bu sebeple hesaplama açısından zaman almaktadır. Hedef değişken (bağımlı değişken) nominal, ordinal ya da sürekli olabilir. Uygulamada genelde, müşteri gruplarını seçmek ve bazı değişkenleri etkileyen diğer değişkenlere nasıl yanıt verdiklerini tahmin etmek için doğrudan pazarlamanın konusu olarak kullanılmaktadır.[135]

3.6. QUEST (Hızlı Objektif Etkili İstatistik Ağacı)

QUEST (Hızlı Objektif Etkili İstatistik Ağacı) diğer yöntemlerin yavaşlıklarını hızlı bir biçimde hesaplayan ve pek çok kategorisi bulunan tahmin değişkenlerinin lehine bunlardan kaçınılmasını sağlayan bir yöntemdir.[136] Wei-yin Loh ve Yu-Shan Shih tarafında geliştirilmiştir. En önemli özelliği bağımlı değişkenin nominal olması gerektiğidir.[137] Hızlı, Objektif, Etkili İstatistik Ağacı (QUEST) algoritması, tek değişkenli ve lineer kombinasyon ayırmalarını destekler.[138] Her bölünme için, her bir giriş özniteliği (bağımsız değişkeni) ile hedef özniteliği (bağımlı değişken) arasındaki ilişki ANOVA F-testi veya Levene testi (ordinal ve sürekli öznitelikler için) veya Pearson'ın ki-karesi (nominal öznitelikler) kullanılarak hesaplanır. Her bir özellik için bir ANOVA F istatistiği hesaplanır.

En büyük F istatistiği önceden tanımlanmış bir eşik değerini aşarsa, en büyük F değerine sahip öznitelik (değişken), düğümü bölmek için seçilir. Aksi halde, Levene'in eşit olmayan varyans testi her özellik için hesaplanır. En büyük Levene istatistik değeri, önceden tanımlanmış bir eşik değerinden daha büyükse, en büyük Levene değerine sahip özellik, düğümü bölmek için kullanılır. Hiçbir özellik eşiği aşmadıysa, düğüm en büyük ANOVA F değeriyle öznitelik kullanılarak ayrılır.

Hedef özniteliği (bağımsız değişken) çok terimli ise, iki sınıf oluşturmak için iki yönlü kümeleme kullanılır. Hedef özniteliğiyle(bağımlı değişkenle) en yüksek ilişkilendirmeyi elde eden öznitelik(bağımsız değişken) bölünme için seçilir. Bağımsız

[135] Gülpınar, s.99
[136] Wei-Yin Loh and Yu-Shan Shih, 'Split Selection Methods for Classification Trees', Statistica Sinica, 7.4 (1997), s.815–840, <https://doi.org/10.2307/24306157>.
[137] Gülpınar, s.100
[138] Loh and Shih, s.815–840

değişkenlerin en uygun ayırma noktasını bulmak için kuadratik diskriminant Analizi (QDA) uygulanır. QUEST ihmal edilebilir bir önyargıya sahiptir ve ikili bir karar ağacı vermektedir. Ağaçları budamak için on kat çapraz doğrulama kullanılır.[139]

CART gibi detaylı araştırma yöntemleri, ağaç oluşturma sürecinde daha çok dağılımlar meydana getirebilen daha ayrı değişkenleri seçme eğilimindedirler. CART'ın getirmiş olduğu bir başka sınırlama da dağılımları araştırırken hesaplamaya çok fazla yatırım yapılıyor olmasıdır.[140] QUEST yöntemi işte bu sorunlara çözüm getirmek amacı ile tasarlanmıştır. QUEST'in, değişken seçimi önyargısı ve hesaplama maliyeti açısından detaylı araştırma yöntemlerinden çok daha uygun olduğu gözlemlenmiştir. Ancak sınıflandırma doğruluğu, dağılım noktalarının değişkenliği ve ağaç boyutu açısından değiştirilmemiş dağılımlar kullanıldığında halen net bir şekilde bir kazanan ortaya çıkamamıştır.

3.7. Karar ağaçlarının Avantajları ve Dezavantajları

İstatiksel bir öğrenme yöntemi olan karar ağacının birçok avantaj literatürde yer alır:[141]

1) Karar ağaçları insan karar mekanizmasına benzemektedirler.

2) Ağaçlar grafiksel olarak gösterilebilir ve uzman olmayanlar tarafından bile daha kolay yorumlanabilir. (özellikle küçük ağaçlar). Karar ağaçları kendiliğinden açıklayıcıdır ve sıkıştırıldığında da takip edilmesi kolaydır. Yani, karar ağacının makul sayıda yaprağı varsa, profesyonel olmayan kullanıcılar tarafından kavranabilir. Dahası, karar ağaçları bir dizi kurala dönüştürülebildiğinden, bu tür temsiller anlaşılır olarak kabul edilir

3) Ağaçlar kukla değişkenlere ihtiyaç duymadan nitel değişkenleri kolayca işleyebilir. Karar ağaçları hem nominal hem de sayısal giriş özelliklerini işleyebilir.

---

[139] Rokach ve Maimon, s.79
[140] Beril Sipahi, 'Data Mining İn Customer Realtionship Management: Model Building And Application in Automotive industry,' İstanbul, (Doktora tezi, 2002), s.63
[141] Rokach ve Maimon, s.81

4) Karar ağacı temsili, herhangi bir ayrık değer sınıflandırıcısını temsil edecek kadar zengindir.

5) Karar ağaçları, hataları olabilecek veri kümelerini işleyebilir.

6) Karar ağaçları eksik değerlere sahip olabilecek tanıtıcı veri kümeleriyle çalışabilir.

7) Karar ağaçları, parametrik olmayan bir yöntem olarak kabul edilir. Karar ağaçları, dağılımla ilgili ve sınıflandırıcı yapısına ilişkin herhangi bir varsayımı içermez.

8) Sınıflandırma maliyeti yüksek olduğunda, karar ağaçları, yalnızca değişken değerlerini kökten yaprağa tek bir yol boyunca sormalarından dolayı daha caziptirler.

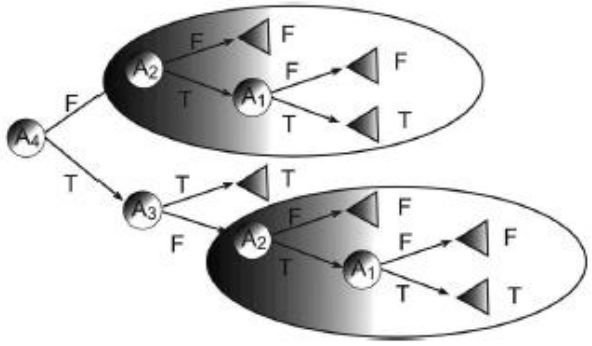
Karar ağaçlarının dezavantajları şunlardır:[142]

1) Algoritmaların çoğu (ID3 ve C4.5 gibi), hedef özelliğinin yalnızca ayrık değerlere sahip olmasını gerektirir.

2) Karar ağaçları "böl ve yönet" yöntemini kullanarak, oldukça alakalı birkaç özellik varsa, iyi performans göstermeye eğilimlidirler, ancak çok karmaşık etkileşimler mevcutsa daha az performans gösterirler. Bunun olmasının nedenlerinden biri, diğer sınıflandırıcıların bir karar ağacını kullanarak göstermeyi zorlayacak bir sınıflandırıcıyı kompakt bir şekilde tanımlayabilmeleridir. Bu olgunun basit bir örneği karar ağaçlarının replikasyon (tekrarlama) problemidir.[143] Çoğu karar ağacı, örnek uzayı bir kavramı temsil etmek için karşılıklı olarak özel bölgelere böldüğünden, bazı durumlarda ağacın sınıflandırıcıyı temsil etmek için aynı alt ağacın birkaç kopyasını içermesi gerekir. Replikasyon problemi alt ağaçların ayrık kavramlara çoğalmasını zorlaştırır. Örneğin, konsept (kavram) aşağıdaki ikili fonksiyonu izliyorsa:

[142] Rokach ve Maimon, s.82
[143] Giulia Pagallo and David Haussler, 'Boolean Feature Discovery in Empirical Learning', *Machine Learning*, 5.1 (1990), s.71–99 , <https://doi.org/10.1023/A:1022611825350>.

$y = (A_1 \cap A_2) \cup (A_3 \cap A_4)$, bu fonksiyonu temsil eden minimal tek değişkenli karar ağacı Şekil 4.2'de gösterilmiştir. Ağacın, aynı alt ağacın iki kopyasını içerdiği görülmektedir.



Şekil 4.2. Karar ağacında replikasyon (tekrarlanma) durumu[144]

3) Karar ağaçlarının açgözlü özelliği, dikkat çekilmesi gereken bir başka dezavantaja yol açmaktadır. Eğitim setine aşırı duyarlılık, ilgisiz nitelikler karar ağaçları özellikle kararsız hale getirir: Köküne yakın bir bölünmede küçük bir değişiklik, aşağıdaki tüm alt ağacı değiştirecektir. Eğitim setindeki küçük değişiklikler nedeniyle, algoritma gerçekten en iyi olmayan bir özellik seçebilir. Yani ağaçlar çok robust (dayanıklı) değillerdir. Verideki küçük bir değişim son tahmin edilen ağaçta büyük değişimlere neden olabilir.[145]

4) Bir başka problem, eksik değerlerle başa çıkmak için gereken çabayı göstermektedir.[146]Eksik değerlerle başa çıkabilme kabiliyeti avantaj olarak değerlendirilirken, bunu başarmak için gereken aşırı çaba bir dezavantaj (engel) olarak kabul edilir. Test edilen bir özellik eksikse doğru dallanma bilinmemektedir ve algoritma eksik değerleri işlemek için özel mekanizmalar kullanmalıdır. Eksik değerler üzerindeki testlerin oluşumunu azaltmak için, C4.5, bilgi kazancını bilinmeyen örneklerin oranına göre cezalandırır ve sonra

[144] Rokach ve Maimon, s.82
[145] James ve diğerleri, s.316
[146] Jerome H Friedman, 'Lazy Decision Trees', 34.2 (1997), s.167–180.

bu örnekleri alt ağaçlara ayırır. CART, çok daha karmaşık özellikli bir özellik şeması kullanır.

5) Karar ağacı indüksiyon algoritmalarının çoğunun tam anlamıyla ileriyi göremeyen doğası, indükleyicilerin sadece bir seviye daha ileriye bakmasıyla yansıtılmaktadır. Spesifik olarak ayırma kriteri onların dolaysız oğullarına dayanarak mümkün olan özellikleri sıraya koyar. Böyle bir strateji, ayırmada yüksek puan alan testleri tercih eder ve nitelik kombinasyonlarını gözden kaçırabilir. Daha derin gözetleme stratejileri kullanmak, hesaplama açısından pahalıdır ve yararlı olduğu kanıtlanmamıştır.

6) Ağaç, ayrılan bölünmelerle genişledikçe, yapılan sınıflandırma sonucunda düğümlerde daha az bilgiye sahip olacaktır. Karar ağaçları veriyi çok fazla sayıda gruba böler. Bu gruplar daha özel bir hal aldıkça küçülmeye başlar. İncelenmesi gereken farklı durum sayıları attıkça, eğitim kümelerinin her biri daha da küçülür. Rakamlardaki azalış nedeniyle sınıflandırmanın doğru bir biçimde yapılmasında daha az güvenilirlik söz konusu olur.[147]

7) Karar ağaçları dinamik bir veriyi kolayca ele alamazlar. Bu nedenle değişkenin etki kümeleri kolayca incelenecek şekilde kategorilere bölünmelidir

[147] Gülpınar, s.63,64

4.BÖLÜM

KARAR ORMANLARI

(TOPLULUK ÖĞRENME ALGORİTMALARI)

Bir topluluk öğrenme yönteminin ana fikri, her biri aynı problemi çözen, daha doğru ve güvenilir tahminler veya kararlar ile tek bir model kullanarak elde edilenden daha iyi birleşik (küresel) bir model elde etmektir. Çoklu modelleri birleştirerek tahmin modelini oluşturma fikri uzun süredir araştırılmaktadır. Aslında, topluluk metodolojisi, önemli bir karar vermeden önce birkaç görüş aramak için insan doğasını taklit eder. Bireysel görüşleri tartar ve nihai karara ulaşmak için bunları birleştirir.[148]

Topluluk yöntemlerinin tahmin performansının iyileştirilmesinde kullanılabileceği bilinmektedir. İstatistik, makine öğrenimi, örüntü tanıma ve veri madenciliği gibi çeşitli disiplinlerden araştırmacılar, topluluk yönteminin kullanımını dikkate almışlardır.

Literatürde, "topluluk yöntemleri" terimi genellikle aynı temel modelin küçük değişikliklerine sahip model koleksiyonları olarak yer almaktadır. Ayrıca, literatürde "çoklu sınıflayıcı sistemler" olarak da bilinmektedir.[149]

Bu bölümde Bagging (Torbolama), Boosting (Hızlandırma) ve Random Forest (Karar Ormanları) yöntemlerinden bahsedilecektir. Bu yöntemler daha güçlü tahminler oluşturmak için yapı taşları olarak karar ağaçlarını kullanırlar.

4.1. Bağımlı Yöntemler

Öğrenme toplulukları için bağımlı yaklaşımlarda, öğrenme çalışmaları arasında bir etkileşim vardır. Böylece, sonraki iterasyonlarda öğrenmeyi yönlendirmek için

[148] Robi Polikar, 'Ensemble Based Systems in Decision Making', IEEE Circuits and Systems Magazine, 6.3 (2006), s.21–44, <https://doi.org/10.1109/MCAS.2006.1688199>.
[149] Rokach ve Maimon, s.99

önceki iterasyonlarda üretilen bilgiden faydalanılır. Aşağıdaki bölümlerde açıklandığı gibi, bağımlı öğrenme için iki ana yaklaşımdan bahsederiz.[150]

4.1.1. Hızlandırma (Boosting)

Boosting (aynı zamanda adaptif yeniden örnekleme ve birleştirme olarak da bilinir) zayıf bir öğrenicinin (sınıflandırma kuralları veya karar ağaçları gibi) performansını iyileştirmek için kullanılan bir yöntem gibi düşünülebilir. Yöntem, çeşitli dağıtılmış eğitim verilerinin üzerinde sürekli olarak zayıf bir öğrenciyi (sınıflandırma kuralları veya karar ağaçları gibi) geliştirerek çalışır. Zayıf öğrenenler tarafından üretilen sınıflandırıcılar daha sonra, zayıf öğrenen sınıflandırıcılarının sahip olabileceğinden daha yüksek bir doğruluk elde etmek için tek bir bileşik güçlü sınıflandırıcıyla birleştirilir.[151]

Boosting regresyon veya sınıflandırma için birçok istatistiksel öğrenme yöntemine uygulanabilecek genel bir yaklaşımdır. Boosting'de oluşturulan ağaçlar sırayla yetiştirilir: Her ağaç daha önce yetiştirilen ağaçlardan bilgi kullanılarak yetiştirilir. Boosting'de oluşturulan her ağaç orijinal veri kümesinin değiştirilmiş bir sürümüne uymaktadır.

Hızlandırma yönteminin temel amacı hata varyansını azaltmaktır. Varyansı azaltmak için aynı eğitim veri setini farklı örneklerle yani her bir iterasyon sonucu belirlenen ağırlıklandırılmış örneklerle eğiterek farklı sınıflandırıcılar üretir. Bu yöntem varyansı önemli ölçüde düşürür.[152]

Hızlandırma algoritması temel algoritma olarak J48 karar sınıflandırıcısıyla kullanılır.[153] Bu yöntem, her gözlem için bir ağırlık kabul eder. Ağırlık ne kadar yüksek

[150] Provost and Fawcett, 'Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions',(1997),s.43-48
[151] Rokach ve Maimon, s.118
[152] Yoav Freund and Robert E. Schapire, 'Experiments With A New Boosting Algorithm', ICML '96: Proceedings of the 13th International Conference on Machine Learning, 1996, s.148–156.
[153] Pal, Mahesh, Random Forest For Land Cover Classification, Geoscience and Remote Sensing Symposium, 2003. IGARSS '03. Proceedings. 2003 IEEE International, 6, 3510 – 3512. Aktaran Özlem Akar Doktora tezi.

olursa o gözlem sınıflandırıcıyı o kadar çok etkiler. Boosting yöntemi üç adet sınıflandırıcı üretir. İşlem adımları;[154]

1) Orijinal veri setinden iadesiz olarak S_1 verisi oluşturur. S_1 verileri eğitilir ve M_1 sınıflandırıcısı oluşturulur.

2) İkinci sınıflandırıcı için S_2 verisi oluşturulur. Bu verilerin yarısı M_1 tarafından yanlış sınıflandırılan gözlemler, diğer yarısı ise M_1 tarafından doğru sınıflandırılan gözlemlerden oluşmaktadır. Buna göre oluşturulan S_2 verisi ile M_2 sınıflandırıcısı oluşturulur.

3) Üçüncü sınıflandırıcı olarak, M_1 ve M_2 sınıflandırıcılarında farklı olarak sınıflandırılan gözlemler ile S_3 verisi oluşturur. S_3 verileri eğitilerek M_3 sınıflandırıcısı oluşturulur. M_1, M_2, M_3 sınıflandırıcılarından en çok oyu alan sınıfa atama yapılır.[155]

Özetleyecek olursak; S, eğitim veri seti; m, toplam örnek sayısı; k, sınıflandırıcı için örnek sayısı; $(k < m)$ üzere;[156]

1) S_1, S'den iadesiz olarak rastgele seçilen gözlemler

2) M_1, S_1 'den oluşturulan sınıflandırıcı

3) S_2, S ve S_1'den iadesiz olarak rastgele seçilen örnekler (yarısı M_1 tarafından yanlış sınıflandırılmış örnekler, diğer yarısı ise M_1 tarafından doğru sınıflandırılmış örnekler)

4) M_2 S_2'den oluşturulan sınıflandırıcı

5) S_3 ise S, S_1 ve S_2'den iadesiz olarak rastgele seçilen örnekler (M_1 ve M_2 sınıflandırıcılarında farklı olarak sınıflandırılan örnekler)

[154] Özlem Akar, 'Rastgele Orman Sınıflandırıcısına Doku Özellikleri Entegre Edilerek Benzer Spektral Özellikteki Tarımsal Ürünlerin Sınıflandırılması, Trabzon (Doktora Tezi , 2013)', s.35-36
[155] Lior Rokach, Pattern Classıfıcatıon Using Ensemble Methods(Serıes In Machine Perceptıon And Artificial Intelligence-Vol. 75),Singapore: World Scientific Publishing Company, 2010, s.28
[156] Rokach, s.29

6) M_1, M_2, M_3 sınıflandırıcılarından en çok oyu alan sınıfına atama yapılır.

Hızlandırmanın amacı, veriler üzerinde iyi performans gösteren bir bileşik sınıflandırıcı oluşturmaktır, ancak çok sayıda iterasyon, tek bir sınıflandırıcıdan önemli ölçüde daha az doğru olan çok karmaşık bir bileşik sınıflandırıcı oluşturabilir. Aşırı öğrenmeden kaçınmanın olası bir yolu, yineleme sayısını mümkün olduğunca küçük tutmaktır. Hızlandırmanın bir başka önemli dezavantajı, anlaşılması zor olmasıdır. Ortaya çıkan topluluk, kullanıcının tek bir sınıflandırıcı yerine birkaç sınıflandırıcıyı yakalaması gerektiğinden daha az anlaşılabilir olduğu düşünülmektedir. Yukarıdaki dezavantajlara rağmen, Breiman (1996), "doksanlı yılların" sınıflandırıcı tasarımında en önemli gelişme olarak boosting yöntemi olduğunu ifade etmiştir.[157]

4.1.1. Artırımlı Toplu Öğrenme

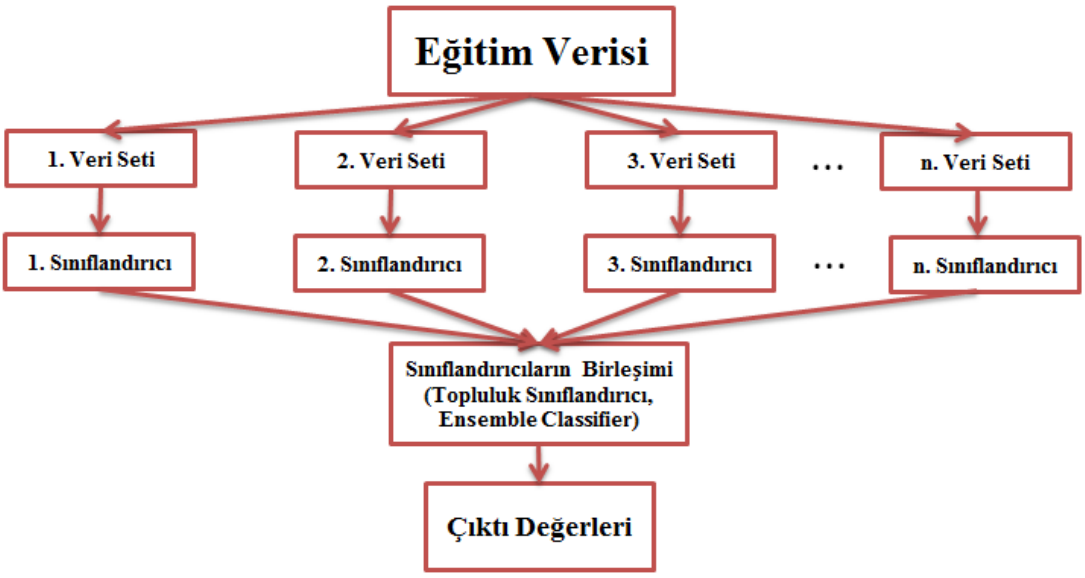
Bu yöntemde, bir iterasyonda üretilen sınıflandırma, bir sonraki iterasyonda öğrenme algoritmasına ön bilgi olarak verilir. Öğrenme algoritması, bir sonraki sınıflandırıcıyı oluşturmak için eski sınıflandırıcının sınıflandırmasıyla birlikte mevcut eğitim setini kullanır. Son iterasyonda oluşturulan sınıflandırıcı son sınıflandırıcı olarak seçilir.[158]

4.2. Bağımsız Yöntemler

Bağımsız topluluk yöntemlerinde, orijinal veri kümesi çoklu sınıflandırıcıların uygulandığı birkaç alt kümeye bölünmüştür.(Şekil 4.1). Orijinal eğitim setinden oluşturulan alt gruplar, ayrık (birbirini dışlayan) veya kesişik olabilir. Belirli bir örnek için tek bir sınıflandırma üretmek için bir birleştirme süreci uygulanır. İndüklenen sınıflandırıcıların sonuçlarını birleştirme yöntemi genellikle indüksiyon algoritmalarından bağımsız olduğu için, her bir alt sette farklı indüktörlerle kullanılabilir. Dahası, bu metodoloji kolaylıkla paralel hale getirilebilir. Bu bağımsız

[157] Rokach ve Maimon, s.122
[158] Rokach ve Maimon, s.122

yöntemler, sınıflandırıcıların tahmini gücünü artırma ya da toplam hesaplama süresini azaltmayı amaçlamaktadır.[159]



Şekil 4.1. Bağımsız Sınıflandırıcı Çalışma Prensibi

4.2.1. Torbalama (Bagging)

1996 yılında Breiman tarafından geliştirilen yöntem olan Bootsrap toplaması (bootstrap aggregating) veya bagging, istatiksel öğrenme metodunun hata varyansını azaltmak için amaçlanan genel bir yöntemdir. Torbalama tahmini, bir tahmincinin birden çok sürümünü oluşturur ve bunları toplu bir tahmin elde etmek için kullanan bir yöntemdir. Torbalama, sayısal bir sonuç tahmin ederken tahminlerin ortalamasını alır ve bir sınıf tahmin ederken oylamaya göre tahmin yapar.[160] Yöntem, öğrenilmiş sınıflandırıcıların çeşitli çıktılarını tek bir tahmin halinde birleştirerek, geliştirilmiş bir bileşik sınıflandırıcı oluşturarak doğruluğu arttırmayı amaçlamaktadır. Yeni bir örneği sınıflandırmak için, her bir sınıflandırıcı bilinmeyen örnek için sınıf tahminini oluşturur. Sonuç olarak, torbalama, orijinal tekli verilerden oluşturulan tek modelden daha iyi performans gösteren birleşik bir model üretmektedir. Breiman (1996), özellikle kararsız indükleyiciler için bunun doğru olduğunu belirtmektedir, çünkü torbalama

[159] Rokach ve Maimon, s.123
[160] Leo Breiman, 'Bagging Predictors', Machine Learning, 24, (1996), s.123–140.

kararsızlıklarını ortadan kaldırabilir. Bu bağlamda, öğrenme setini bozmak, yapılandırılmış sınıflandırıcıda önemli değişikliklere neden olabiliyorsa, bir indükleyici kararsız olarak kabul edilir. Bu yöntem karar ağaçları için yararlı ve sıklıkla kullanılır.[161]

Breiman makalesinde bu yöntemi şöyle açıklamaktadır.[162] $\varphi(x, L)$ gibi tek sınıflandırıcı kullanmak yerine $\varphi(x, L_k)$ gibi birden çok sınıflandırıcıyı kullanarak $\varphi(x, L)'$ den daha iyi şekilde *y*'yi tahmin etmektir. L_k, L'den üretilen bootsrap örneklemlerini ifade etmektedir.[163]Verilen bir $L = \{(y_n, x_n), n = 1, \ldots N\}$ veri seti için, burada *y* bağımlı değişkeni (nitel veya nicel), *x* girdi(bağımsız değişken) veri setini *N* gözlem sayısını göstermektedir. $\varphi_A(x) = E_L \varphi(x, L)$ formüldeki E_L, L üzerinden olasılıkları ifade etmektedir. φ_A 'daki *A* da aggregating (birleştirme) olarak ifade edilir. $\varphi(x, L)$, $j \in \{1, \ldots, J\}$ sınıfları için tahminlerde bulunur. $\varphi(x, L_k)$ $N_j \neq \{k; \varphi(x, L_k) = j\}$ her bir L_k için ağaç oluşturulur ve her sınıf için yapılan tahminler toplanarak oylanır. Oylama işlemi, aggregating (birleştirme) metotlarından biridir. $\varphi_A(x) = \arg\max_j N_j$ maksimumdur. Bu formüle göre *x* verisi yoğun olarak j sınıfına atanmış ise bu sınıf toplamlarının sayısı da N_j'dir. Özetle, formülle x in atandığı sınıflar N_j toplanır (aggregating). Daha sonra tahminlere bakılır *x* 'in atandığı sınıflardan x, yoğun olarak hangi sınıfa atandıysa onun oyu çok olur. Bu durumda x, en çok oyu alan sınıf hangisiyse o sınıfa atanır.

Torbalama algoritmasını kısaca özetlemek gerekirse; Bir bootstrap örneği ile iadeli olarak eğitim setinden eşit oranda *m* tane örnek içeren örneklemler üretilir. *T* boostsrap örneklemleri B_1, B_2, ... , B_T üretilir ve her bir bootstrap örneği B_i bir C_i sınıflandırıcısı oluşturulur. Son sınıflandırıcı olan C^*, C_1,C_2, ...,C_T sınıflandırıcılarının en çok tahmin ettiği sınıfı baz alarak elde edilen sınıflandırıcıdır.[164]

---

[161] Rokach ve Maimon, s.122
[162] Leo Breiman, 'Bagging Predictors', Machine Learning, 24, (1996), s.123–140.
[163] Akar, s.27-28.
[164] Eric Bauer and Ron Kohavi, 'An Empirical Comparison of Voting Classification Algorithms : Bagging , Boosting , and Variants', Machine Learning, 36 (1999), s.105–139.

Torbalama, hızlandırma gibi, farklı sınıflandırıcılar üreterek ve çoklu modelleri birleştirip bir sınıflandırıcının doğruluğunu geliştirilen bir tekniktir. Her ikisi de aynı tipteki farklı sınıflandırıcıların çıktılarını birleştirerek sınıflandırma için bir çeşit oy kullanırlar. Hızlandırma, torbalamadan farklı olarak, her bir sınıflandırıcıda daha önce yapılmış olan hatalara daha fazla dikkat etmeye çalışan yeni sınıflandırıcı, daha önce yapılmış olanların performansından etkilenir. Torbalamada, her bir örnek eşit olasılıkla seçilir, hızlandırmada ise, örnekler, ağırlıkları ile orantılı bir olasılıkla seçilir.[165]

4.2.1.1 Torba Dışı Hata Tahmini (Out-of-Bag Error Estimation) (OOB)

Torbalanmış bir modelin test hatasını tahmin etmek için çapraz doğrulama veya doğrulama seti yaklaşımına gerek kalmadan çok basit bir yol torba dışı (OOB) yöntemidir. Her bir bagged ağaç ortalama olarak gözlemlerin üçte-ikisini kullanır. Geriye kalan gözlemlerin üçte biri belirli bir bagged ağaçta kullanılmaz torba dışı (OOB) gözlemler olarak adlandırılır. Gözlemin OOB olduğu ağaçların her birini kullanarak *i*'inci gözlem için alacağı değer tahmin edilebilir. *i*'inci gözlemin tek bir tahmin değerini elde etmek için bu tahmini değerler ortalanır (eğer bağımlı değişken nicelse), veya çoğunluk oyu alınabilir.(eğer bağımlı değişken nitelse). Bu *i*. gözlem için tek bir OOB tahmini ortaya çıkar. Her bir *n* gözlem için bu yolla bir OOB tahmini elde edilebilir, tüm OOB, *MSE* veya sınıflandırma hatası (sınıflandırma için) hesaplanabilir. Elde edilen OOB hatası bagged model için test hatasının geçerli bir tahminidir, çünkü her gözlemin tahmini o gözlemin kullanılmadığı ağaçlar yardımıyla tahmin edilir. Test hatasının tahmin edilmesi için OOB yaklaşımı çapraz doğrulamanın hesaplama açısından zorlayıcı olduğu büyük veri setlerinde bagging yapılırken uygundur.[166]

4.2.1.2 Değişken Önemlilik Ölçümleri (Variable Importance Measures)

Bagging tipik olarak tek bir ağaç kullanarak tahmin etmede daha fazla doğrulukla sonuçlanır. Ancak sonuçta ortaya çıkan modeli yorumlamak zor olabilir. Karar ağaçlarının avantajı cazip ve kolay yorumlanabilir diyagramıdır. Ne yazık ki çok sayıda ağaç torbalandığında (bagging), tek bir ağacı kullanarak ortaya çıkan istatiksel öğrenme prosedürünü temsil etmek mümkün değildir ve artık hangi değişkenin yöntem

[165] Rokach ve Maimon, s.124
[166] James ve diğerleri, s.318

için en önemli olduğu net değildir. Bagging yorumlanabilirlik pahasına tahmin doğruluğunu artırır.

Bagging regresyon ağaçlarında, ortalanmış tüm *B* ağaçları üzerinde verilen bir değişken (tahminci) üzerinde bölünmeler sayesinde RSS'nin azaldığı toplam miktar bulunabilir. Büyük bir değer önemli bir değişkeni gösterir. Benzer şekilde Bagging sınıflandırma ağaçlarında, tüm *B* ağaçları üzerinde Gini indeksi, verilen belirli bir tahmin edici üzerindeki bölünmeler tarafından toplam miktar bulunabilir.[167] Bu yöntemde, *m*. değişken için dallara ayırma gerçekleşmeden önce gini değeri hesaplanır. Sonra *m*. değişken alt dallara ayrıldığında bölünen veri için gini değeri tekrar hesaplanır. Bölünmeden önceki verinin gini değeriyle, bölünmeden sonraki verinin gini değeri arasındaki fark alınır. Ormanda *m*. değişkeni kullanılarak oluşan her ağaç için bölünmeden önceki gini değeri ile bölünmeden sonraki gini değeri arasındaki fark bulunur ve tüm ağaçlar oluştuktan sonra aradaki farklar toplanır. Bulunan değer *m*. değişkenin gini önem derecesini verir. Bu işlemler tüm değişkenler için hesaplanır.[168]

Bir diğer yöntem ise OOB hatasını kullanarak yapılır. *m*. değişkenin önem derecesi için, karar ağacı oluşturulduktan sonra, OOB test verisi ağaca yerleştirilir ve doğru sınıflama sayısı yazılır. Daha sonra, OOB test verisindeki *m*. değişkenin değerleri kendi içinde karıştırılır yani tüm değerlerin yeri değiştirilir. Değiştirilmiş OOB test verisi önceden oluşturulan karar ağacına yerleştirilir ve doğru sınıflama sayısı yazılır.[169] Her ağaçta ve her değişken için bu işlem yapılır ve OOB hataları yazılır. Değerler karıştırılmadan hesaplanan OOB hataları ile değişkenin değerlerinin karıştırılması sonucu elde edilen OOB hatası karşılaştırıldığında hata oranı artmış ise *m*. değişkenin sınıflandırmayı önemli ölçüde etkilediği söylenir.[170]

[167] James ve diğerleri, s.319
[168] Muhammet Akman, Veri Madenciliğine Genel bakış ve Random Forest Yönteminin İncelenmesi: Sağlık Alanında Bir Uygulama, Ankara(Yüksek Lisans Tezi,2010), s.37
[169] Akman s.37
[170] Akar, s.32

4.2.2 Rastgele Orman (Random Forest)

Rastgele orman (Random Forest) 2001 yılında Leo Breiman tarafından geliştirilmiştir. Rastgele orman, bagging yöntemi ve Ho tarafından önerilen rastgele alt uzay yöntemlerinin birleşiminden oluşmaktadır. Rastgele alt uzay yönteminde en uygun dallara bölünmeyi sağlayacak değişken tüm değişkenler arasından rastgele seçilmiş az sayıda değişken tarafından belirlenir.[171] Rastgele ormanlar, ağaç tipi sınıflandırıcılar topluluğudur, öyle ki, her ağaç bağımsız olarak örneklenen bir rasgele vektörün değerlerine ve ormandaki tüm ağaçlar aynı dağılıma bağlıdır.[172] Torbalama yönteminin gelişmiş bir şekli olarak kabul edilebilir. Bu yöntemi Breiman makalesin şöyle açıklamaktadır. *k*.'ıncı ağaç için, geçmiş rastgele vektörlerden Q_1, ..., ,Q_{k-1} bağımsız ancak aynı dağılımla sahip Q_k rasgele vektörü oluşturulur; ve bir ağaç , *x*'in giriş vektörü olduğu $h(x,Q_k)$ sınıflandırıcısının sonucu olarak eğitim veri seti ve Q_k kullanılarak oluşturulur.[173] Örneğin, bagging'de, rasgele vektör *Q*, *N* eğitim veri setindeki gözlem sayısı olmak üzere, kutularda rastgele atılan *N* gözlemlerinden elde edilen *N* kutudaki sayımlar olarak oluşturulur. Rastgele bölünme seçiminde *Q*, *1* ile *k* arasında bir dizi bağımsız rastgele tam sayıdan oluşur. *Q*'nun yapısı ve boyutsallığı, ağaç yapımındaki kullanımına bağlıdır.[174]

Rastgele bir orman, ağaç yapılı sınıflandırıcılar $\{h(x,Q_k), k=1,...\}$ şeklindeki sınıflandırıcılarının toplanmasından oluşan bir sınıflandırıcıdır. Burada Q_k, bağımsız özdeş dağılan rastgele bir vektördür.[175]

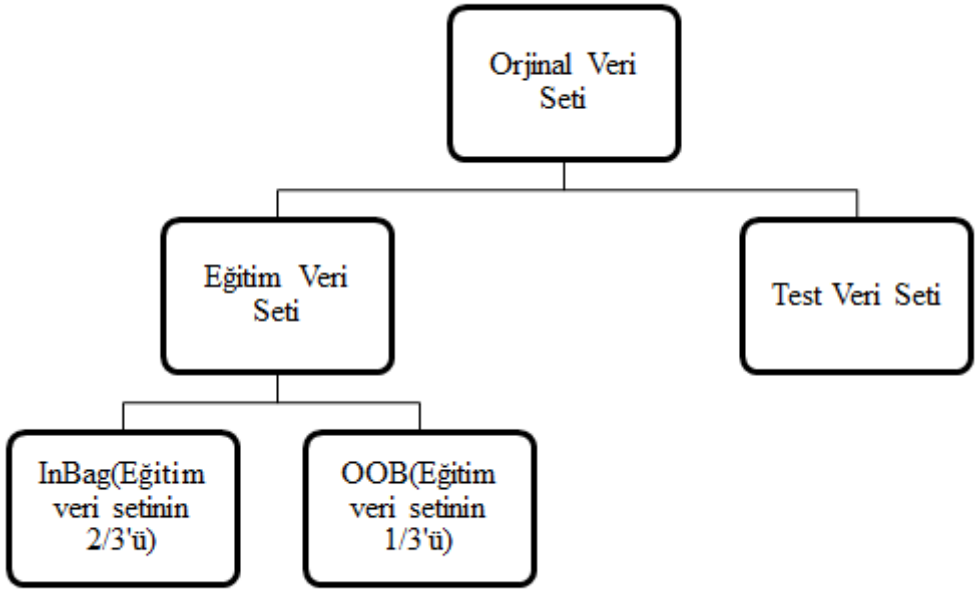
Rastgele ormanlarda, torbalama (bagging) rastgele değişken (özellik) seçimi ile birlikte kullanılır. Her yeni eğitim seti, orijinal eğitim setinden iadeli olarak (bootsrap yöntemiyle), çekilir. Ardından rastgele değişken (özellik) seçimi kullanılarak yeni eğitim setinde bir ağaç yetiştirilir. Yetiştirilen ağaçlarda budama yapılmaz.[176] Yapılan

[171] Betül Uzbaş, Sayısal Dental Modellerden Otomatik Cinsiyet Tespiti, Konya (Doktora Tezi,2017),s.49
[172] Leo Breiman, 'Random Forests - Random Features, Tecnical Report 567, Statistic Department, University of California, Berkeley, (Https://Www.Stat.Berkeley.Edu/~breiman/Random-Forests.Pdf, 08.10.2018'de Erişildi)', 1999, s.1–29.
[173] Leo Breiman, 'Random Forests', Machine Learning, 45.1 (2001), s.5–32 <https://doi.org/10.1023/A:1010933404324>.
[174] Breiman, 'Random Forests, s.5-32
[175] Breiman, 'Random Forests'. s.5-32
[176] Breiman, 'Random Forests', s.5–32

çalışmalar, özellik seçimi ölçütlerinin değil, budama yöntemlerinin seçiminin ağaç tabanlı sınıflandırıcıların performansını etkilediğini göstermektedir.[177] Torbalamanın kullanılmamasının iki nedeni vardır. Birincisi, rastgele değişkenler (özellikler) kullanıldığında torbalama kullanımının doğruluğu arttırdığı görülmektedir. İkincisi, genelleştirilmiş hataların torba dışı (Out-of-bag (OOB)) hesaplanmasıdır.[178] Budamanın olmaması rastgele ormanları diğer karar ağacı yöntemlerinden daha avantajlı hale getirmektedir. Bagging ve rastegele ormanlar arasındaki temel fark *m* boyutlu değişkenlerin seçimidir. Örneğin, m=p olarak oluşturulursa, o zaman bu basitçe bagging'e eşit olur. $m = \sqrt{p}$'yi kullanan rastgele ormanlar hem test hatasında hem de bagging üzerindeki OOB hatasında bir azalışa yol açar.[179]

Karar ormanı içinde kaç ağaç oluşturulacaksa o sayıda bootstrap örnekleme yapılır ve her bir örneklem için inBag ve OOB verileri belirlenir. Her bir ağaç inBag verisi ile oluşturulur ve OOB verisi ile her birinin hata oranı hesaplanır. Modelin hata oranı tüm ağaçlardan elde edilen hata oranının ortalaması alınarak belirlenir. Modelin testi için ayrılmış ayrı test verisi varsa bununla da hata hesabı yapılabilir. Test verisi ile elde edilen hata oranı OOB hata oranına yakın bir değer olmaktadır.[180]

[177] John Mingers, 'An Empirical Comparison of Selection Measures for Decision Tree Induction', Machine Learning, 3 (1989), 319–342, <https://doi.org/10.1007/BF00116837>.
[178] Breiman, 'Random Forests',s.5–32
[179] James ve diğerleri, s.320
[180] Akman, s.33



Şekil 4.2. Rastgele orman yönteminde veri seçimi[181]

Bir ağaç oluşturmak için her düğümde kullanılan özelliklerin (bağımsız değişkenlerin) sayısı ve çoğaltılacak ağaç sayısı, rasgele bir orman sınıflandırıcısı oluşturmak için gereken kullanıcı tanımlı parametrelerdir. Her düğümde, en iyi bölünme için yalnızca seçilen özellikler aranır. Böylece, rasgele orman sınıflandırıcı *N* ağaçdan oluşur, burada *N*, yetiştirilecek ağaç sayısıdır, bu da kullanıcı tarafından tanımlanan herhangi bir değer olabilir.[182] Ağaç sayısı *N*'nin artması sonucun daha kararlı olmasını sağlar, fakat bunun yanında hesaplama süresinin uzamasına neden olabilir. Diğer belirlenmesi gereken parametre, ağaç oluşturmak için *p* adet değişkenden *m* adedi, rassal seçilerek elde edilir. Bu seçim $m \approx \sqrt{p}$ işlemiyle yapılır.[183] Yani her bölünmede dikkate alınan değişkenlerin sayısı toplam tahminci sayılarının yaklaşık olarak kareköküne eşittir. Breiman, bu değişken seçimiyle genellikle optimum sonuçlar verileceğini ifade etmektedir.[184] Bu *m* değişken için bilgi kazancı en yüksek olana göre

[181] Akman, s.34
[182] M. Pal, 'Random Forest Classifier for Remote Sensing Classification', International Journal of Remote Sensing, 26.1 (2005), s.217–222 <https://doi.org/10.1080/01431160412331269698>.
[183] James ve diğerleri, s.319
[184] Leo Breiman , Manual On Setting Up, Using, And Understanding Random Forests V3.1, 2002, (https://www.stat.berkeley.edu/~breiman/Using_random_forests_V3.1.pdf, 08.10.2018'de erişildi.)

en iyi bölünme (split) hesaplanmaktadır. Her bir düğümü, dallara ayırmada seçimin rastgele yapılmasının amacı ağaçlar arasındaki korelasyonu minimum yapmak ve hata oranını düşürmektir.[185] Eğer tüm değişkenler ağaç oluşumunda kullanılırsa ağaçlar arasında yüksek korelasyon olur ve hata oranı artar. Her bir düğüm de dallar CART algoritmasının kriterlerine göre oluşturulur rastgele orman sınıflandırıcısı bölünme yöntemi olarak gini indeksini kullanır.[186] Rastgele ormanlar, her bir ayrımı sadece tahmincilerin bir alt kümesini dikkate alırlar. Bu nedenle, bölünmelerin ortalama $(p-m)/p$'si güçlü tahmincileri dikkate almayıp diğer tahmincilere daha fazla şans verilir. Bu süreci ağacı dizayn etme olarak düşünülebilir, dolayısıyla ortaya çıkan ağaçların ortalaması daha az değişken ve bundan dolayı daha güvenilir hale getirilir.[187]

Rastgele ormanlar yönteminde kullanılan bir ölçüm olan proximity (yakınlık) derecesi, veri setindeki kayıtlı verilerin birbirleriyle ne düzeyde ilişkili olduğunu gösterir. Her bireysel ağaç oluşturulduktan sonra, her bir veri ağaçta tahmin edilir ve aynı yaprak düğümde (terminal düğüm) sonlanan deneklerin sayıları yazılarak bir proximity matrisi oluşturulur. Bu sayılar toplam ağaç sayısına bölünerek normalize edilir. Proximity değeri, veri setindeki bir verinin diğer verilerle olan mesafesini ölçen bir gösterge olduğundan, sınıfından uzakta olan aşırı değerlerin (outlier) tespit edilmesinde kullanılabilir. Aşırı değerlerin, sınıflama sonucunda aynı sınıfta yer aldığı tahmin edilen diğer verilerle arasındaki proximity değeri düşük çıkmaktadır. Proximity değerleri kullanılarak veri setindeki eksik değerlerin tahmini yapılabilmektedir. Proximity matrisi ile birbirine yakın verileri bir araya toplayarak kümeleme de yapılmaktadır.[188]

Rasgele orman karar ağaçları için tanımlanmış olmasına rağmen, bu yaklaşım tüm sınıflandırıcılar için geçerlidir. Rastgele orman yönteminin önemli bir avantajı, çok

[185] Akar, s.31
[186] Pal, 217-22
[187] James ve diğerleri, s.320
[188] Akman, s.39

sayıda girdi özniteliğini ele almasıdır.[189] Rastgele ormanın bir başka önemli özelliği de hızlı olmasıdır.

[189] Marina Skurichina and Robert P W Duin, 'Bagging, Boosting and the Random Subspace Method for Linear Classifiers', Pattern Analysis and Applications, 5.2 (2002), s. 121–135 <https://doi.org/10.1007/s100440200011>.

5.Bölüm

UYGULAMA

5.1 Uygulama Amacı

Teknolojik gelişmeler ve internete olan erişimin artması sonucunda dijital reklamcılık, reklam sektörü için yeni ve gerekli bir haline gelmiştir.[190] Dijital reklamcılık, günümüzde çevrimiçi platformların (internet site ve mobil uygulamaları) ve içeriklerin en yüksek payını finanse eden baskın bir faaliyet haline gelmiştir.[191] İnternet ağlarının yaygınlaşması ve internete erişiminin bir ihtiyaç haline gelmesi internet sitelerinde ve diğer dijital platformlardaki reklamların kullanılmasını yaygınlaştırmıştır. Markalar için vazgeçilmez olan bu platformda yapılan reklamlar detaylı bir şekilde ölçümlenebilmektedir. Belirli ölçümleme araçları (Google Analytics, Google Adwords, Criteo, vb.) kullanılarak yapılan reklam yayınların detaylı bir istatistiği çıkarılır. Amaç bu verilere dayanarak ve çeşitli istatistiksel yöntemler kullanarak reklam yayınlarını daha etkin bir biçimde kullanmaktır. Bir başka deyişle ise firmalar ve markalar için bir maliyet unsuru olan reklam yayınlarında harcanan bütçeyi daha etkin kullanmaktır.

Dijital reklam yayınlarında genelde hedef kitle belirlenir ve bu hedef kitleye uygun reklam gösterimleri yapılır. Bu reklam gösterimleri çeşitli internet sitelerinde resim ve video reklam olarak, arama ağlarında (google, yandex vb.) metin reklam olarak internet kullanıcılarının karşısına çıkmaktadır.

Uygulamada bir inşaat firmasının, belirli bir dönemde yapmış olduğu dijital reklam yayınlarından elde edilen veriler (kullanıcı verileri) R[192] programında randomForest[193] kütüphanesi kullanılarak analiz edilmiştir. analiz edilmiştir. İnşaat

[190] Ken Burtenshaw, Caroline Barfoot ve Nicholas Mahon, The Fundamentals of Creative Advertising,2.Basım,CH. AVA publishing, Lausanne,2006, s.64. Aktaran Kübra YÜREKLİ, Dijital Reklamcılıkta Reklam Ajansı - Reklam Veren İlişkisinin Analizi, İstanbul, (Yüksek Lisans Tezi,2016)
[191] Andrew McStay, Digital Advertising, New York: Palgrave Macmillan, 2010, s.12. Aktaran Kübra Yürekli (Yüksek Lisans Tezi,2010)
[192] R Core Team (2017). R: A Language and Environment for Statistical Computing. R Foundation for St atistical Computing, Vienna, Austria
[193] Andy Liaw ve Matthew Wiener, Classification and Regression by randomForest. R News 2(3),2 002, s.18-22.

firmasının reklamlarını dijital platformlarda görmüş olan kullanıcılar bu reklamlar sayesinde firma sitesine gelmiş ve burada konut alımıyla ilgili olarak iletişim bilgilerini firmaya bir form aracılığıyla göndermişlerdir. Firma çalışanları kullanıcıların iletişim bilgileri sayesinde onlara ulaşmış ve konutlar hakkında daha ayrıntılı bilgi vermek için uygun bir zamanda satış ofislerine davet etmişlerdir. Tüm bu süreç yani kullanıcının reklamı görmesi, ilgili reklam sayesinde firma sitesine gelmesi, form doldurması ve satış ofisine gelmesi kayıt altına alınmıştır. Uygulama sayesinde bu veri seti analiz edilmiş ve kullanıcılar satış ofisine gelen ve gelmeyen olarak oluşturulan model tarafından tahmin edilmiştir. Amaç bundan sonraki süreçte oluşturulan model yardımıyla yeni kullanıcıların satış ofisine gelip gelmeyeceğinin tahmin edilmesidir. Böylelikle firma hangi kullanıcıların satış ofisine geleceğini önceden öngörerek hedef kitle analizini gerçekleştirmiş ve elindeki kullanıcı verilerini daha etkin şekilde kullanmış olacaktır.

5.2 Uygulama Kapsamı ve Veri Yapısı

Uygulamada bağımlı değişken olarak kullanıcıların satış ofise gelip gelmemesi, bağımsız değişken olarak ise kullanıcın iletişim bilgilerini **hangi gün** firma çalışanlarına gönderdiği, kullanıcının cinsiyeti, reklamı **hangi internet sitesinde (mecra)** görüp firma sitene geldiği, reklamı **hangi reklam alanında** (doğal, 300*250 görsel boyutu vb.) gördüğü, **hangi cihazdan** (bilgisayar veya mobil) gördüğü, kullanıcının daha önce ilgili firmada kayıtlı olup olmaması ve bu formu hangi amaçla doldurduğu(yatırım, ev sahibi olma vb.) olmak üzere toplamda 7 adet bağımsız değişken kullanılmıştır.

Uygulamada kullanılan Rastgele Ormanlar yöntemiyle veri seti analiz edilmiş ve kullanıcılar satış ofisine gelen ve gelmeyen olarak model tarafından tahmin edilmiştir.

Bağımlı değişkene göre eğitim ve test verileri;

Eğitim Veri Seti;

Satış Ofisi	0	1	Toplam
	397	44	441

Test Veri Seti;

Satış Ofisi	0	1	Toplam
	81	14	95

İlk başta 1.492 kullanıcı verisi elde edilmiş ancak veri temizleme aşamasında geriye 536 kullanıcı verisi kalmıştır. Bu temizleme işlemi kullanıcıların hatalı iletişim bilgisi vermesi, ulaşılamamaları, formu yanlışlıkla doldurmaları ve konut alımıyla ilgilenmedikleri bilgisini vermeleri üzere analizden çıkarılmıştır. Analiz bu 536 kullanıcı verisiyle yapılmıştır. Bu kullanıcı verisinin %80'i eğitim,%20'si test verisi olarak kullanılmıştır. Bağımlı değişken olan satış ofisine gelme durumunu gösteren değişken, gelenler 1, gelmeyenler 0, sembolleriyle gösterilmiştir.

Uygulamada kullanılan bağımsız değişkenlerin hepsi kategoriktir.

Gün değişkeni; 1'den 7'ye kadar olan rakamlarla gösterilmiştir. 1 pazartesi gününe karşılık gelmekte olup sırayla devam ederek 7 Pazar gününe karşılık gelmektedir. Kullanılan R programında gun diye kodlanmıştır.

Cinsiyet değişkeni; 0 ve 1 rakamlarıyla gösterilmiştir. 0 erkeklere 1 kadınlara karşılık gelmektedir. Kullanılan R programında cins diye kodlanmıştır.

Cihaz değişkeni; 0 ve 1 rakamlarıyla gösterilmiştir. 0 mobil(cep telefonu) 1 bilgisayara karşılık gelmektedir. Kullanılan R programında cih diye kodlanmıştır.

Mecra değişkeni;1'den 23'e kadar olan rakamlarla gösterilmiştir. 1. mecra, 2. mecra olarak 23'e kadar numaralandırılmıştır. Kullanılan R programında mec diye kodlanmıştır.

Kreatif değişkeni; 1'den 21'e kadar olan rakamlarla gösterilmiştir. 1. kreatif, 2. kreatif olarak 21'e kadar numaralandırılmıştır. Kullanılan R programında kre diye kodlanmıştır.

Kayıt değişken; 0 ve 1 rakamlarıyla gösterilmiştir. 0 kayıtlı olmayan kullanıcıya, 1 kayıtlı kullanıcıya karşılık gelmektedir. Kullanılan R programında kay diye kodlanmıştır.

Görüş değişken; 1'den 6'ya kadar olan rakamlarla gösterilmiştir. Kullanılan R programında grs diye kodlanmıştır.

1; 1+1 konutlarla ile ilgilenen kullanıcı

2;2+1 konutlarla ilgilenen kullanıcı

3;3+1 konutlarla ilgilenen kullanıcı

4; detaylı bilgi almak isteyen kullanıcı

5; yatırım yapmak isteyen kullanıcı

6; diğerler amaçlar için, olarak ifade edilmiştir.

Bağımsız değişkenlerin 0 ve 1 arasında olması ve değişken normlarının eşit olması için gün, mecra, kreatif ve görüş değişkenlerine aşağıdaki denklemde gösterildiği gibi normalizasyon dönüşümü yapılmıştır.[194]

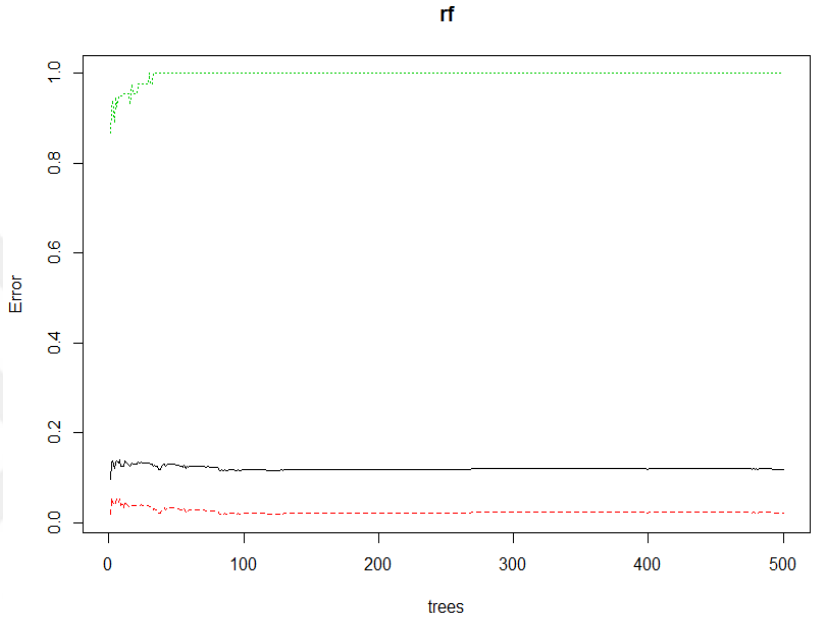
$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \qquad (5.1)$$

5.3 Elde Edilen Bulgular

Uygulama analiz sonuçları, R programında randomForest paketi kullanılarak elde edilmiştir.

[194] Han, Kamber ve Pei, s.114

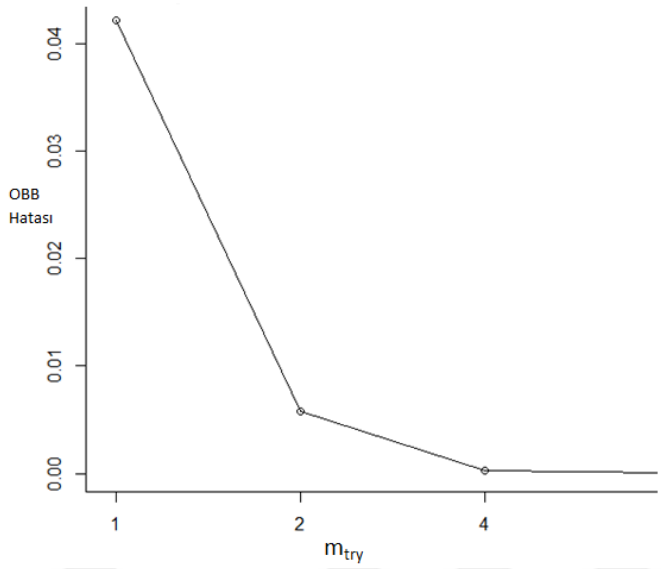
Öncelikle ağaç sayısı ve ağaç oluşumda her bölünmede dikkate alınan değişkenlerin sayısı keyfi olarak belirlenir ve yapılan analiz sonucunda bu değerler için optimum değerler bulunarak bu parametreler yeniden belirlenir. Keyfi olarak ormanda oluşan ağaç sayısının 500 seçildiği ve ağaç oluşumda her bölünmede dikkate alınan değişkenlerin yine keyfi olarak 2 olarak seçildiği yöntemin çıktılarına bakacak olursak.



Grafik 5.1. Ağaç sayısı grafiği

Grafik OOB hatasının ağaç sayısına bağlı olarak grafiğini çizmektedir. Ağaç sayısının artıkça bu hatanın sabitleştiğini görülmektedir. Yeni oluşturulacak parametre değerinde ağaç sayısı 200 olarak alınabilir ya da 500 olarak kalabilir. Ağaç sayısının fazla olması OOB'de bir azalışa yol açmasa da ormanın daha kararlı olmasını sağlar.[195]

[195] James ve diğerleri, s.319



Şekil 5.2. Ağaç oluşumda her bölünmede dikkate alınan değişken sayısı

OBB hatasını en aza indirecek her bölünmede dikkate alınan değişken sayısının grafiğine bakarak bu parametre değerinin 4 olması gerektiğini söyleyebiliriz.

Parametrelerinin optimal değerinin ne olacağı belirlendikten sonra algoritma yeniden çalıştırılır ve aşağıdaki sonuçlar elde edilir.

Tablo 5.1 Eğitim verisi için sınıflandırma tablosu

	Gerçek	
Tahmin	**0**	**1**
0	**395**	**17**
1	**2**	**27**

Tablo 5.2 Eğitim verisi için sınıflandırma tablosuna ait değerlendirme ölçütleri

Doğruluk=	**95,69%**
Yanlış Sınıflandırma Oranı=	**4,31%**
Kesinlik=	**93,10%**
Doğru Pozitif Oran=	**61,36%**
Yanlış Pozitif Oran=	**4,55%**
Doğru Negatif Oran=	**99,50%**
Yanlış Negatif Oran=	**38,64%**
Kappa=	**71,73%**

Kappa istatsistiği; Cohen'in kappa katsayısı olarak bilinir ve iki değerleyici arasındaki karşılaştırmalı uyuşmanın güvenirliğini ölçen bir istatistik yöntemidir. Eğer Pr(a) iki değerleyici için örtüşen uyuşmaların toplama olan oranı ve Pr(e) ise bu uyuşmanın şans eseri ortaya çıkma olasılığı ise, Cohen'in kappa katsayısı bulunması için kullanılacak formül şu olur: Aşağıdaki formül yardımıyla hesaplanır. Kappa değerinin 1'e yakın olması iki değerleyicinin o derece iyi bir şekilde uyuşmakta olduğunu göstermektedir.[196]

$$\kappa = \frac{\Pr(\alpha) - \Pr(e)}{1 - \Pr(e)} \qquad (5.2)$$

Tabloda da görüldüğü gibi eğitim verisinde doğruluk %95 gibi çok yüksek bir değer çıkmıştır. Doğru pozitif oranın %61,36 olduğunu görmekteyiz. Bu değerler karar ormanın eğitim verisini iyi anlamda öğrendiğini göstermektedir. Kappa istatistiğinin %70 olması da modelin yüksek oranda gerçek değerlerle uyuştuğunun bir göstergesidir.

[196] Jacob Cohen , A coefficient of agreement for nominal scales, Educational and Psychological Measurement Vol.20, No.1, 1960,pp.37-46

Tablo 5.3. Test verisi için sınıflandırma tablosu

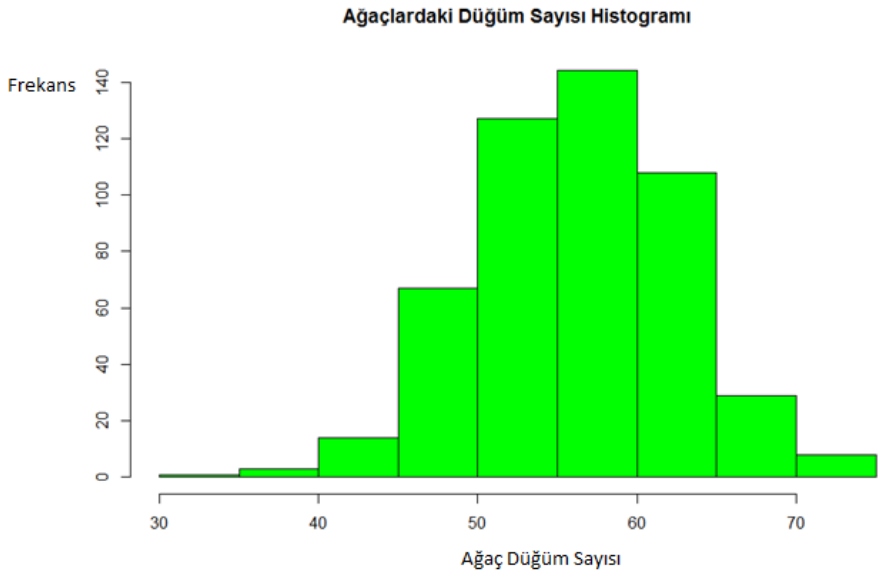
	Gerçek	
Tahmin	**0**	**1**
0	**77**	**13**
1	**4**	**1**

Tablo 5.4. Eğitim verisi için sınıflandırma tablosuna ait değerlendirme ölçütleri

Doğruluk=	**82,11%**
Yanlış Sınıflandırma Oranı=	**17,89%**
Kesinlik=	**20,00%**
Doğru Pozitif Oran=	**7,14%**
Yanlış Pozitif Oran=	**28,57%**
Doğru Negatif Oran=	**95,06%**
Yanlış Negatif Oran=	**92,86%**
Kappa=	**0,05%**

Test verisinin doğruluğunun yüksek %82 gibi yüksek bir değer olmasına rağmen doğru pozitif oranının %7 olması veri yapısından kaynaklanmaktadır. Bu durumu Brieman ve arkadaşları bir makalesinde şöyle açıklamaktadır. Random forest sınıflandırıcısı, çoğunluk sınıfını azınlık sınıfına göre daha doğru sınıflandırma eğilimi gösterir. Bu nedenle azınlık sınıfını iyi bir şekilde sınıflandıramaz. Bu durumda ağırlıklandırılmış Rf algoritması kullanılır. Ancak yapılan çalışmalar dengesiz veriyi

öğrenmek için kullanılan ağırlıklı Rf ile dengeli Rf (uygulamada kullanılan) arasında net bir kazananın olmadığını ortaya çıkarmıştır.[197]

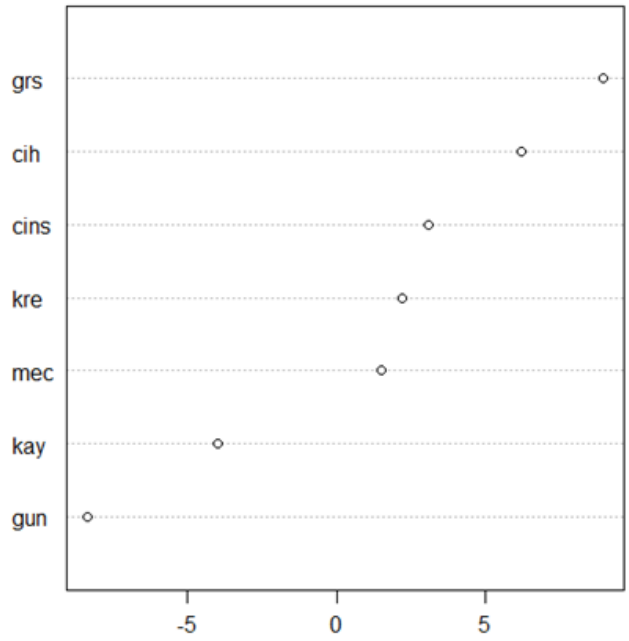


Şekil 5.3. Ağaçlardaki düğüm sayısı histogramı

Toplamda 501 tane ağaçtan oluşan ormanda düğüm sayılarının histogramı yukarıda verilmiştir. Ağaçlardaki düğüm sayısı yukarıdaki grafikten görüldüğü gibi ortalama olarak 50-60 arasında değişmektedir.

Ortalama Düşüş Doğruluğu

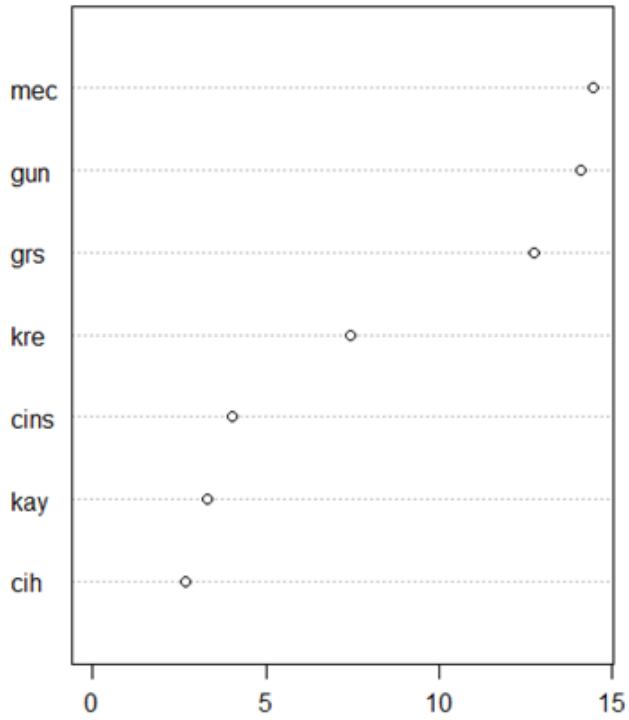
[197] Chao Chen, Andy Liaw, and Leo Breiman, 'Using Random Forest to Learn Imbalanced Data', (2003), s.1–12.



Şekil 5.4 Değişkenlerin doğruluğa olan katkıları

Grafik herhangi bir değişken olmadığında model performansının ne kadar kötü olduğunu test eder. Ağaçları yaparken herhangi bir değişkenini kaldırırsak, doğrulukta ortalama azalma ne olur sorusunun cevabını veren grafiktir. Örneğin grs (görüş) değişkeninin değeri çok yüksek olduğundan doğruluğun hesaplanmasında yüksek bir öneme sahip olduğu söylenir.

Gini Değeri Ortalama Düşüşü



Şekil 5.5 Düğümlerin değişkenlere göre saflık değerleri

Bu grafik, herhangi bir değişken olmadan ağacın sonunda düğümlerin ne kadar saflıkta olduğunu ölçer. Yani değişkenlerden biri modelden çıktığında Gini katsayında ne kadar bir azalma olacağının cevabını verir. Örneğin mec (mecra) değişkeni değeri yüksek olduğu için diğer değişkenlere göre Gini indeksinde yüksek katkısı olduğu söylenir.

Tablo 5.5. Değişkenlerin doğruluk ve gini katkıları için sayısal değerler

Değişkenler	Ortalama Doğruluk	Gini
gun (gün)	-8,36	14,09
cins (cinsiyet)	3,11	4,03
cih (cihaz)	6,22	2,67
mec (mecra)	1,49	14,46
kre (kreatif)	2,2	7,43
grs (görüşme)	8,97	12,75
kay (kayıt)	-4	3,31

Bu tablo yukarıdaki verilen grafiklerde yer alan değişkenlerin sayısal değerlerini göstermektedir.

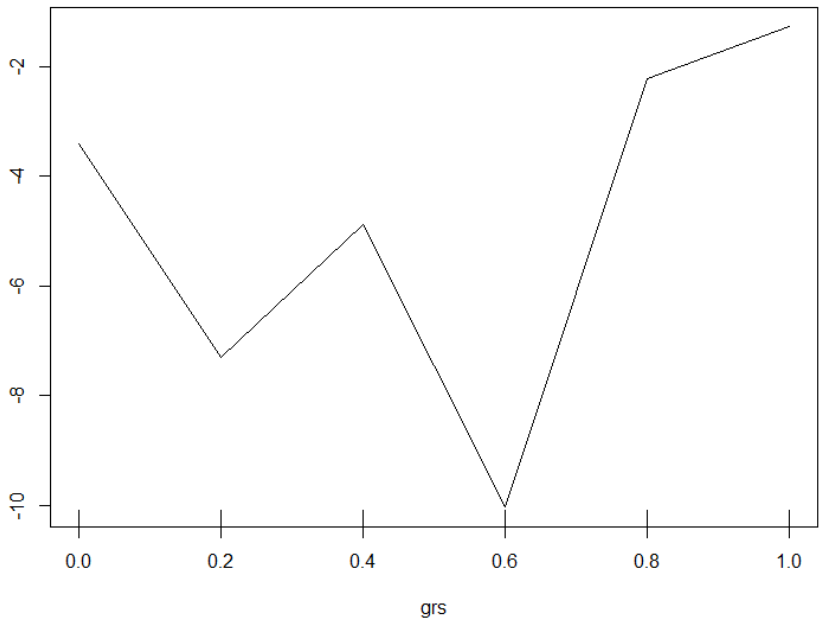
Tablo 5.6. Değişken kullanım sayıları

Değişkenler	Kullanım Sayıları
gun (gün)	7.760
cins (cinsiyet)	2.215
cih (cihaz)	873
mec (mecra)	7.119
kre (kreatif)	4.120
grs (görüşme)	3.534
kay (kayıt)	2.275

Bu tablo, değişkenlerin ağaç oluşumdaki kullanım sayılarını göstermektedir. Örneğin en fazla kullanılan değişken gun(gün) değişkenidir. Bu değişkenin ortalama doğruluğa katkısı -8,36 iken gini indeksine katsısı ise 14,09 olarak hesaplanmıştır.

Kısmi Bağımlılık grafikleri;

Bu grafikler bir değişkenin, sınıf olasılığı üzerindeki marjinal etkisini vermektedir.



Şekil 5.6 Görüş değişkenin kısmi bağımlılığı

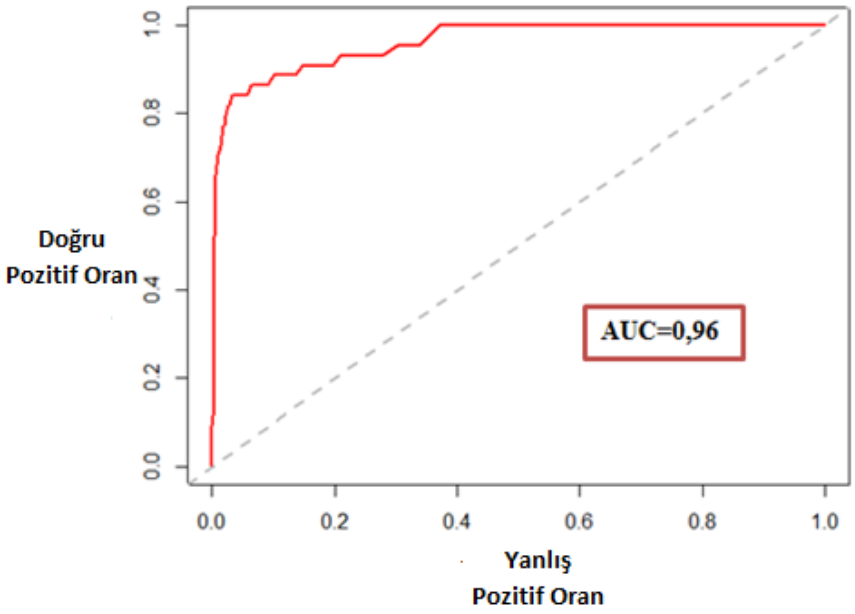
Grafiği yorumlandığında grs (görüş) değişkeni 0,6 değerinden büyük olduğunda 1 sınıfını 0,6 değerinden küçük olmasına göre daha kuvvetli tahmin etmektedir.

Diğer değişkenlerin kısmi bağımlılıkları da o değişkenlere göre çizilen grafiklere bakılarak yapılabilir. Bu grafikler ekler kısmında verilmiştir.

ROC Eğrileri

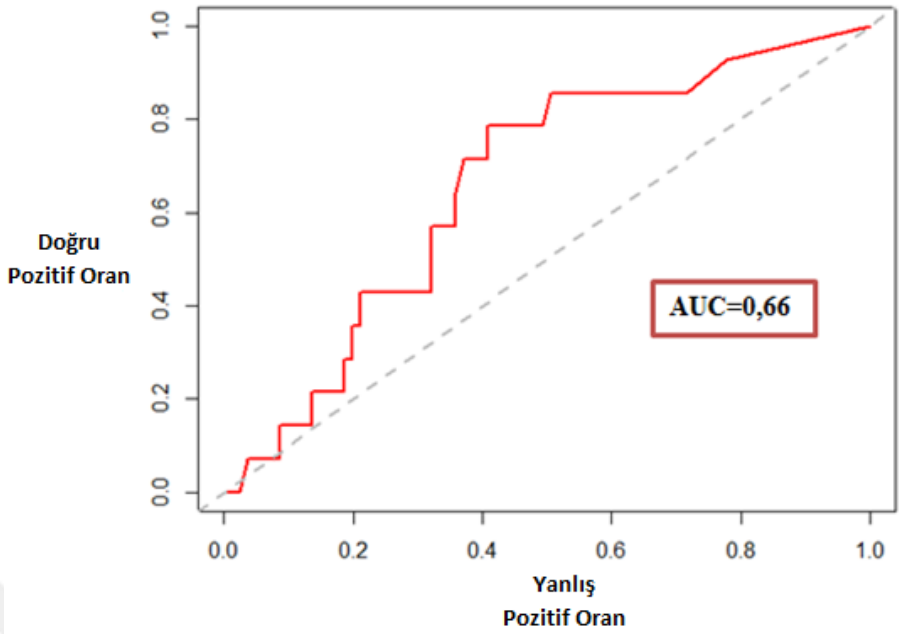
Kullanılan random forest algoritması gözlemleri aldığı oylara göre sınıf atamaları yapmaktadır. Bu oylar her bir gözlem için hesaplanır ve OOB örnek ağacı sınıflandırması için de hesaplanır. Bu oylar kısaca bir olasılığı temsil eder ve bu nedenle ROC grafiği çizilip AUC değeri hesaplanabilir.[198]

[198] https://stats.stackexchange.com/questions/188616/how-can-we-calculate-roc-auc-for-classification-algorithm-such-as-random-forest- 12.11.2018 tarihinde erişildi.



Şekil 5.7. Eğitim verisi için ROC eğrisi

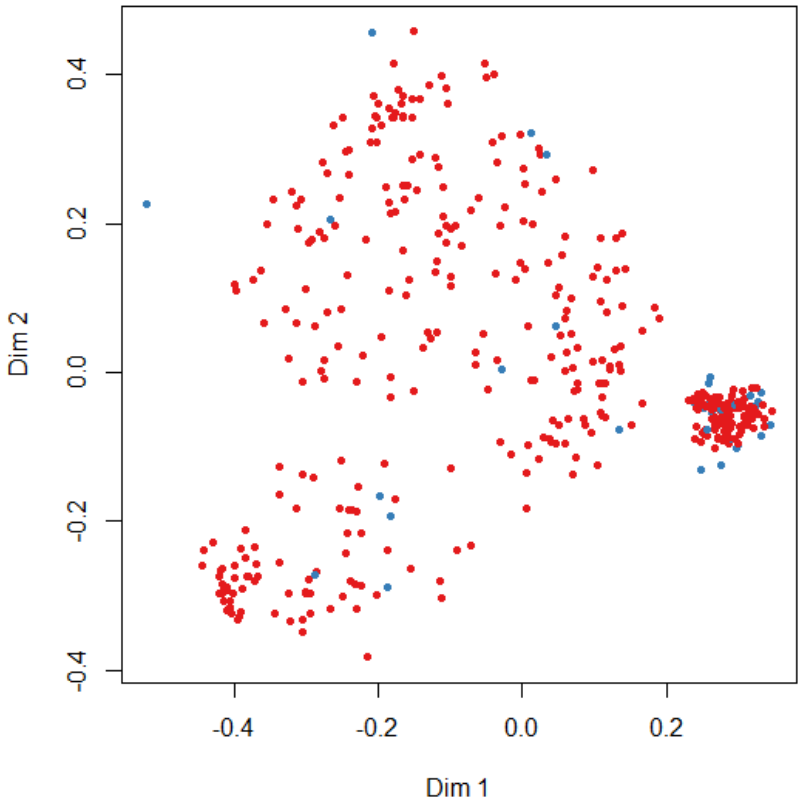
R programındaki ROCR paketi kullanılarak çizilen ROC eğrisi şekil 5.7'de gösterilmektedir. Yine aynı paket kullanılarak hesaplanan AUC (area under curve) değeri **0,96** olarak hesaplanmıştır. Bu değer sınıflandırma performansının çok iyi olduğunu göstermektedir.



Şekil 5.8. Test verisi için ROC eğrisi

Test verisi için hesaplanan ROC eğrisi şekil 5.8'de gösterilmiştir. Bu eğri altında kalan AUC değeri de 0,66 olarak bulunmuştur. Bu değer 0,5'nin üzerinde olduğu için sınıflandırma performansısın iyi olduğu söylenebilir.

Çok Boyutlu Ölçekleme Grafiği



Şekil 5.9 Eğitim verisindeki geliş değişkenin çok boyutlu ölçekleme grafiği

Veride 0 sınıfına ait olan gözlemler çoğunlukta olduğu için kırmızı renkli olanlar 0 sınıfını, mavi renkli olanlar ise 1 sınıfını göstermektedir.

Sonuç ve Öneriler

Çalışmada bir inşaat firmasının dijital reklam yayınları sonucunda elde edilen kullanıcı verileri analiz edilmiştir. Veri analizinde Rastgele Orman Yöntemi kullanılarak sınıflandırma modeli oluşturulmuştur. Kullanıcıların satış ofisine gelip gelmemesinin bağımlı değişken olduğu sınıflandırma modelinde, kullanıcın iletişim bilgilerini hangi **gün** firma çalışanlarına gönderdiği, kullanıcının **cinsiyeti**, reklamı hangi **internet sitesinde** görüp firma sitesine geldiği, reklamı hangi **reklam alanında** gördüğü, hangi **cihazdan** (bilgisayar veya telefon) gördüğü, kullanıcının daha önce ilgili firmada **kayıtlı olup olmaması** ve bu formu hangi **amaçla** doldurduğu (yatırım, ev sahibi olma vb.) olmak üzere toplamda 7 adet bağımsız değişken kullanılmıştır.

Bağımsız değişkenlerin normlarını eşitlemek için tüm bağımsız değişkenleri 0-1 aralığına dönüştüren normalizasyon dönüşümü yapılmıştır. Oluşturulan sınıflandırma modelinde eğitim verisinde sınıflandırma doğruluğu %95 seviyesinde olurken test verisinde bu oran %82 seviyelerinde gerçekleşmiştir. Genellikle bu tarz çalışmalarda test verisi dikkate alınarak değerlendirme yapılmaktadır.

Değişken bazında en fazla görüş değişkeni olmak üzere, cihaz değişkeninin de doğruluğu artırıcı bir etkisi olduğu ortaya çıkmıştır. Yani kullanıcıların (müşterilerin) satış ofisine gelmelerindeki en önemli değişken (tahmin doğruluğunu arttıran değişken) görüş değişkeni ve cihaz değişkeni olduğu görülmektedir. Bu değişkenler tahmin doğruluğuna en fazla katkıyı veren değişkenlerdir. Bu sonuç bundan sonraki form doldurma sayfalarında kullanıcıların konut alma amaçlarını yansıtacak bir seçeneğin olması gerektiğinin ön bilgisini vermektedir. Mecra ve gün değişkenlerinin gini indeksine en fazla katkıyı sağlayan değişkenler olduğu görülmüştür. Mecra ve gün değişkenleri ormandaki karar ağaçlarında en fazla kullanılan değişkenler olmuşlardır. Ayrıca oluşturulan karar ağaçları ortalama 50-60 düğümden oluşmaktadır.

Kullanılan random forest algoritmasında gözlemlerin aldıkları oylara göre hangi sınıfta oldukları belirlenir. Bu oylar bir olasılığı temsil etmektedir. Böylece ROC grafiği çizilip AUC değeri hesaplanabilir. Çizilen ROC eğrilerinde eğitim verisi için eğri sol üst köşeye yakın olup eğri altında kalan alan ise %92'dir. Test verisi için

çizilen ROC eğrisi sol üst köşeye fazla yakın olmamasına rağmen eğri altında kalan alan %66 seviyelerindedir. Bu değerlere bakarak da yapılan sınıflandırmanın iyi bir seviyede olduğu söylenebilir.

Oluşturulan sınıflandırma modeli ile veri tabanına gelen yeni bir kullanıcı bu modele göre belirli hata oranlarıyla sınıflandırılabilir. Böylece yeni kullanıcılar hakkında önceden bir ön bilgiye sahip olunabilir. Firma müşteri portföyünü bu bilgiler ışığında değerlendirme imkânına kavuşur. Bir başka kullanım amacı ise şu anki mevcut durumda satış ofisine gelmeyen ancak model tarafından satış ofisine geldi olarak tahmin edilen kullanıcılar da belirlenip bu kullanıcılara özel hedeflemeler ve satış ofisine gelmelerini sağlayacak farklı pazarlama stratejileri geliştirilebilir.

Eğitim verisinde doğru negatif oran (duyarlılık) değeri %99 seviyelerinde, doğru pozitif oran(özgüllük) değeri %61 gibi iyi bir seviyelerde, test verisinde doğru negatif oran (duyarlılık) değeri %95 seviyesinde ve doğru pozitif oran(özgüllük) %7 seviyelerinde olmuştur. Test verisinde özgüllük değerini küçük ve duyarlılık değerinin yüksek çıkma nedeni azınlık sınıf varlığından kaynaklanmaktadır. Yani 0 olarak etiketlenen sınıftaki gözlem sayısı, 1 olarak etiketlenen sınıftaki gözlem sayısından çok fazladır. Bu tip verilerde çoğunluk sınıfı iyi bir seviyede tahmin edilirken azınlık sınıfın tahmin doğruluğu düşük seviyelerde gerçekleşmektedir. Bu durumda azınlık sınıfında gözlemlerin sayısı arttırılabilir SMOTEBoost veya XGBoost gibi farklı algoritmalar kullanılabilir.

KAYNAKÇA

Akar, Özlem, 'Rastgele Orman Sınıflandırıcısına Doku Özellikleri Entegre Edilerek Benzer Spektral Özellikteki Tarımsal Ürünlerin Sınıflandırılması, Trabzon (Doktora Tezi , 2013)

Akpınar, Haldun, 'Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği, İ.Ü. İşletme Fakültesi Dergisi,C:29, S: 1/Nisan 2000, s. 1-22

Alpaydin, Ethem, Introduction to Machine Learning, 2.Edition, London,The MIT Press, 2010, <https://doi.org/10.1016/j.neuroimage.2010.11.004>

Altaş, Dilek, ve Murat Çinko, 'Bootstrap Yönteminin Rıdge Regresyonda Uygulanması', Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi Cilt XXII, Sayı 1, 2003, s,281–292

Bauer, Eric, and Ron Kohavi, 'An Empirical Comparison of Voting Classification Algorithms : Bagging , Boosting , and Variants', Machine Learning, 36 (1999), s.105–139

Bohanec, Marko, and Ivan Bratko, 'Trading Accuracy for Simplicity in Decision Trees', Machine Learning, 15 (1994), s.223–250 <https://doi.org/10.1023/A:1022685808937>

Breiman, Leo, 'Bagging Predictors', Machine Learning, 24 (1996), s.123–40

Breiman, Leo, 'Random Forests', Machine Learning, 45 (2001), s.5–32 <https://doi.org/10.1023/A:1010933404324>

Breiman, Leo, ‘Random Forests - Random Features, Tecnical Report 567, Statistic Department, University of California, Berkeley, (Https://Www.Stat.Berkeley.Edu/~breiman/Random-Forests.Pdf, 08.10.2018’de Erişildi)’, 1999, s.1–29

Breiman, Leo, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone, Classification And Regression Trees, Chapman & Hall/CRC Texts in Statistical Science Series, 1984, i <https://doi.org/10.1002/widm.8>

Chen, Chao, Andy Liaw, and Leo Breiman, ‘Using Random Forest to Learn Imbalanced Data’, s.1–12

Cohen, Jacob, A coefficient of agreement for nominal scales, Educational and Psychological Measurement Vol.20, No.1, 1960,pp.37-46

Doğan, Nurhan, and Kazım Özdamar, ‘CHAID Analizi ve Aile Planlaması Le Lgili Bir Uygulama’, T Klin Tıp Bilimleri 2003, 23 (2003), s.392–398

Fayyad, Usama, Gregory Piatetsky-shapiro, and Padhraic Smyth, ‘From Data Mining to Knowledge Discovery in Databases, AI Magazine Volume 17 Number 3,1996’, 17 (1996), s.37–54

Freund, Yoav, and Robert E. Schapire, ‘Experiments With A New Boosting Algorithm’, ICML ’96: Proceedings of the 13th International Conference on Machine Learning, 1996, s.148–156

Friedman, Jerome H, ‘Lazy Decision Trees’, Aaai, 34 (1997), s.167–180

Gülpınar, Vildan, ‘Avrupa Birliği Ülkeleri Ile Türkiye’nin Ekonomik Göstergelerinin Karar Ağacı Yöntemi Ile Karşılaştırılması,İstanbul,(Yüksek Lisans Tezi,2008)’, 2008

James, Gareth, Daniela Witen, Trevor Hastie, and Robert Tibshirani, An Introduction to Statistical Learning with Applications in R, Performance Evaluation, 2014, LXİV <https://doi.org/10.1016/j.peva.2007.06.006>

Jiawei Han, Micheline Kamber ve Jian Pei, Data Mining Concepts and Techniques, Third Edit, Waltham, Morgan Kaufman Publıshers,2012,

Jiawei Han ve Micheline Kamber, Data Mining:Concepts and Techniques (Simon Fraser University: Morgan Kaufman Publıshers, 2000)

Kass, G. V., ‘An Exploratory Technique for Investigating Large Quantities of Categorical Data’, Applied Statistics, 29 (1980), 119 <https://doi.org/10.2307/2986296>

Kolcz, Aleksander, Abdur Chowdhury, and Joshua Alspector, ‘Data Duplication: An İmbalance Problem?’, 2003 <https://doi.org/10.1.1.72.8356-1>

Lee, Sauchi Stephen, ‘Noisy Replication in Skewed Binary Classification’, Computational Statistics and Data Analysis, 34 (2000), 165–91 <https://doi.org/10.1016/S0167-9473(99)00095-X>

Liaw, Andy ve Wiener, Matthew, Classification and Regression by randomForest. R News 2(3),2002, s.18-22.

Loh, Wei-Yin, and Yu-Shan Shih, ‘Split Selection Methods for Classification Trees’, Statistica Sinica, 7 (1997), 815–40 <https://doi.org/10.2307/24306157>

Mingers, John, ‘An Empirical Comparison of Selection Measures for Decision Tree Induction’, Machine Learning, 3 (1989), s.319–342 <https://doi.org/10.1007/BF00116837>

Moore, Samuel A, James Kurinskas, and Gary M Weiss, ‘Are Decision Trees Always Greener on the Open (Source) Side of the Fence?’, Dmin 2009, s.185–188

Özkan, Yalçın, ‘Veri Madenciliği Yöntemleri, 1.Basım, İstanbul: Papatya Yayınları, 2008’

Pagallo, Giulia, and David Haussler, ‘Boolean Feature Discovery in Empirical Learning’, Machine Learning, 5 (1990), s.71–99 <https://doi.org/10.1023/A:1022611825350>

Pal, M., ‘Random Forest Classifier for Remote Sensing Classification’, International Journal of Remote Sensing, 26 (2005), s.217–222 <https://doi.org/10.1080/01431160412331269698>

Polikar, Robi, ‘Ensemble Based Systems in Decision Making’, IEEE Circuits and Systems Magazine, 6 (2006), s.21–44 <https://doi.org/10.1109/MCAS.2006.1688199>

Provost, Foster, and Tom Fawcett, ‘Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions’, KDD-97 Proceedings, 1997, s.43–48

Provost, Foster, and Tom Fawcett, ‘Robust Classification for Imprecise Environments’, Machine Learning, 42 (2001), s.203–231 <https://doi.org/10.1023/A:1007601015854>

Quinlan, J. R., ‘Induction of Decision Trees’, Machine Learning, 1 (1986), s.81–106 <https://doi.org/10.1023/A:1022643204877>

Quinlan, J. R., ‘Simplifying Decision Trees’, International Journal of Man-Machine Studies, 27 (1987), s.221–234 <https://doi.org/10.1016/S0020-7373(87)80053-6>

R Core Team (2017). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria

Rokach, Lior, Pattern Classıfıcatıon Using Ensemble Methods (Serıes In Machine Perceptıon And Artificial Intelligence-Vol. 75) (Singapore: World Scientific Publishing Company, 2010)

Rokach, Lior, and Oded Maimon, Data Mining With Decision Tree Theory and Applications,2.Edition,World Scientific Publishing,2015,

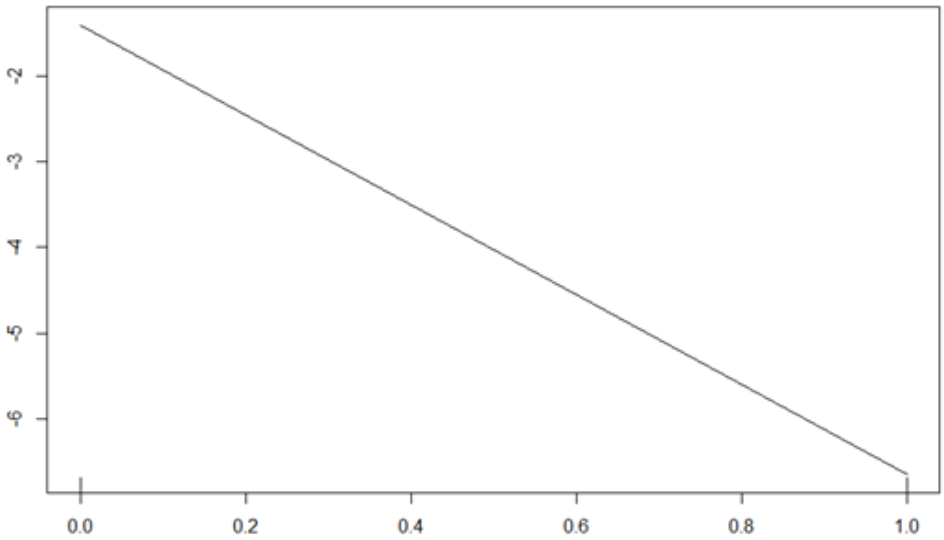
Shannon, Claude E, ‘A Mathematical Theory of Communication’, The Bell System Technical Journal, 27 (1948),s.379–423 <https://doi.org/10.1145/584091.584093>

Skurichina, Marina, and Robert P W Duin, ‘Bagging, Boosting and the Random Subspace Method for Linear Classifiers’, Pattern Analysis and Applications, 5 (2002), s.121–135 <https://doi.org/10.1007/s100440200011>

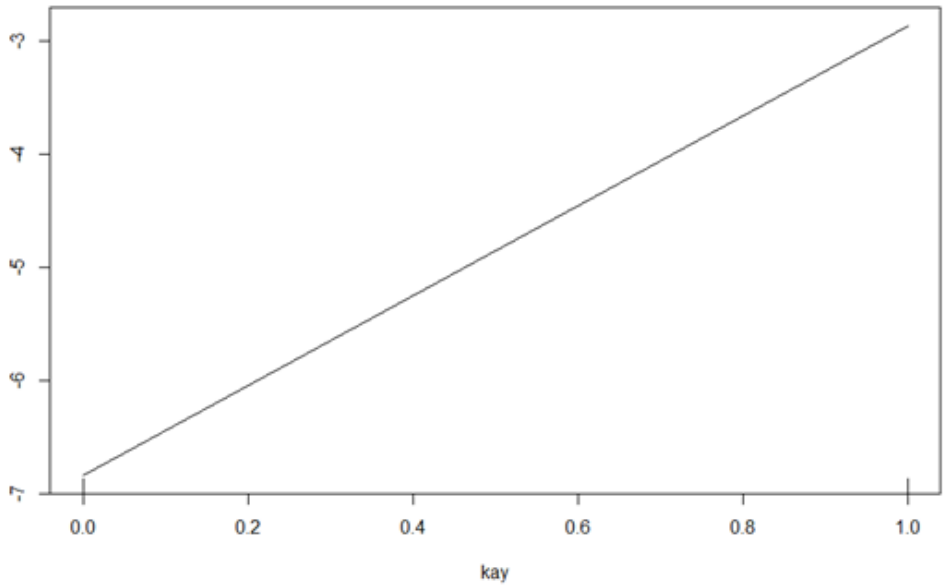
Zaïane, Osmar R., ‘Introduction to Data Mining,University of Alberta Fall, 1999, s.1–15

EKLER

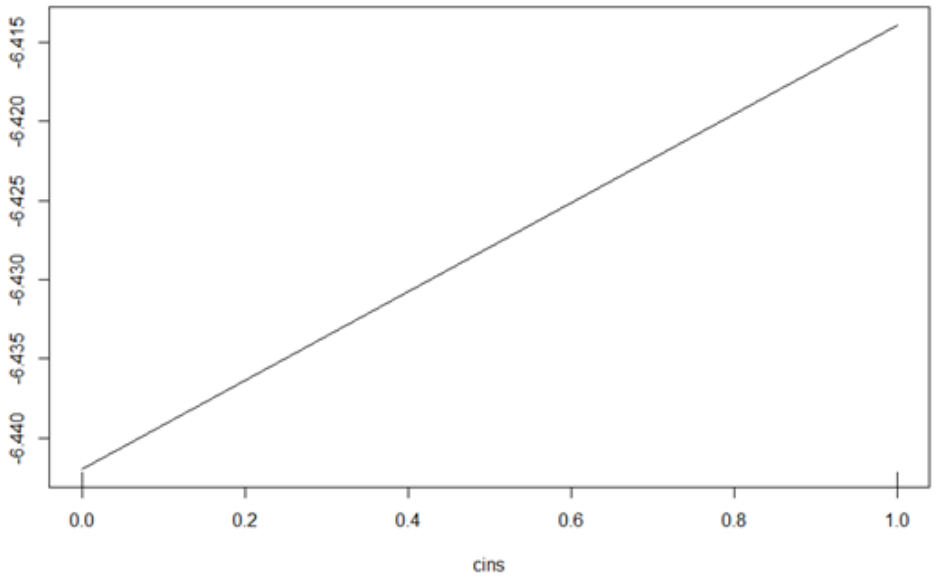
Cihaz değişkeni için kısmi bağımlılık grafiği;



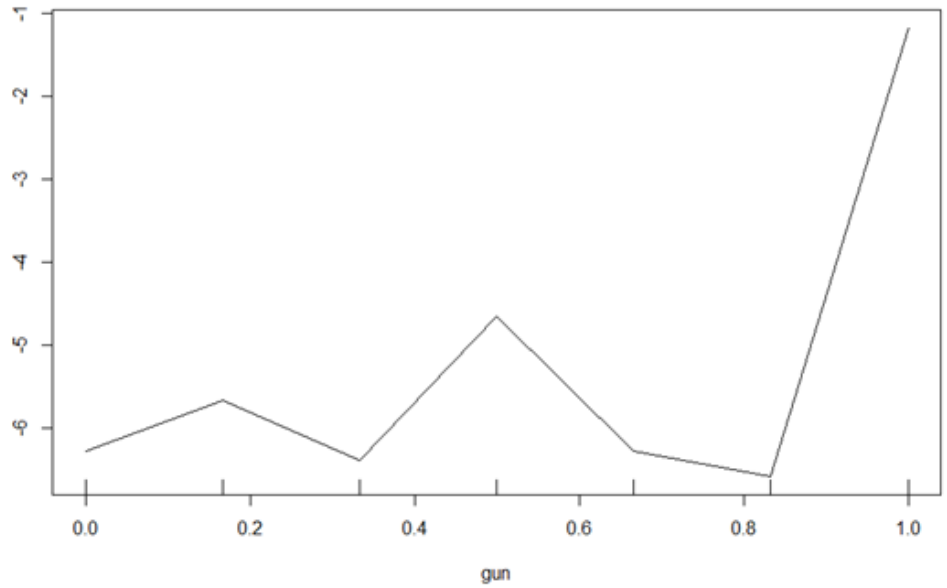
Kayıt değişkeni için kısmi bağımlılık grafiği;



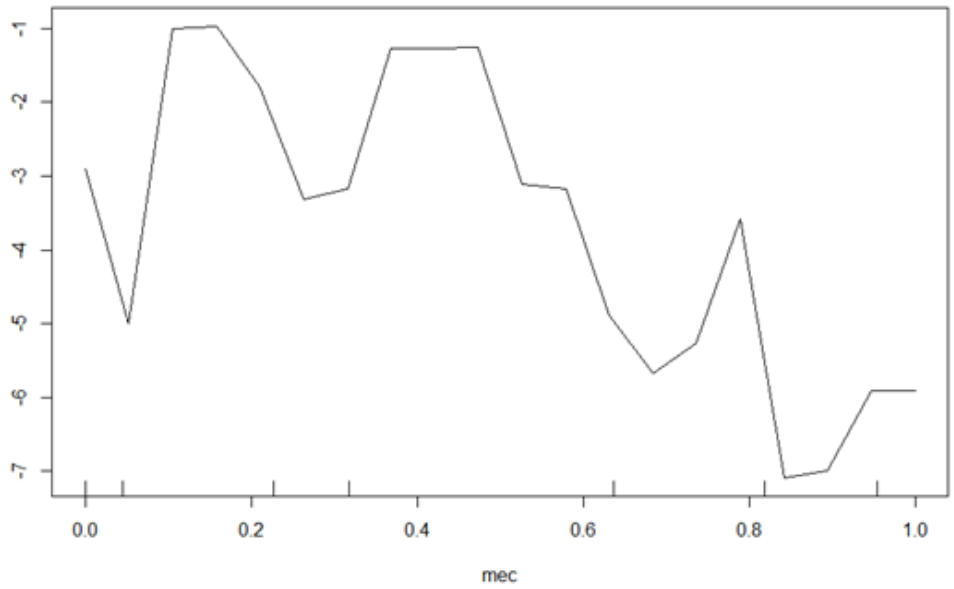
Cinsiyet değişkeni için kısmi bağımlılık grafiği;



Gün değişkeni için kısmi bağımlılık grafiği;



Mecra değişkeni için kısmi bağımlılık grafiği;



Kreatif değişkeni için kısmi bağımlılık grafiği;