T.C.
MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK ANABİLİM DALI

AĞAÇ TEMELLİ MAKİNE ÖĞRENMESİ YÖNTEMLERİNİN KARŞILAŞTIRILMASI VE HASTALIK TANISI İÇİN UYGULANMASI

YÜKSEK LİSANS TEZİ

YUNUS EMRE CEYLAN

HAZİRAN 2021
MUĞLA

Y.E. CEYLAN İSTATİSTİK ANABİLİM DALI YÜKSEK LİSANS TEZİ MUĞLA 2021

T.C.
MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK ANABİLİM DALI

AĞAÇ TEMELLİ MAKİNE ÖĞRENMESİ YÖNTEMLERİNİN KARŞILAŞTIRILMASI VE HASTALIK TANISI İÇİN UYGULANMASI

YÜKSEK LİSANS TEZİ

YUNUS EMRE CEYLAN

HAZİRAN 2021
MUĞLA

MUGLA SITKI KOÇMAN ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

TEZ ONAYI

YUNUS EMRE CEYLAN tarafından hazırlanan **AĞAÇ TEMELLİ MAKİNE ÖĞRENMESİ YÖNTEMLERİNİN KARŞILAŞTIRILMASI VE HASTALIK TANISI İÇİN UYGULANMASI** başlıklı tezinin, 25/06/2021 tarihinde aşağıdaki jüri tarafından İstatistik Anabilim Dalı'nda yüksek lisans derecesi için gerekli şartları sağladığı oybirliği/oyçokluğu ile kabul edilmiştir.

TEZ SINAV JURİSİ

Dr. Öğr. Üyesi Aytaç PEKMEZCİ **(Jüri Başkanı)** İmza:

İstatistik Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla _____

Doç. Dr. Eralp DOĞU **(Danışman)** İmza:

İstatistik Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla _____

Dr. Öğr. Üyesi Elvan Hayat AKTÜRK **(Üye)** İmza:

Ekonometri Anabilim Dalı, Ege Üniversitesi, İzmir _____

ANA BİLİM DALI BAŞKANLIĞI ONAYI

Prof. Dr. Dursun AYDIN İmza:

İstatistik Ana Bilim Dalı Başkanı,
Muğla Sıtkı Koçman Üniversitesi, Muğla _____

Doç. Dr. Eralp DOĞU İmza:

Danışman, İstatistik Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla _____

Savunma Tarihi: 25/06/2021

Tez çalışmalarım sırasında elde ettiğim ve sunduğum tüm sonuç, doküman, bilgi ve belgelerin tarafımdan bizzat ve bu tez çalışması kapsamında elde edildiğini; akademik ve bilimsel etik kurallarına uygun olduğunu beyan ederim. Ayrıca, akademik ve bilimsel etik kuralları gereği bu tez çalışması sırasında elde edilmemiş başkalarına ait tüm orijinal bilgi ve sonuçlara atıf yapıldığını da beyan ederim.

Yunus Emre CEYLAN

25/06/2021

ÖZET

AĞAÇ TEMELLİ MAKİNE ÖĞRENMESİ YÖNTEMLERİNİN KARŞILAŞTIRILMASI VE HASTALIK TANISI İÇİN UYGULANMASI

Yunus Emre CEYLAN

Yüksek Lisans Tezi

Fen Bilimleri Enstitüsü

İstatistik Anabilim Dalı

Danışman: Doç. Dr. Eralp DOĞU

Haziran 2021, 78 sayfa

Bu çalışmanın amacı, tiroit hastalığının teşhisinde ağaç temelli makine öğrenmesi algoritmalarının kullanımı potansiyelinin araştırılmasıdır. Bu amaçla, Muğla Sıtkı Koçman Üniversitesi Tıp Fakültesi Hastanesi'nde toplanan kronik otoimmün tiroit veri seti ve University of California Irvine Machine Learning Repository sitesinden çekilen açık veri seti kullanılmıştır. Dengesiz dağılım gösteren kronik otoimmün tiroit veri setine sentetik azınlık aşırı örnekleme tekniği (SMOTE) uygulanmıştır. Veri setlerine beş farklı sınıflandırma yöntemi uygulanmıştır. Uygulanan sınıflandırma yöntemleri C5.0 karar ağacı algoritması, CART karar ağacı algoritması, CTREE karar ağacı algoritması, rastgele orman ve xgboost yöntemleridir. Uygulamada açık kaynak kodlu R programlama dili kütüphaneleri kullanılmıştır. Sınıflandırma yöntemlerinin performanslarını değerlendirmek için karmaşıklık matrisinden faydalanılmış ve karmaşıklık matrisi yardımı ile her bir model için doğruluk, doğruluk için güven aralığı, hassasiyet, kesinlik, özgüllük, F skor değeri ve kappa istatistiği değerleri hesaplanmıştır. Bu sonuçların ışığında en iyi sonucu veren teknikler belirtilmiştir. Ayrıca modeller karar kuralları açısından değerlendirilmiştir. Böylece, hem yüksek performans gösteren hem de yorumlaması basit olan yöntemler incelenmiştir.

Anahtar Kelimeler: Makine Öğrenmesi, Karar Ağacı, Rastgele Orman, Aşırı Gradyan Arttırma(XGBOOST), Açık Kaynak Kodlu R Programlama Dili

ABSTRACT

COMPARISON OF TREE-BASED MACHINE LEARNING METHODS AND ITS APPLICATION TO DIAGNOSIS

Yunus Emre CEYLAN

Master of Science (M.Sc.)

Graduate School of Natural and Applied Sciences

Department of Statistics

Supervisor: Assoc. Prof. Dr. Eralp DOĞU

June 2021, 78 pages

The aim of this study is to investigate the potential of using tree-based machine learning algorithms in the diagnosis of thyroid disease. For this purpose, the chronic autoimmune thyroid dataset collected at Muğla Sıtkı Koçman University Medical Faculty Hospital and the open dataset from the University of California Irvine Machine Learning Repository were used. Synthetic minority oversampling technique (SMOTE) was applied to the imbalanced chronic autoimmune thyroid dataset. Five different classification methods were applied to the data sets. The applied classification methods are C5.0 decision tree algorithm, CART decision tree algorithm, CTREE decision tree algorithm, random forest and xgboost methods. Open source R programming language libraries were used in the application. The confusion matrix was used to evaluate the performance of the classification methods based on accuracy, confidence interval for accuracy, sensitivity, precision, specificity, F score value and kappa statistics. In the light of these results, the techniques that gave the best results are specified. In addition, the models were evaluated in terms of decision rules. Thus, methods that are both high-performing and simple to interpret were examined.

Keywords: Machine Learning, Decision Tree, Random Forest, Extreme Gradient Boosting (XGBOOST), Open Source R Programming Language

ÖNSÖZ

Tez çalışma sürecinde tüm bilgi ve deneyimleriyle yanımda olan değerli danışman hocam Doç. Dr. Eralp DOĞU'ya teşekkür ederim.

Tezin veri toplama aşamasında destek olan Doç. Dr. Gülhan AKBABA hocama teşekkür ederim.

Lisans ve yüksek lisans sürecinde daima yanımda olan ve akademik camiaya adım atmam için beni destekleyen kıymetli hocalarım Kazakistan Ahmet Yesevi Üniversitesi Öğretim Görevlisi Eren AKDENİZ ve Tokat Gaziosmanpaşa Üniversitesi Öğretim Görevlisi Cemil GÜNDÜZ'e teşekkür ederim.

Tez çalışma sürecinde maddi manevi her türlü desteğiyle yanımda olan değerli aileme sonsuz teşekkür ederim.

Hangi koşulda olursa olsun yanımda olan değerli dostlarıma hoşgörülerinden dolayı teşekkür ederim.

İÇİNDEKİLER

ÇİZELGELER DİZİNİ

ŞEKİLLER DİZİNİ

KISALTMALAR VE SEMBOLLER DİZİNİ

φ:Phi.

θ: Teta.

γ: Gama.

λ: Lamda.

σ (Sigma): Standart sapma.

μ (Mü): Normal dağılım ortalamasıdır.

k-nn: K en yakın komşu.

USG: Genel ultrason görüntüsü.

USGboyutsağ1: 1. sağ ultrason görüntüsü.

USGboyutsağ2: 2. sağ ultrason görüntüsü.

USGboyutsağ3: 3. sağ ultrason görüntüsü.

USGboyutsol1: 1. sol ultrason görüntüsü.

USGboyutsol2: 2. sol ultrason görüntüsü.

USGboyutsol3: 3. sol ultrason görüntüsü.

TSH: Tiroit uyarıcı hormonudur.

sT3: Tiroit bezinin triiyodotironin hormonudur.

sT4: Tiroit bezinin tiroksin hormonudur.

ATG: Anti Tiroglobulin tiroit bezi tarafından üretilen proteindir.

ATPO: Tiroit bezinin ürettiği antikora anti TPO adı verilir.

LAP: Lenf bezi büyümesi bir diğer adı ile Lenfadenopati adı verilir.

DTSH: Tiroit fonksiyon testi.

DP: Doğru pozitif sınıftır.

DN: Doğru negatif sınıftır.

YP: Yanlış pozitif sınıftır.

YN: Yanlış negatif sınıftır.

UCL: University of California Irvine Machine Learning Repository.

CAT: Kronik otoimmün tiroit.

1. GİRİŞ

Bilgisayar teknolojilerindeki gelişmeler ve bilgisayar donanımının ucuzlaması, kamu kurum ve kuruluşlarında, fabrikalarda, hastanelerde bilgisayar vazgeçilmez olmuştur. Teknolojinin gelişmesiyle birlikte veri toplanması ve depolanması basitleşmiştir. Kurumlar, devletler ve kişiler tarafından çok sayıda veri toplanmaktadır. Bilgisayarlar büyük boyutlu verilerin depolanabilmesine olanak tanımıştır. Bu veriler genellikle elektronik ortamlarda depolanmaya başlamıştır. Telefon, tablet ve bilgisayar gibi cihazlarla yapılan işlemler kayıt altına alınmaktadır. Büyük veri tabanlarında saklanan bu verilerin kullanımı ile veri tabanlarında bilginin keşfi kavramı ortaya çıkmıştır. Makine öğrenmesi, istatistiksel metotlar ile bilgisayar algoritmalarını kullanarak depolanan bu veri kümelerini anlamlandıran yani veri tabanında toplanan verilerin çöp olmamasını sağlayan süreçtir. Makine öğrenmesi yöntemleri düzenlenmiş veri setlerinin analizi için uygun hale getirilmiştir. Tüm bu gelişmelerle makine öğreniminin popülerliğini arttırmıştır. Makine öğrenmesi tekniklerinin yoğun olarak kullanıldığı alanlar ise şu şekildedir: Hastalık teşhisi, dolandırıcılık tespiti, kredi başvurusunun değerlendirilmesi, DNA dizinlerinin sınıflandırılması, insansız hava aracı… gibi pek çok örnek verilebilir (Sevli O., 2019). Sağlık sektöründe tanı ve teşhisin sadece insan gücüyle yönetilmesi mümkün değildir. Hastalıkların teşhis, tedavi ve rehabilitasyonunu içeren sağlık hizmetlerinin yürütülebilmesi için sağlık yönetimi önem kazanmaktadır. Özellikle sağlık alanında son zamanlarda makine öğreniminin kullanımı artmıştır. Bilim insanları tiroit hastalığının teşhisinde makine öğrenmesi yöntemlerini de kullanmaktadır. Tiroit, halk arasında "adem elması" olarak bilinen insan vücudunda boyun bölgesinin altında bulunur. Kendini belli eden bu yapı yutkunma ile birlikte aşağı yukarı hareket eder. Tiroit bezi, T3 (triiyodotironin) ve T4 (tiroksin) hormonlarını üreterek kan dolaşımı ile birlikte vücut metabolizmasını düzenler (Kaba M., 2013). Bu çalışmada ağaç temelli makine öğrenmesi yöntemleri ile tiroit hastalığının teşhisi ve sınıflandırması yapılacaktır.

1.1. Amaç

Bu çalışmanın üç amacı vardır;

- Ağaç temelli makine öğrenme yöntemlerinin araştırılması,
- Ağaç temelli makine öğrenmesi yöntemlerinin açık kaynak kodlu R programlama dilinde uygulanması,
- Karar kurallarının incelenmesi.

Bu kapsamda Muğla Sıtkı Koçman Üniversitesi Eğitim ve Araştırma Hastanesinde toplanan kronik otoimmün tiroit veri seti ve University of California Irvine Machine Learning Repository sitesinden elde edilen açık veri seti ile uygulamalar yapılmıştır.

Veri setlerinde tiroit hastalığına ilişkin nitelikler mevcuttur. Kronik otoimmün veri seti 16 nitelik UCL verisi ise 6 nitelikten oluşmaktadır.

Kronik otoimmün tiroit veri setinde dengesiz sınıf problemi olduğu için veri ön işlemeye tabi tutulmuştur. Daha sonra kronik otoimmün tiroit veri setinden bazı nitelikler çıkarılarak yeni bir veri seti oluşturulmuştur. Kronik otoimmün tiroit veri setine makine öğrenme teknikleri uygulanmıştır. Makine öğrenmesi tekniklerinden C5.0 karar ağacı algoritması, sınıflandırma ve regresyon ağaçları (CART), koşullu çıkarım ağacı (CTREE), rastgele orman ve aşırı gradyan arttırma yöntemleri uygulanmıştır. UCL verisinde dengesiz sınıf problemi olmadığı için veri setine herhangi bir ön işleme yapılmadan yöntemler uygulanmıştır. Veri setlerine uygulanan makine öğrenmesi tekniklerine ilişkin karar analizleri yapılmıştır.

2. MAKİNE ÖĞRENMESİ

2.1. Makine Öğrenmesi

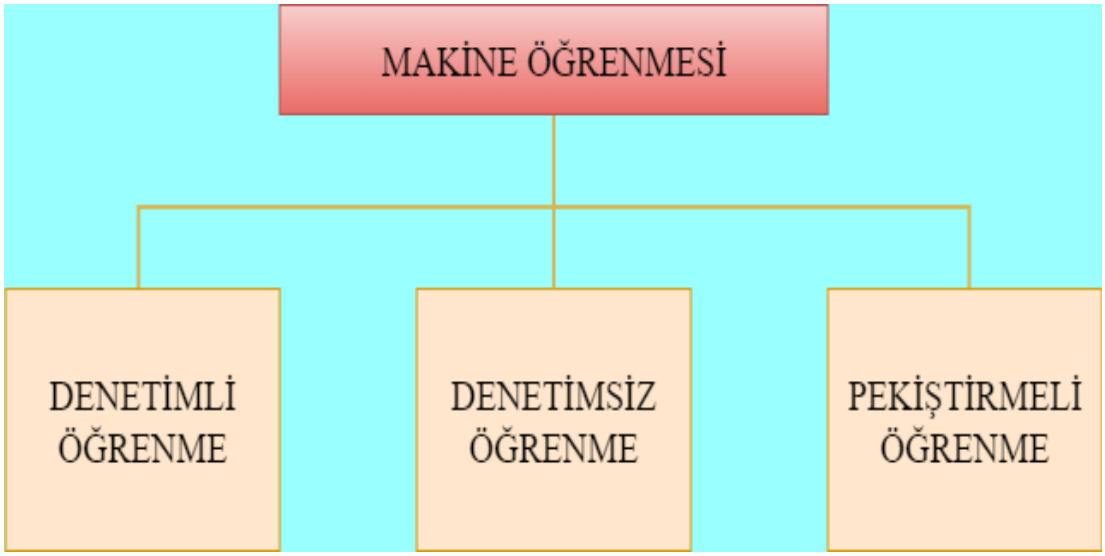
Bilgisayarın keşfinden 1950'lere kadar sadece insanlığa kolaylık sağlaması için belli görevleri yapan bilgisayarlar gün geçtikçe teknolojinin gelişmesi ile birlikte yapılan işler hakkında istenilen verileri toplayıp depolanması sağlandı. Bilgisayarlarda toplanan bu veriler anlamlandırılmadıktan sonra hiçbir şey ifade etmeyip bir çöp yığınına dönüşüp kalıyordu. Makine öğrenmesi ile bilgisayarların da insanlar gibi eğitilip öğretilerek belli bir seviyeye ulaştırılması gerekli olmuştur. Bu nedenle "Gözlemler sonucu elde edilen verileri bilgisayar yardımıyla işlenerek, bu gözlemlerin içinde yer alan örüntüler keşfedilip tanımlanabilir mi?" sorusu makine öğrenmesinin temelini oluşturmaktadır (Emir Ş., 2013).

Amerikalı bilgisayar bilimci Arthur Samuel'in 1959'da bir araya getirdiği "makine öğrenmesi" ifadesi, yapısal olarak öğrenebilen ve veriler üzerinde anlamlı tahminler yapabilen bilgisayar algoritmalarının genel adıdır. Arthur Samuel IBM'de çalışırken makine öğrenmesi ile bilgisayarın oynayabildiği dama oyununu geliştirmiştir (Samuel A., 1959).

Makine öğrenmesi; çıktı değerlerini kabul edilebilir bir aralıkta tahmin etmek için girdi verilerini alan ve analiz eden programlanmış algoritmalar kullanır. Bu algoritmalara yeni veriler gönderilirken, performansı iyileştirmek ve zamanla 'zekâ' geliştirmek için operasyonları öğrenir ve optimize ederler (Bayer H. ve Çoban T., 2015).

Makine öğrenimi algoritmaları geniş sınıflandırması, denetimli öğrenme (supervised learning), denetimsiz öğrenme (unsupervised learning) ve pekiştirmeli öğrenme (reinforcement learning) öğrenme olmak üzere üç kategoride yapılır(Sharma Nv.d , 2021a).

Makine öğrenimi yaklaşımları şekil 1'de aşağıdaki gibi gösterilir.



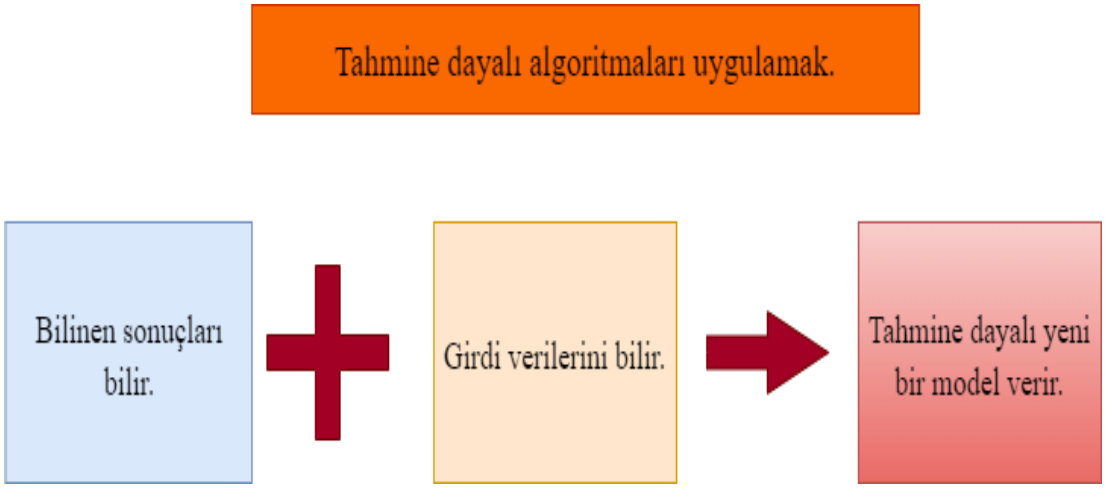
Şekil 1. Makine öğrenimi yaklaşımları

2.1.1. Denetimli öğrenme (Supervised learning)

Denetimli öğrenme, daha sonra tahminlerde bulunabilecek bir model oluşturmak için tanımlanan veri kümelerini kullanan bir tür makine öğrenimi yöntemidir (Barnes J., 2015).

Denetimli öğrenme, tanımlanan veri kümelerini eğitim ve test veri kümelerine ayırır. Eğitim veri kümelerinden, denetimli öğrenme algoritmaları, bilinen sonuçlarla birlikte yeni girdi değerlerine dayalı tahminler yapabilen yeni bir model oluşturmaya çalışır (Sharma N.v.d, 2021b) .

Algoritmalar, öğrendiklerinden yola çıkarak tahmin yapmak için etiketli verileri kullanır. Yani eğitimde kullanılacak veri ve veriye ait sınıflar (kategoriler/etiketler) önceden bilinir. Bu bilgi ile sistem öğrenir ve yeni gelen veriyi bu öğrendikleriyle yorumlar. Denetimli öğrenmeye ilişkin akış şeması şekil 2’ de verilmiştir.



Şekil 2. Denetimli öğrenme akış şeması

Denetimli öğrenme algoritmaları sınıflandırma ve regresyon amacı ile kullanılır. Çalışmada sınıflandırma yöntemleri kullanılacağı için sınıflandırma yöntemleri ele alınmıştır.

2.2. Sınıflandırma Yöntemleri

Literatürü incelendiğinde en sık kullanılan makine öğrenmesi yöntemlerinin sınıflandırma yöntemleri olduğu görülmektedir. Sınıflandırma yöntemi geçmiş verilerin ait olduğu sınıfları değerlendirerek, yeni gelen verilerin hangi sınıfa ait olduğunun tahmin edilmesidir (Kaya T., 2015).

Sınıflandırmaya örnek verilecek olursa şeker hastalığının teşhisi için geçmişteki veriler temel alınarak deneklerin aç karnına kandaki şeker oranı (Glikoz) 70-100 mg arası olursa hastalık yok olarak sınıflanmıştır. Glikoz miktarı 100 mg üzerinde olursa hastalık var şeklinde sınıflanmıştır. İki kategorili sınıflandırma işlemine göre bir sonraki deneğin aç karnına vermiş olduğu kan tahlili sonucuna göre kandaki glikoz miktarı 100 mg'dan yüksek çıkarsa algoritmalar deneği hastalık var şeklinde sınıflandırır.

Literatürde sınıflandırma işlemi için çok fazla yöntem bulunmaktadır. Bazıları şunlardır; Karar Ağaçları (Decision Trees), Rastgele Orman (Random Forest), Naive

Bayes, Destek Vektör Makineleri (Support Vector Machines), K-en Yakın Komşu (K-Nearest Neighbor) algoritmalarıdır.

2.2.1. Karar ağaçları (Decision trees)

Karar ağaçları bir dizi ikili sınıflandırma yoluyla veri setinin birkaç alt gruba bölündüğü diyagramlardır (Namazkhan M. v.d., 2020).

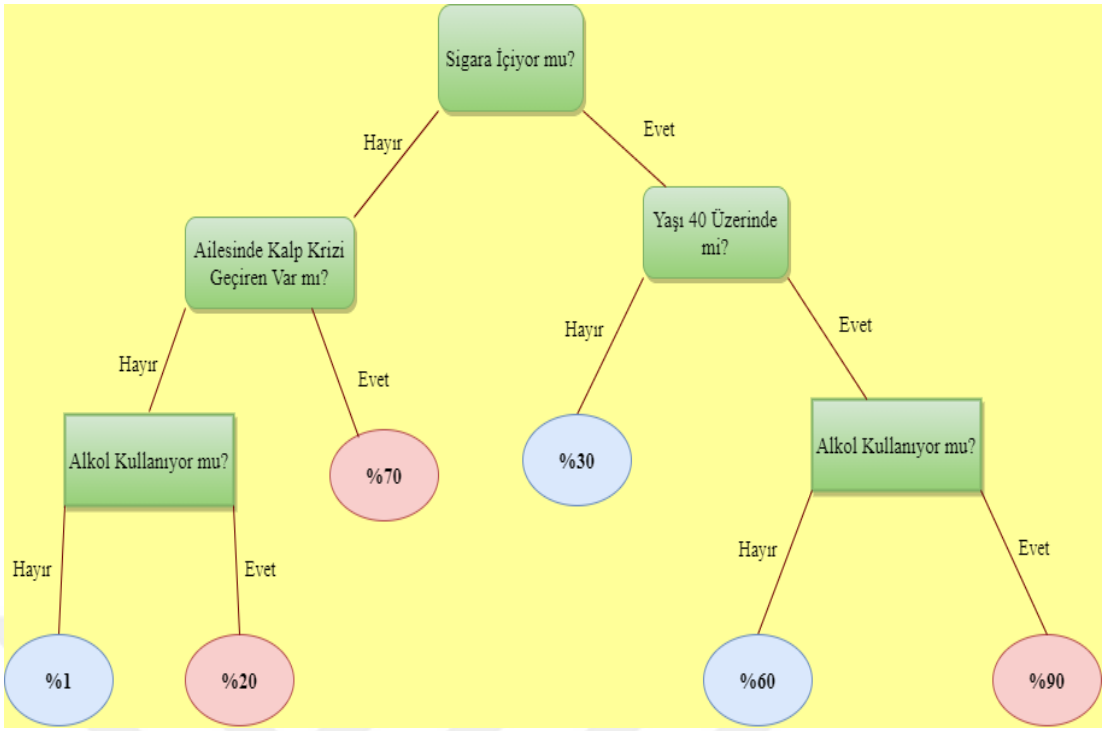
Karar ağaçları yorumlanmaları kolay olması ve veri setleri ile uygulanması kolay olduğu için yaygın olarak tercih edilen bir sınıflandırma yöntemidir (Alan A., 2020).

Karar ağaçlarının yapısı, aynı bir ağaç gibi kök düğüm, dal ve yapraklardan oluşmaktadır. Yaprak kısmında oluşan değer çıktı olarak adlandırılmaktadır ve araştırılan problem sınıflandırma problemi ise sınıf etiketini alır. Regresyon problemi ise, sayısal bir değeri almaktadır. Karar ağacı, kullanılan verinin durumundan etkilenmektedir. Eğer veri seti karmaşık ise ağaç dallanıp büyümektedir. Karar düğümü seçim kriteri tanımlanan her nitelik için bir sıralama sağlamakta ve hangi niteliğin tercih edileceğine karar verir. Hangi düğümün seçileceğini belirleyen ölçütler ise kazanım değeri, kazanım oranı ve gini indeksi ile belirlenir. Veri setine uygulanan yöntem sonucu birden fazla ağaç elde edilmesine rağmen en küçük boyutlu ağaç seçilir. Değişkenin belirlenmesi aşamasında algoritmanın karar ağacı modelinde karmaşıklık yaşamaması için bulunduğu düğümdeki bütün öğeleri aynı sınıfta yer alması gerekmektedir. Yapraklarda öğelerin hepsi aynı sınıfta yer alacağından sınıflandırma yapacak değer kalmayacağından dolayı karar ağacı modelindeki döngü durur ve en uygun karar ağacı modeli oluşmuş olur (Başer Ö. B. , v.d, 2021).

Karar ağacı bileşenleri;

Karar ağacının bileşenleri farklı geometrik semboller kullanılarak temsil edilir. Karar problemindeki karar noktaları karar düğümü olarak adlandırılan kare şekli ile belirtilir. Doğal durumlar daire şekli ile gösterilir ve şans düğümü olarak adlandırılır. Düğümleri birbirine bağlayan çizgiler dal olarak ifade edilir. Karar düğümünden çıkan çizgiler karar dalı ve şans düğümünden çıkan çizgiler şans dalı olarak tanımlanır. Sonuç değerleri ise bitiş düğümünde gösterilir (Lezki Ş., 2014).

Karar ağacının yapısı kalp krizi geçiren kişilere ilişkin varsayımsal bir veri üzerinden değişkenler ve sonuçlar ile birlikte şekil 3'de gösterilmiştir.



Şekil 3. Örnek karar ağacı diyagramı

Şekil 3 incelendiğinde sigara içen kişi 40 yaşı üzerinde ise ve alkol kullanıyorsa kalp krizi geçirme riski %90'dır. Sigara içiyor ve alkol kullanmıyorsa %60 olasılık ile kalp krizi geçirme riski vardır. Sigara içmiyor ailede kalp krizi geçiren varsa %70 olasılık ile kalp krizi geçirme riski vardır. Ailede kalp krizi geçiren yoksa bir başka sınıflama adımı alkol kullanımı olmaktadır. Denek alkol kullanmıyor ise kalp krizi geçirme olasılığı %1'dir. Alkol kullanılıyorsa %20 olasılık ile kalp krizi geçirme riski vardır. Şekilde karar düğümleri (decision node) dikdörtgenler ve yaprak düğümleri ise (leaf node) dairelerdir. Her bir yaprak, en uygun hedef değeri temsil eden bir sınıfa atanır. Alternatif olarak, yaprak, hedef özniteliğin olasılığı olacak şekilde belli bir değere sahip olasılık vektörüne sahip olabilir. Örnekler, ağacın kökünden yaprağa kadar bu örnekleri dolaşarak, yol boyunca testlerin sonucuna göre sınıflandırılır.

Örneklerden oluşan bir küme kullanılarak karar ağacının oluşturulmasını sağlayan çok sayıda öğrenme yöntemi vardır. Her farklı ölçüt için bir karar ağacı algoritması karşılık gelmektedir.

2.2.1.1. Karar ağacı algoritmaları

Karar ağaçları oluşturulurken kullanılan algoritmanın hangisi olduğu önemlidir. Çünkü kullanılan algoritmaya göre oluşturulan ağacın şekli değişebilir. Değişik ağaç yapıları da farklı sınıflandırma sonuçları verir. İlk temelleri AID yöntemi ile atılan karar ağacı yöntemleri çeşitli algoritmalar ile sürdürülmüştür. Başlıca karar ağacı algoritmaları ise şu şekildedir; Entropiye dayalı ağaçlar (ID3 ve C4.5) koşullu ağaç (CTREE), C5.0 karar ağacı ile sınıflandırma ve regresyon ağacı (CART) algoritmaları biçiminde birçok yöntem geliştirilmiştir (Haciefendioğlu Ş., 2012).

2.2.1.2. Bilgi kazancı (Entropi)

Bilgi kazancı ID3 ve C4.5 algoritmalarında eğitim veri setinde karar ağacının en üstteki değişkeninin belirlenmesini sağlar. Karar ağaçlarının oluşturulması sırasında dallanmaya hangi nitelikten başlanılacağı önem taşımaktadır. Algoritmalar bilgi kazancının alacağı değere göre ağacın dallanmasını sağlar. Bilgi kazancı sistemdeki belirsizliği ve beklenmeyen durumun ortaya çıkma olasılığını gösterir. S bir kaynak olsun. Bu kaynağın $\{m_1, m_2, \dots, m_3\}$ olmak üzere n mesaj üretebildiğini varsayalım. Tüm mesajlar birbirinden bağımsız olarak üretilmektedir ve m_i mesajının üretilme olasılıkları p_i'dir. $P = \{ p_1, p_2, p_3, .., p_n \}$ Olasılık dağılımına sahip mesajları üreten S kaynağının bilgi kazancı (S) aşağıda verildiği gibidir (Farboudi S., 2009a).

Bilgi kazancı denklem (2.1)'deki gibi bulunur;

$$Bilgi\ (S) = -\sum_{i=1}^{n} \frac{(Ci,T)}{|T|} * \log 2 \left(\frac{(Ci,T)}{|T|}\right) \quad (2.1)$$

Örneğin P hedeflenen sınıfta bulunsun. ($p1, p2, \dots, pm$) toplamları 1 olan olasılıklardır.

- Örnekler aynı sınıfa ait ise $bilgi\ kazancı = 0$
- Örnekler sınıflar arasında eşit dağılmışsa $bilgi\ kazancı = 1$
- Örnekler sınıflar arasında rastgele dağılmışsa $0 < bilgi\ kazancı < 1$ aralığındadır.

Veri tabanından eğitim için elde edilen eğitim kümesini ele alalım. Eğitim kümesi sınıf niteliğinin alacağı değerlere göre $\{C_1, C_2, \dots, C_i\}$olmak üzere i sınıfa ayrıldığını varsayalım. Bu sınıflarla ilgili olarak ortalama bilgi miktarına ihtiyaç duyulabilir.

T sınıf sayılarının olduğu kümedir. P_T sınıfların olasılık dağılımıdır ve denklem (2.2)'deki gibi hesaplanmaktadır (Farboudi S., 2009b).

$$P_T = (p1, p2, \dots, pk) = \left(\frac{|C_1|}{|T|}, \frac{|C_2|}{|T|}, \dots, \frac{|C_i|}{|T|}\right) \quad (2.2)$$

$|Ci|$: C_isınıfındaki elemanların sayısıdır.

T için ortalama bilgi kazanımı ise şöyledir;

$$bilgi(T) = -\sum_{i=1}^{n} P_i * log2(P_i) \quad (2.3)$$

2.1.1.3. Gini indeksi

Gini indeksi, CART karar ağacı oluşturulması sırasında kullanılır. CART karar ağacında dallanmaya ilk hangi nitelikten başlanacağı ve dallanma değeri gini indeksine göre gerçekleşir (Adak F. M. ve Yurtay N., 2013).

Nitelik değerlerinin sağda ve solda olmak üzere iki bölünme esasına dayanan gini indeksi aşağıdaki gibi hesaplanır:

Her nitelik değerleri ikili gruplandırılır. Bu şekilde elde edilen sol ve sağ bölünmelere karşılık gelen sınıf değerleri gruplandırılır. Her bir nitelikle ilgili sol ve sağ taraftaki bölünmeler için $Ginisol$ ve $Ginisağ$ değerleri denklem (2.4) ve denklem (2.5)'deki gibi hesaplanır.

$$Gini_{sol} = 1 - \sum_{i=1}^{n} \left(\frac{L_i}{|t_{sol}|}\right)^2 \quad (2.4)$$

$$Gini_{sağ} = 1 - \sum_{i=1}^{n} \left(\frac{R_i}{|t_{sağ}|}\right)^2 \quad (2.5)$$

T:Bir düğümdeki örnekler

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

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

Li: Sol tarafta i kategorisindeki örneklerin sayısı

Ri: Sağ tarafta i kategorisindeki örneklerin sayısı

Her j niteliği için, n eğitim kümesindeki satır sayısı olmak üzere aşağıdaki bağıntının değeri denklem (2.6)'da hesaplanır.

$$Gini_j = \frac{1}{n}(\left|T_{sol}\right| Gini_{sol} + \left|T_{sağ}\right| Gini_{sağ}) \quad (2.6)$$

Her j niteliği için hesaplanan $Gini_j$ değerleri arasından en küçük olanı seçilir ve bölünme bu nitelik üzerinden gerçekleştirilir. Kalan veri setine yukarıdaki adımlar tekrar uygulanır ve bölünme hesaplanır (Farboudi S., 2009c).

2.2.1.1.1. ID3 karar ağacı algoritması

J. Ross Quinlan tarafından 1983 yılında geliştirilen basit bir karar ağacı algoritmasıdır. ID3'te amaç, veri setinde çok fazla nitelik varsa ve eğitim verisi çok fazla kayıt içeriyorsa bilgi kazancı ile fazla hesaplama yapmadan en uygun ağacı oluşturmaktır (Peng W. v.d., 2009).

ID3 algoritmasının adımları ise aşağıda verilmiştir:

- C bir eğitim kümesi olmak üzere, eğer C'deki bütün kayıtlar aynı sınıf üyesi iseler, sınıfın adında bir düğüm oluşturulur ve algoritma sonlanır karar düğümü oluşturulur.
- C kümesi, karar düğümüne göre alt kümelere ayrılır: $C_1, C_2, C_3, \dots, C_n$.
- Algoritma her bir C_i kümesine özyinelemeli bir şekilde uygulanır (Koçdağ Y. M., 2016).

2.2.1.1.2. C4.5 karar ağacı algoritması

1993 yılında J. Ross Quinlan ID3 algoritmasının gelişmiş versiyonu olan C4.5 karar ağacı algoritmasını oluşturmuştur. C4.5 karar ağacı algoritması ID3'deki sayısal sonuçları vermemesi gibi eksiklerin ortadan kaldırması için tasarlanmıştır (Singh S. ve Gupta P., 2014a).

C4.5 karar ağacı algoritması iki adımda gerçekleşir. Bu adımlardan biri ağaç oluşturma işlemi diğer adım ise budama işlemidir. Budama işlemi nitelik sayısı fazla olan veriler için sınıflandırma doğruluğunu artırmak veya o yaprak tarafından tahmin edilen sınıfa ait olmayan örnekleri sınıflandırmak için kullanılır. Karar ağacı oluşturulduktan sonra budama işlevi başlar. Ağaç düğümlerini kontrol eder ve istenmeyen düğümleri yaprak düğümlerle değiştirerek dalları azaltmaya çalışır (Gümüşçü A. v.d, 2016).

C4.5 karar ağacı algoritmasının avantajları şu şekilde açıklanabilir;

Hem kategorik hem de sayısal veri setleri kullanılabilir. Sürekli nitelikleri işlemek için "*t*" eşlik değeri oluşturur. Eşlik değerinin üstünde olanlara, ona eşit ve ondan küçük olanlara göre böler. Eğitim verilerinde eksik öznitelik değerlerinin bulunması durumunda uygulanabilir. Çünkü eksik nitelikler için bilgi kazancı kullanmaz. Eğitim veri setinin istenmeyen değerleri elimine edilebilir. Aşırı uygunluk problemini ortadan kaldırılır.

C4.5 karar ağacı algoritmasının dezavantajları ise şu şekilde açıklanabilir;

Boş dallar oluşturur. Sıfır değerli ya da sıfıra yakın çok sayıda düğüm oluşur. Bu durumlar ağacın daha büyük olması ve karmaşık olmasına sebep olur. Algoritma modeli olağandışı özelliklere sahip verileri aldığında aşırı derecede uydurma ağaç oluşturur (Singh S. ve Gupta P., 2014b).

2.2.1.1.3. C5.0 karar ağacı algoritması

C4.5 tarafından üretilen ağaçlar hem küçük hem de doğrudur, bu da hızlı, güvenilir sınıflandırıcılar sağlar. C4.5 karar ağacı algoritması ID3 karar ağacı algoritmasının kurallarını takip eder. Benzer şekilde C5.0 karar ağacı algoritması da C4.5 karar ağacı algoritmasının kurallarını takip eder. Kısaca açıklamak gerekirse C5.0 karar ağacı algoritması C4.5 karar ağacı algoritmasının gelişmiş halidir (Siknun P. G. ve Sitanggang S. I., 2016).

C5.0 karar ağacı türetme algoritması, tek bir düğümle başlamakta ve en uygun sınıfın belirlenmesi için bilgi kazancına dayalı bir ölçü kullanmaktadır. C5.0 karar ağacı algoritması kök düğümden yaprak düğüme kadar karar kurallarını verir. C5.0 karar ağacı C4.5 karar ağacına göre daha hızlıdır. Bellek kullanımı C4.5'den daha verimlidir. C4.5'e göre daha küçük karar ağaçları oluşturur ve hata oranı daha

düşüktür. C4.5 ile karşılaştırıldığında doğruluğu daha iyi sonuç verir. C5.0 karar ağacı algoritması kullanılmayacak olan nitelikleri otomatik olarak kaldırır (Pandya R. ve Pandya J., 2015).

C5.0 karar ağacı algoritması veri kümesinde karar düğümlerini sınıflara, kategorilere ayırmayı ve bölmeyi amaçlar. C5.0 uygulaması sonucunda deneme sayısı (trials) ve model tipi gibi ayar metrikleri ile beraber gelir (Gottipati S. v.d, 2018).

C5.0 karar ağacı hiper-parametreleri aşağıda verilmiştir.

Denemeler (Trials) C5.0 yönteminde artırma yinelemelerinin sayısını belirten bir tam sayıdır. Sonucu yöntemde en uygun model için denenen model sayısını verir.

2.3.1.1.4. Sınıflandırma ve regresyon ağaçları (CART)

CART (Classification and Regression Trees) sınıflandırma ve regresyon ağaçları 1984 yılında Breiman tarafından oluşturulmuştur (Gordon v.d, 1984).

CART karar ağacı algoritması tüm veri setinden başlayarak her tekrarlanan bölünme için iki alt düğüm oluşturmak amacıyla tüm nitelikleri tahmin olarak kullanır ve veri setinin alt kümelerini bölerek oluşturulur. En iyi düğümü seçmek için gini indeksini kullanır. Gini indeksi her kök düğüm için kendi ana düğümünden daha sadece olacağı şekilde, her düğümde bir bölünme seçerek çalışır. Daha sonra, tahmin ediciler karşılaştırılır ve sonraki bölünme için en iyi sonucu veren tahminci seçilir. Bu durum durdurma kuralları aktif olana kadar tekrarlanır (Burdur Z. ve Atay E. C., 2018).

CART karar ağacı algoritması aşağıdaki kriterlerin değerlerine göre en uygun bölünme işlemi gerçekleştirmektedir.

t. düğümdeki s. aday bölünmelerinin uygunluk ölçüsü olan $\Phi(s|t)$ olasılığı denklem (2.7)'de hesaplanır,

t_L: t düğümünün sol taraftaki bölünmesi, t_R: t düğümünün sağ taraftaki bölünmesi olmak üzere,

$$\Phi(s/t) = 2P_L P_R \sum_{j=1}^{n} P(j|t_L) - P(j|t_R) \tag{2.7}$$

$$P_L = \frac{t_L'deki\ kayıtların\ sayısı}{Eğitim\ verisindeki\ kayıtların\ sayısı} \tag{2.8}$$

$$P_R = \frac{t_R\text{'deki kayıtların sayısı}}{\text{Eğitim verisindeki kayıtların sayısı}} \quad (2.9)$$

$$P_{(J/t_L)} = \frac{t_L\text{deki j sınıflarının sayısı}}{t_L\text{'deki kayıtların sayısı}} \quad (2.10)$$

$$P_{(J/t_R)} = \frac{t_R\text{deki j sınıflarının sayısı}}{t_R\text{'deki kayıtların sayısı}} \quad (2.11)$$

En uygun bölünme t düğümündeki olası tüm bölünmeler için $\Phi(s/t)$ uygunluk ölçüsü maksimize edilerek gerçekleşir (Gemici E. ve Yakut E., 2017).

2.3.1.1.5. Koşullu çıkarım ağaçları (CTREE)

Koşullu çıkarım ağaçları C5.0 karar ağacı algoritmasına benzer çalışır. Sınıflama niteliğini seçmek için anlamlılık testi tekniklerini kullanır (Gottipati S. v.d, 2018).

Koşullu çıkarım ağaçları (CTREE) permütasyon testlerini kullanarak algoritmanın dağılımsal özelliklerini dikkate alan bir istatistiksel yaklaşım öne sürmektedir. CTREE'de bağımlı değişken ile ortak değişkenler arasındaki ilişkiyi ölçen, istatistiklerin koşullu dağılımı dikkate alınarak, farklı ölçeklerde ölçülmüş ortak değişkenler arasından yansız bir seçim yapılmasını sağlamaktadır (Yabacı A., 2017).

Koşullu ağaç yöntemi için hiper-parametreler şu şekildedir;

Mincriterion ölçeği bölünmeyi uygulamak için aşılması gereken test istatistiği değerini verir. Bu kritere göre, doğruluk ve kappa istatistikleri sonucu en iyi sonucu veren model seçilir.

2.2.2.Torbalama (bagging) sınıflama yöntemi

Torbalama yöntemi Leo Breiman tarafından 1996 yılında bulunmuştur. Ağaç tabanlı sınıflandırıcıları kullanan bir yöntemdir. Torbalama yöntemi hem sınıflandırma istikrarını hem de doğruluğu arttırmak amacıyla kullanılır. Torbalama yöntemi, N çaplı eğitim veri setinden ön yükleme (bootstrap) örnekleme tekniği ile m adet n çaplı yeni eğitim veri setleri oluşturur $(n \leq N)$. Ön yükleme örnekleme yöntemi, eğitim veri setinden yerine koyarak örnekleme tekniği ile x adet örneğin seçilmesinden oluşur. m adet ön yükleme örneklemi $B_1 \ldots B_m$ üretilir ve her bir ön yükleme örneklemi

için C_i gibi m (ağaç sayısı) adet sınıflayıcı oluşturulur. m adet sınıflayıcılar arasında en yüksek doğruluk değerine sahip ağaç sonuç ağacı olarak seçilir. Torbalama yönteminin algoritması ise şu şekildedir;

Girdi: $S =$ Eğitim veri seti, I: İndüktör, $m =$ ön yükleme örneklem sayısı,

Her bir $i = 1$'den $m'ye$ $\{S^1 = Eğitim\ veri\ setinden$ ön $yükleme$ $örneklemi\ C_i = I\ (S^1)\}$, $C(x) = argmax \sum i: C_i(x) = y^1$ $(en\ sık\ tahmin\ etiketi\ y)$. Çıktı: $C'inci$ sınıf (Korkem E., 2013).

2.2.3. Rastgele orman (Random forest)

Rastgele orman, temeli karar ağaçlarına dayanan ve Leo Breiman tarafından geliştirilen bir makine öğrenme yöntemidir. Birden çok karar ağacının birleştirilmesiyle karar ormanı oluşturulur ve buradaki her bir karar ağacından elde edilen tahmin sonuçları birleştirilerek sonuç tahmini yapılır (Şanlıtürk E., 2018).

Rastgele orman sınıflandırma ve regresyon ağaçlarının geliştirilen bir versiyonu olmasının yanı sıra, en başarılı sonuç veren yöntemlerden biridir. Rastgele orman yöntemi sınıflandırma ve regresyon ağaçlarından farklı olarak çok sayıda karar ağacı üretir. Üretilen bu ağaçların kombinasyonu üzerinden yorum yapma imkanı sunar. Rastgele orman yönteminde karar ağaçlarının oluşturduğu yapıya orman denir (Özdemir S., 2018).

Rastgele orman, nitelikler arasından en iyi dalı seçerek her bir düğümü dallara ayırmak yerine, düğümlerde rastgele seçilen nitelikler arasından en iyisini kullanarak her bir düğümü dallara ayırır. Ardından rastgele özellik seçimi kullanılır ve ağaçlar geliştirilir. Geliştirilen bu ağaçlar budanmaz. Bu strateji rastgele ormanın doğruluk oranını yüksek tutar. Bu yöntem aynı zamanda çok hızlıdır, aşırı uyuma karşı dayanıklıdır ve ne kadar istenirse o kadar ağaçla çalışabilir (Güngör O. ve Akar Ö., 2012).

Rastgele orman yönteminde ağaçların sayısı yalnızca iki parametre üzerinden belirlendiği için kullanımı çok kolaydır. Bu parametreler, en iyi dallanmayı belirlemek için her bir düğümde kullanılan niteliklerin sayısı (m) ve geliştirilecek ağaçların sayısı N'dir. Her bir düğümde m nitelikleri tüm nitelikler arasından rastgele olarak seçilir ve bu değişkenler arasından en iyi dal belirlenir. Ayrıca rastgele orman sınıflandırıcısı aykırı değerlere duyarlı değildir (Zeybek M., 2021).

Rastgele ormanın matematiksel tanımlamaları da aşağıdaki gibidir;

Rastgele orman, $\{h(x,\theta_k), k = 1,2,3,\dots,n\}$ biçiminde tanımlanmış ve her $\{\theta_k\}$'nın tanımlanmış bağımsız rastgele vektörlerden meydana geldiği ağaç biçiminde sınıflandırıcı topluluğudur ve x burada girdi elemanıdır.

Verilen oluşturulmamış sınıflandırıcılar $h_1(x), h_2(x), \dots, h_k(x)$ sınıflama topluluğu verildiğinde rastgele vektörü $X\ ve\ Y$ eğitim kümesi üzerindeki farklılıklar fonksiyonu denklem (2.12)'de ifade edilir;

$$mg(X,Y) = av_k I(h_k(X) = Y) - \max_{j \neq Y} av_k I(h_k(X) = j) \tag{2.12}$$

I ( ) burada işlev fonksiyonudur. Farklılıklar ne kadar büyük olursa sınıflandırma da o kadar büyük olur.

$$PE^* = P_{X.Y}(mg(X,Y) < 0) \tag{2.13}$$

Burada X'in alt küme olduğu yerde Y'nin olasılığı X'in olasılığından büyüktür. Y uzayı rastgele ormanda $h_k(X) = h_k(X,\theta_k)$ büyük sayılı ağaç yapısını izler.

Ağaçların sayısı arttıkça $\theta_1' den\ PE^*$'ye kadar birbirine yakınsar.

$$P_{X,Y}(P_\theta(h(X,\theta) = Y) - \max_{j \neq Y} P_\theta(h(X,\theta) = j) < 0) \tag{2.14}$$

Bu sonuçlar rastgele ormanda daha fazla ağaç eklendikçe bu ağaçların uymadığını gösterir. Fakat genelleme hatasının sınırlayıcı bir değerini ürettiğini açıklar (Breiman, 2001).

Rastgele orman yöntemi için hiper-parametreler şu şekildedir;

Mtry rastgele orman yönteminde her düğüm için rastgele örneklenen girdi değişkenlerinin sayısını verir.

2.2.4. Gradyan arttırma (Gradient boosting)

Gradyan arttırma yöntemi, Friedman tarafından 2001 yılında tanıtılan güçlü bir makine öğrenme tekniğidir. Gradyan arttırmanın temel amacı, yüksek oranda tahmini doğruluğa sahip güçlü bir öğrenme modeli oluşturmaktır. Güçlü öğrenme modeli için

temel öğrenme modellerini yinelemeli olarak birleştirmektedir (Yüce T. ve Kabak M., 2021).

Gradyan arttırma, genellikle karar ağaçları ile birlikte kullanılmaktadır. Gradyan arttırma hem regresyon hem de sınıflandırma problemleri için kullanılan bir yöntemdir. Gradyan arttırma matematiksel olarak şu şekilde ifade edilir;

- Öncelikle ilk veri setinden bir karar ağacı oluşturulur ve tahmin ile çıktı arasındaki hata bulunur.
- Veri setinin örnekleri için yeni çıktı değerleri olarak bu hataları kullanır.
- Hatalarla birlikte yeni bir karar ağacı oluşturur ve önceki ağacın hatalarının yeniden oluşması için ağaç eğitilir.
- Önceki çıktı ile yeni çıktı arasındaki hata oranı eşit olana kadar bu döngü devam eder (Yangın G., 2019).

Gradyan arttırma matematiksel olarak aşağıdaki gibi ifade edilir;

Veri kümesine "D" diyelim. Tanımlanan $L_2: R^2 \rightarrow R$ fonksiyonunu gradyan arttırma deneye dayalı riski en aza indirmek için yinelemeli olarak $f: X \rightarrow R$ modelini oluşturur.

$$L(f\ /\ d) = E_D[L(F(x), y)] \tag{2.15}$$

her "t" yinelemede model güncellenir,

$$f^{(t)}(x) = f^{(t-1)}(x) + \epsilon h^{(t)}(x) \tag{2.16}$$

Denklem (2.16)'da $f^{(t-1)}$önceki yinelemede oluşturulmuş modeldir $\epsilon h^{(t)}(x)$ ise hataların öğrenme oranıdır. Temel modellerde genellikle $h^{(t)}$ negatif gradyanı tahmin etmek için kullanılır.

$$-g^{(t)}(x, y) = -\frac{\partial L(y,s)}{\partial s}, s = f^{(t-1)}(x) \tag{2.17}$$

$$h^{(t)} = arg_{h \in K} min E_D[\left(-g^{(t)}(x, y) - h(x)\right)^2] \tag{2.18}$$

Denklem (2.18)'de temel öğrenme $h^{(t)}$, $\Phi^{(t)} \in R^d$ parametreleri ile ilişkilidir. Bu ilişkiyi göstermek için $h^{(t)}(x, \Phi^{(t)})$ ifadesi yazılır. Temel öğrenme olan "H"

kümesini yinelemeli olarak bölen karar ağaçları oluşturulur. Ağacın her bir yaprağına R_j karşılık gelen bölüme y tahmini değeri atanır. Oluşturulan denklem (2.19)'da ifade edilir;

$$h\left(x, \Phi^{(t)}\right) = \sum_{j=1}^{d} \Phi_j^{(t)} 1\,, x \in R_j \tag{2.19}$$

dolayısıyla karar ağacı $\Phi^{(t)}$'nin doğrusal bir fonksiyonudur. Nihai olarak gradyan arttırmada f modeli karar ağaçlarının toplamıdır ve tam modelin gösterimi θ ile ifade edilir.

Sınıflandırma için bir modeli negatif kayıtlama olasılığı ile eğitilirse veri belirsizliği tahmin edilir. x'in y üzerinde koşullu $p(y \,/\, x, \theta)$ dağılımının parametreleri tahmin edilir ve y normal dağılım varsayılır ve negatif logaritmalarının olasılığı alınır. x girdisi verildiğinde f modeli normal dağılımın ortalaması (μ) ve standart sapması (σ) denklem (2.20) ve denklem (2.21)'de tahmin edilir.

$$p\left(y/x\,, \Phi^{(t)}\right) = N(y/\mu^t\,, \sigma^t),\ \{\mu^t, log\ \sigma^t\} = f^t(x) \tag{2.20}$$

$$L(\theta/D) = E_D[-logp(y/x\,, \theta)] = -\frac{1}{N}\sum_{i=1}^{N} logp(y^i/x^i, \theta) \tag{2.21}$$

Burada θ'nın işlevi ortalama ve standart sapmanın logaritmasını tahmin etmek için kullanılan vektörlerin birleşimidir (Malinin A. v.d, 2020).

2.2.5. Aşırı gradyan arttırma (xgboost: Extreme gradient boosting)

Aşırı gradyan arttırma (Xgboost) gradyan arttırma yönteminin çeşitli düzenlemeler ile optimize edilmiş yüksek performanslı halidir. Xgboost 'un orijinal hali 2002 yılında Friedman tarafından sunulmuştur. Daha sonra Tianqi Chen ve Carlos Guestrin tarafından 2016 yılında makale olarak yayınlanmıştır. Yöntemin en önemli özellikleri yüksek tahmin gücü elde edebilmesi, aşırı öğrenmenin önüne geçebilmesi, boş verileri yönetebilmesi ve bunları hızlı yapabilmesidir (Üstüner M. v.d, 2020).

Xgboost yöntemi rastgele ormanda olduğu gibi sadece sayısal özellikli veri kümeleri ile çalışabilir. Sistem uygulanmadan önce kategorik özellikler sayısal girdi olarak tanımlanmalıdır (Alsharı H. v.d, 2021).

Xgboost yöntemi sınıflandırma ve regresyon problemlerinde uçtan uca ağaç yükseltme sistemini oluşturur. Xgboost tekniği K sınıf için $K_E^i / i \in 1,2,3,\dots,n$ ağaç oluşumunda CART algoritmasını kullanır. Burada K tahmini ağaçların sayısını verir.

$$\widehat{y_i} = \varphi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{2.22}$$

Denklem (2.22)'de burada x_i eğitim sınıfının üyeleridir, y_i karşılık gelen sınıf etiketleridir, f_k k' inci ağaç için yaprak değeridir ve F tüm CART'lar için tüm K değerlerinin setidir. Sonucu düzenlemek için uygulama yapılır,

$$L(\varphi) = \sum_i l(\widehat{y_i}, y_i) + \sum_k \Omega(f_k) \tag{2.23}$$

denklem (2.23)'de ilk terim l hedef y_i ile tahmin $\widehat{y_i}$ arasındaki farkı ölçen türevlenebilir kayıp fonksiyonunu temsil eder, ikinci terim aşırı uydurmayı engeller, Ω modelin karmaşıklığının durdurur,

$$\Omega(\mathrm{f}) = \gamma \mathrm{T} + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{2.25}$$

denklem (2.24)'de lamda (λ), gama (γ) düzenlilik derecesini kontrol eden sabitlerdir, T ağaçtaki yaprak sayısıdır ve w her yaprağın değeridir. Gradyan arttırma, regresyon ve sınıflandırma problemlerinde etkili olduğu için gradyan arttırma kayıp fonksiyonunu kullanarak, ikinci dereceden Taylor genişlemesi ile genişletildi ve sabit terim çıkarılarak t adımında basitleştirilmiş bir amaç elde edilmiştir,

$$L^t = \sum_{i=1}^{n} [g_i f_i(x_i) + \frac{1}{2} h_i f_i^2(x_i)] + \Omega(f_i) \tag{2.25}$$

$$= \sum_{i=1}^{n} [g_i f_i(x_i) + \frac{1}{2} h_i f_i^2(x_i)] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{2.26}$$

$$= \sum_{j=1}^{T} \left[\left(\sum_{i \in i_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in i_j} h_i + \lambda \right) w_j^2 \right] + \gamma T \tag{2.27}$$

Burada $I_j = \{i / q(x_i) = j\}$ "t" yaprağının örnek kümesini gösterir ve

$$g_i = \frac{\partial l\left(\widehat{y_i}^{(t-1)}, y_i\right)}{\partial \widehat{y_i}^{(t-1)}}, h_i = \frac{\partial^2 l\left(\widehat{y_i}^{(t-1)}, y_i\right)}{\partial \left(\widehat{y_i}^{(t-1)}\right)^2} \tag{2.28}$$

Denklem (2.28) kayıp fonksiyonun birinci ve ikinci dereceden gradyan istatistiklerini verir. Optimum değer olan w_j yaprağının kalitesi ve verilen bir ağaç yapısı $q(x_i)$ denklem (2.30)'da hesaplanır,

$$w_j = -\frac{\sum_{i\epsilon l_j} g_i}{\sum_{i\epsilon l_j} h_i + \lambda} \tag{2.30}$$

$$L^t(q) = -\frac{1}{2}\sum_{j=1}^{T} \frac{\left(\sum_{i\epsilon l_j}^{T} g_i\right)^2}{\left(\sum_{i\epsilon l_j} h_i + \lambda\right)} + \lambda T \tag{2.31}$$

Uygulamada dallanmadan sonra sol I_L ve sağ I_R düğümlerin örnek kümelerindeki değerler kullanılarak değerlendirme yapılır. $I = I_L \cup I_R$ dallanmadan sonraki kayıp

$$L_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i\epsilon I_L} g_i\right)^2}{\left(\sum_{i\epsilon I_L} h_i + \lambda\right)} + \frac{\left(\sum_{i\epsilon I_R} g_i\right)^2}{\left(\sum_{i\epsilon I_R} h_i + \lambda\right)} \frac{(\sum_{i\epsilon I} g_i)^2}{(\sum_{i\epsilon I} h_i + \lambda)}\right] \tag{2.32}$$

yukarıdaki gibi bulunur (Thongsuwan S. v.d, 2021).

XGBOOST yöntemi için hiper-parametreler aşağıdaki gibidir (Budholiya K. v.d, 2020).

XGBOOT yönteminde eta parametresi gradyan artırma sırasında sapmayı azaltmak için küçülme adımını verir.

Gamma parametresi XGBOOST yöntemi uygulanırken ağacın yaprak düğümünde başka bir bölüm oluşturmak için minimum kayıp vermeye çalışır. Gamma ne kadar büyük olursa sonuç da o kadar iyi olur.

Max Deph parametresi XGBOOST yönteminde maksimum derinliği ayarlar.

Min child weight yönteminde ihtiyaç duyulan minimum örnek ağırlığı miktarıdır. Ağaç bölümleme adımı, örnek ağırlığının toplamı daha az olan bir yaprak düğümle sonuçlanırsa min child weight, ağaç oluşturma işlemi daha fazla bölünmeyi bırakacaktır. Min child weight ne kadar büyük olursa sonuç daha iyi olur.

Nrounds maksimum arttırma sayısını verir.

Alt örnekler (Subsample) eğitim örneklerinin alt örnek oranını verir. Bu parametre fazla sapmayı önler.

Colsample bytree oluşturulan her ağaç için alt örneklerin oranını verir.

Doğruluk, kappa, hassasiyet, özgüllük ve doğruluk için güven aralığı performans metrikleri aşağıda yeni bir başlık adı altında açıklanmıştır.

2.3. Sentetik Azınlık Yüksek Hızla Örnekleme Tekniği (SMOTE)

Sentetik azınlık yüksek hızla örnekleme tekniği (SMOTE) fonksiyonu ile veri setlerinde az temsil edilen sınıfa ait değerler sentetik olarak çoğaltılır. SMOTE azınlık sınıfta her bir örnek için k en yakın komşusu kadar sentetik örnekler sunarak örneklenir. Gereken aşırı örnekleme için k en yakın komşuları rastgele seçer (Chawla V. N. v.d., 2002).

SMOTE'nin çalışma adımları ise şu şekildedir;

- Azınlık sınıfına ait her gözlem için k en yakın komşu aranır,
- Azınlık sınıfına ait gözlem ile k en yakın komşusu olan gözlem arasındaki fark alınır,
- (0,1) arasında rastgele bir sayı (α) seçilir, farkın sonucu bu sayı ile çarpılır,
- Denklem (2.33)'deki eşitlik kullanılarak yeni sentetik gözlem elde edilir,

$$x_{yeni} = x_i + (x_j - x_i) * \alpha \tag{2.33}$$

- İstenilen sayıda sentetik gözleme ulaşmak için yukarıdaki aşamalar yinelemeli olarak tekrarlanır (Yavaş M. vd., 2020).

2.4. Sınıflandırma Yöntemlerinde Kullanılan Metrikler

Karar ağaçları, rastgele orman ve xgboost yöntemlerinin uygulamasında en uygun modeli belirlemek için model eğitim sonuçları ve modelin test performans sonuçlarına göre başlıca performans metrikleri incelenecektir. Bu parametreler ile en uygun modeller seçilir.

Çizelge 1. Karmaşıklık matrisi

	Tahmin	
Hedef	Pozitif Sınıf (P S)	Negatif Sınıf (N S)
Pozitif Sınıf (P S)	Doğru Pozitif (D P)	Yanlış Negatif (Y N)
Negatif Sınıf (N S)	Yanlış Pozitif (Y P)	Doğru Negatif (D N)

Karmaşıklık matrisi, tahminlerin doğruluğu hakkında bilgi veren bir ölçüm aracıdır.

Doğruluk (Accuracy) ölçeği modeller arasında en uygun model seçilmesine yarayan metriktir. Yalnızca doğruluk değeri ile model seçimi yapmak yeterli olmaz diğer parametrelerinde iyi sonuç vermesi en uygun model seçilmesine yardımcı olur (Rezapour M. v.d, 2019a).

$$Doğruluk = \frac{DP+DN}{DP+DN+YP+YN} \tag{2.34}$$

Güven aralığı doğruluk değerinin alabileceği alt sınır ile üst sınırları verir.

Kappa istatistiği ise tahmin edilen ve gerçek sınıf değerleri arasındaki korelasyonu verir. Kappa istatistiği, Cohen (1960) tarafından geliştirilen iki veya daha fazla gözlemcinin yaptığı değerlendirmeler arasındaki uyuşmayı belirlemek için kullanılır (Şimşekli Y. v.d, 2011).

Özgüllük (Specificity) sınıflandırmada doğru şekilde tanımlanan negatiflerin oranını verir (Rezapour M. v.d, 2019b).

$$Özgüllük = \frac{DN}{DN+YP} \tag{2.35}$$

Hassasiyet (Sensitivity) sınıflandırmada doğru şekilde tanımlanan pozitiflerin oranını verir (Uzunlu Y. B. ve Hussain M. S.a).

$$Hassasiyet = \frac{DP}{DP+YP} \tag{2.36}$$

Kesinlik (Precision) tüm sınıfların doğru olarak ne kadar tahmin edildiğinin bir ölçüsüdür (Rezapour M. v.d, 2019c).

$$Kesinlik = \frac{DP}{DP+YP} \tag{2.37}$$

F skoru ya da F ölçeği gerçek pozitif değerlerin oranının (recall) ve hassasiyetin (precision) harmonik ortalamasıdır. Sınıflandırıcının ne kadar iyi performans gösterdiğinin ölçüsüdür ve sınıflandırıcıları karşılaştırmakta sıklıkla kullanılır (Uzunlu Y. B. ve Hussain M. S.b).

$$F = 2 * \frac{Kesinlik * Hassasiyet}{Kesinlik + Hassasiyet} \quad (2.38)$$

3. TİROİT HASTALIĞINDA AĞAÇ TEMELLİ MAKİNE ÖĞRENME YÖNTEMLERİNİN KULLANILMASI

3.1 TİROİT HASTALIĞI

3.1.1 Tiroit bezi tanımı ve işlevi

Tiroit bezi, boynun ön tarafında, soluk borusunun her iki yanında yerleşmiş bulunan, sağ ve sol lob olmak üzere iki lobdan oluşan, C5 T1 vertabralar arasında yer alan bir salgı bezidir. Tiroit bezi, vücut metabolizmasını ve hızını düzenleyen T3(triiyodotironin) ve T4(tiroksin) hormonlarını salgılayarak dolaşım sistemine gönderir. Salgılanan hormon fazla olursa vücut metabolizması hızlanır ve hipertiroit hastalığı görülür. Tiroit hormonunun gereğinden az salgılanması durumuna hipotiroit denilmektedir (Solmaz R. vd., 2013).

Beyinde bulunan hipofiz bezinden salgılanan TSH (Tiroit Uyarıcı Hormon – Tiroit Stimulan Hormon) ile tiroit bezinin çalışması kontrol edilir. Kandaki tiroit hormonlarının(T3 ve T4) azalması durumunda beyindeki hipofiz bezinden salgılanan TSH artar ve bu sayede tiroit bezi T3 ve T4 salgısını arttırır. Eğer kandaki T3 ve T4 miktarı artarsa bu sefer hipofiz bezi TSH salgısını azaltır ve bunun neticesinde tiroit bezi T3 ve T4 salgı miktarını azaltır (Yıldız A., 2019a).

3.1.2 Tiroit bezi hastalıkları

3.1.2.1 Guatr

Tiroit bezinin normal kabul edilenden fazla büyümesi sonucu oluşan hastalıktır (Kaba M., 2013a).

3.1.2.2 Nodül

Tiroit bezi içerisinde tiroit bezi dokusuna benzemeyen farklı dokuların oluşması durumudur. Nodül olan hastaların yaklaşık %5 oranında kansere yakalanma riski

vardır. Nodüller soğuk, sıcak ve ılık olmak üzere 3 kategoride incelenir. Sıcak nodüllü hastalarda kansere yakalanma riski %5 oranının daha altındadır. Baş boyun bölgesine veya mediastene radyasyon uygulaması özellikle çocuklarda artmış tiroit kanseri insidansı ile ilişkilidir (Tosun C. F., 2013a).

3.1.2.3 Tiroidit

Tiroit bezinin apse olması sonucu oluşur (Kaba M., 2013b).

3.1.2.4 Hipertiroidi

T3 ve T4 tiroit hormonlarının gereğinden fazla salgılanması sonucu ortaya çıkar. Bu hastalığın nedenleri arasında vücudun kendi tiroit organını yabancı bir doku olarak algılaması sonucu oluşan Graves hastalığı otoimmün hastalıklardan biridir. Belirtileri ise şu şekildedir; Ani görülen aşırı kilo kayıpları, aşırı terleme, uyku düzensizliği, ellerde titreme, kalp çarpıntısı gibi rahatsızlıklar gözlemlenmektedir (Begum A. ve Parkavi A., 2019a).

3.1.2.5 Hipotiroidi

T3 ve T4 hormonlarının gereğinden az salgılanması sonucu ortaya çıkar. Bu hastalığın nedenleri arasında iltihaplanma ve tiroit bezi hasarı gösterilir. Belirtileri ise şu şekildedir; Halsizlik, güç kaybı, normal dışı üşüme, ses kısılması ve kalınlaşması, ellerde, bacaklarda, yüz bölgesinde ve gözlerin etrafında şişme, ciltte kuruma ve kalınlaşma, terlemede azalma gibi rahatsızlıklar gözlemlenmektedir (Begum A. ve Parkavi A., 2019b).

3.1.3 Tiroit tanısı

Tiroit tanısı kan testleri, ultrasonografi, sintigrafi ve ince iğne aspirasyon biyopsisi yöntemleriyle yapılmaktadır (Gül S. S., 2020).

3.1.4 Kan testleri

Kandaki T3, T4 ve TSH hormonlarının miktarlarının ölçülmesiyle yapılır. Tiroit bezinin durumunu gösteren en belirleyici ölçüm, TSH miktarının belirlenmesidir. Eğer kandaki TSH miktarı olması gerekenden az ise tiroit bezinin fazla çalıştığı kanaatine varılır, TSH miktarı olması gerekenden fazla ise tiroit bezinin az çalıştığı kanaatine varılır (Yıldız A., 2019b).

3.1.5 Tiroit ultrasonografisi

Tiroit bezinin büyüklüğü, eğer nodül varsa nodülün yerinin tespiti yapılır ve hacmi belirlenir. Tedavi neticesinde nodülün küçülüp küçülmediği de bu yol ile belirlenir (İğci E. ve Göktay Y., 1996).

3.1.6 Tiroit sintigrafisi

Teknesyum isimli bir maddenin kana enjekte edilerek yapılan nükleer tıp yöntemidir. Bu tetkik, TSH'ı düşük ve nodülü olan hastalara uygulanır. Nodülün sıcak veya soğuk olduğunu belirlemek için yapılır. Tiroit dokusunun varlığı, büyüklüğü, şekli, yeri ve fonksiyonunun değerlendirilmesi için kullanılan bir görüntüleme metodudur (Tosun C. F., 2013b).

3.2 Tiroit Hastalığı ve Makine Öğrenmesi

Literatürde makine öğrenme yöntemleri ve tiroit hastalığı konulu makaleler incelendiğinde çalışmalarda genel anlamda, bazı karar ağacı algoritmaları ve rastgele orman yöntemi en iyi sonuçları vermektedir.

Begum A. ve Parkavi A. ,(2019) çalışmanın amacı, tiroit hastalığının farklı sınıflandırma teknikleri kullanılarak tahmin edilmesi ve ayrıca hipertiroit, hipotiroit ile ilgili TSH, T3, T4 değişkenlerinin korelasyonunu bulmaktır. Çalışmada kullanılan veri kümesi, UCL makine öğrenimi havuzundan çekilmiştir. Veri tabanı hastaların tiroit kayıtlarından oluşmuştur. Her tiroit hastası kaydı 15 nitelikten oluşmuştur.

Çalışmada kullanılan veri setine ID3 ve C4.5 karar ağacı teknikleri uygulanmıştır (Begum A. ve Parkavi A., 2019c).

Kousarrizi N. v.d. (2012) çalışmasında, 2 farklı veri kümesi kullanılmıştır. Birinci veri kümesi UCL makine öğreniminden alınmıştır. İkincisi, geçmişte kalan gerçek verilerdir. Birinci veri seti 215 hasta ve 5 nitelikten oluşmaktadır. İkinci veri seti ise 2010 yılında altı ay boyunca İmam Humeyni hastanesinin salgı koğuşunda bulunan K. N. Toosi Teknoloji Üniversitesi Akıllı Sistem Laboratuvarı tarafından toplanmıştır. Bu veri seti, her biri 15 ikili ve 6 sürekli olmak üzere 21 özellikten oluşan 1538 hastadan oluşmaktadır. Bu veri setinde 331 hasta Hyper sınıfına, 648 hasta Hypo sınıfına ve 559 hasta Normal sınıfına aittir. Özellikler cinsiyet, yaş, T3, T3RU, T4, FT4, TSH, çarpıntı, uyuşukluk, ekzoftalmi, ishal, kabızlık, ödem, adet kanaması, terleme, ısı intoleransı, soğuğa tahammülsüzlük, kilo değişikliği, iştah, titreme ve sinirliliktir. Makalede sınıflandırma yöntemlerinden SVM tekniği kullanılmıştır (Kousarrizi N. v.d., 2012).

Akgül G. v.d. (2020) hipotiroidi hastalığının tanı sürecinde hastalara sorulan soru ve uygulanan test sonuçlarını kullanarak hipotiroidi hastalığının doğru teşhis oranını arttıracak veri madenciliği temelli bir sistem kullanmıştır. Diğer amaç ise dolaylı olarak teşhis için kullanılan girişimsel testlerden oluşabilecek komplikasyonları azaltmaktır. Bu amaçlar doğrultusunda UCL makine öğrenmesi veri tabanında yer alan ve 151 tanesi hipotiroidi geri kalanı hipotiroidi olmayan toplam 3163 örnekten oluşan veri seti kullanılarak yeni örneklerin hipotiroidi olup olmadığı tahmin edilmiştir. Veri setindeki dengesiz dağılımı ortadan kaldırmak için veri setine farklı örnekleme teknikleri uygulanarak Lojistik Regresyon, K En Yakın Komşu ve Destek Vektör Makinesi sınıflandırıcıları ile hipotiroidi hastalığını teşhis edecek modeller oluşturulmuştur (Akgül G. v.d., 2020).

Sidiq U. v.d.(2019) Kashmir'deki önde gelen teşhis laboratuvarlarından alınan veri setine veri madenciliği tekniklerini uygulamışlardır. Veri seti, neredeyse tüm yaş gruplarından 807 hastanın kaydını içermektedir. 807 hastadan (224 Erkek ve 583 Kadın) 553'ü normale, 218'i hipotiroidiye, 36'sı hipertiroidiye aittir. En yakın komşular, Destek vektör makinesi, Karar ağacı ve Naive bayes gibi sınıflandırma teknikleri kullanılarak deneysel bir çalışma yürütülmüştür. Karar Ağacı, diğer sınıflandırma tekniklerine göre en yüksek doğruluğu% 98,89 ile elde etmiştir (Sidiq U. v.d., 2019).

Bao W. v.d. (2019) tiroit nodüllerinin sınıflandırılması için doğrusal ve doğrusal olmayan makine öğrenme algoritmalarını kullanmışlardır. Sınıflandırma yöntemlerinin uygulaması açık kaynak kodlu R programlama dili ile yapılmıştır. Çalışmada sınıflandırma yöntemlerinden rastgele orman, knn, nnet, k-SVM, NB ve glmnet algoritmaları kullanılmıştır. Doğrusal ve doğrusal olmayan makine öğrenme algoritmaların doğruluk ve kappa değerleri karşılaştırılırmıştır. Kappa katsayısının 0,7'den büyük olması iyi uyum olarak kabul edilmiştir. Genel olarak doğrusal olmayan algoritmalar doğrusal olan algoritmalar ile benzer performans göstermiştir. En iyi sonucu veren algoritmalar ise rastgele orman ile k-SVM'dir (Bao W. v.d., 2019).

Margret J. v.d. (2012) karar ağacı algoritmaları kullanarak çeşitli bölme kuralı analiz edilmiş ve karşılaştırılmıştır. Çalışmada kullanılan veri kümesi, UCL makine öğrenimi havuzundan çekilmiştir. Veri seti 21 öznitelik ve 3 sınıftan oluşmaktadır. Çalışmada normalleştirilmiş tabanlı bölme kurallarının yüksek doğruluk ve hassasiyete veya gerçek pozitif orana sahip olduğu sonucuna ulaşılmıştır. Bu çalışma herhangi bir tıbbi veri seti için genişletilerek kullanılabileceği öngörülmüştür (Margret J. v.d., 2012).

Banu G.R. (2017) tiroit hastalığının teşhisi için karar ağacı ve veri madenciliği tekniklerini kullanmıştır. Çalışmada kullanılan hipotiroit veri seti UCL makine öğrenimi havuzundan alınmıştır. Hipotiroit veri kümesi, 3481 vakanın negatif kategorisine ait olduğu, 194 vakanın kompanse hipotiroit kategorisine ait olduğu, 95 vakanın birincil hipotiroit kategorisine ait olduğu ve 2 vakanın ikincil hipotiroit kategorisine ait olduğu 3772 örnekten oluşur. Toplam 30 nitelikten oluşan veri setinden verileri sınıflandırmak için kullanılacak sadece 12 nitelik alınmıştır. Uygulama WEKA programı ile yapılmış sınıflandırma algoritmaları yüksek doğruluk oranına sahiptir. KNN %96.35, SVM %94.44, C4.5 %99.47 ve rastgele orman %99.47 doğruluk oranlarına ulaşılmıştır. Araştırmada en iyi sonucu veren algoritmalar %99.47 ile C4.5 ve rastgele orman algoritmalarıdır (Banu G. R., 2017).

Bu çalışmada uygulama bölümünde Muğla Sıtkı Koçman Üniversitesi Eğitim ve Araştırma Hastanesinde toplanan kronik otoimmün tiroit veri seti ve University of California Irvine Machine Learning Repository sitesinden çekilen açık veri seti kullanılmıştır. Kronik otoimmün veri seti 2 farklı model kurularak incelenmiştir.

3.3. Kronik Otoimmün Tiroit Verisi 1

Kurulacak olan birinci modelde açıklayıcı değişkenler arasında Ultrason (USG) görüntülerinin de yer aldığı toplamda iki sınıflı bir yanıt değişkeni ve 15 açıklayıcı değişken yer alacaktır. Birinci modelin veri seti "CATveri1" adı ile tanımlanmış ve uygulamanın yapılacağı açık kaynak kodlu R program diline aktarılır. CATveri1 olarak kodlanan veri seti içerisinde yanıt değişkeni gruplar olarak tanımlanmıştır. Açıklayıcı değişkenler ise (yas, cinsiyet, TSH, sT3, sT4, ATPO, ATG, USGboyutsağ1, USGboyutsağ2, USGboyutsağ3, USGboyutsol1, USGboyutsol2, USGsol3, USG, LAP) olarak tanımlanmıştır. Veri setindeki değişkenlerin kısaltmalarının karşılıkları ise şu şekildedir;

Gruplar: Kronik otoimmün tiroit (CAT) ve kontrol grupları.

USG: Genel ultrason görüntüsü.

USGboyutsağ1: 1. sağ ultrason görüntüsü.

USGboyutsağ2: 2. sağ ultrason görüntüsü.

USGboyutsağ3: 3. sağ ultrason görüntüsü.

USGboyutsol1: 1. sol ultrason görüntüsü.

USGboyutsol2: 2. sol ultrason görüntüsü.

USGboyutsol3: 3. sol ultrason görüntüsü.

TSH: Tiroit uyarıcı hormonudur.

sT3: Tiroit bezinin triiyodotironin hormonudur.

sT4: Tiroit bezinin tiroksin hormonudur.

ATG: Anti Tiroglobulin tiroit bezi tarafından üretilen proteindir.

ATPO: Tiroit bezinin ürettiği antikora anti TPO adı verilir.

LAP: Lenf bezi büyümesidir. Diğer bir adı ile Lenfadenopati adı verilir.

CATveri1 için kurulan model ise şu şekildedir;

$$Y = Gruplar \sim (\text{yas}, \text{cinsiyet}, \text{TSH}, \text{sT3}, \text{sT4}, \text{ATPO}, \text{ATG}, \text{USGboyutsağ1}, \text{USGboyutsağ2},$$

$USGboyutsağ3, USGboyutsol1, USGboyutsol2, USGsol3, USG, LAP)$

Veri setinde kayıp gözlemler olduğu için ham veri setine "complete.cases" komutu uygulanarak eksik gözlemler silinmiştir.

3.4. Kronik Otoimmün Tiroit Verisi 2

İkinci model için ise ultrason (USG) görüntüleri ve LAP değişkeni çıkarıldığı zaman karar kurallarının nasıl gerçekleştiğini görmek için "CATveri2" adında iki sınıflı yanıt değişkeni ve 7 tane açıklayıcı değişkeni olan yeni bir veri seti oluşturulmuştur. İkinci modelin veri seti "CATveri2" adı ile tanımlanmış ve uygulamanın yapılacağı açık kaynak kodlu R program diline aktarılır. CATveri2 olarak kodlanan veri seti içerisinde yanıt değişkeni gruplar olarak tanımlanmıştır. Açıklayıcı değişkenler ise (yas, cinsiyet, TSH, sT3, sT4, ATPO, ATG) olarak tanımlanmıştır. Veri setindeki değişkenlerin kısaltmalarının karşılıkları ise şu şekildedir;

Gruplar: Kronik otoimmün tiroit (CAT) ve kontrol grupları.

TSH: Tiroit uyarıcı hormonudur.

sT3: Tiroit bezinin triiyodotironin hormonudur.

sT4: Tiroit bezinin tiroksin hormonudur.

ATG: Anti Tiroglobulin tiroit bezi tarafından üretilen proteindir.

ATPO: Tiroit bezinin ürettiği antikora anti TPO adı verilir.

CATveri2 için oluşturulan modelimiz ise şu şekildedir.

$Y = Gruplar \sim (\text{yas}, \text{cinsiyet}, \text{TSH}, \text{sT3}, \text{sT4}, \text{ATPO}, \text{ATG}, \text{USG})$

Veri setinde kayıp gözlemler olduğu için ham veri setine "complete.cases" komutu uygulanarak eksik gözlemler silinmiştir.

3.5. UCL Verisi

University of California Irvine Machine Learning Repository sitesinden çekilen açık veri seti 215 kişiye ait tiroit verilerini içermektedir.

Veri seti “UCLveri” adı ile tanımlanmış ve uygulamanın yapılacağı açık kaynak kodlu R program diline aktarılır. UCL olarak kodlanan veri seti içerisinde 3 sınıflı yanıt değişkeni sınıflar olarak tanımlanmıştır. Açıklayıcı değişkenler ise (RT3U, T4, T3, TSH, DTSH) şeklinde tanımlanmıştır.

Veri setindeki değişkenlerin kısaltmalarının karşılıkları ise şu şekildedir;

Sınıflar: Tiroit, hipotiroit ve hipertiroitten oluşan 3 sınıflı yanıt değişkenidir.

TSH: Tiroit uyarıcı hormonudur.

sT3: Tiroit bezinin triiyodotironin hormonudur.

sT4: Tiroit bezinin tiroksin hormonudur.

DTSH: Tiroit fonksiyon testidir.

UCLveri olarak kodladığımız UCL Machine Learning’ den aldığımız veri seti için ise kurulan model ise şu şekildedir;

$Y = Sınıflar \sim (\mathrm{RT3U}, \mathrm{T4}, \mathrm{T3}, \mathrm{TSH}, \mathrm{DTSH})$

4. UYGULAMA

Veri setlerine ilişkin tanımlayıcı istatistikler incelendiğinde, CATveri1 ve CATveri2 modelleri içinde kontrol gruplar ile CAT gruplarının dengesiz dağılmış durumda olduğu görülmektedir. Yöntemler uygulandığında sağlıklı sonuçlar vermeyeceği için SMOTE fonksiyonu ile veri setleri dengelenir. Smote yapılmış veri setleri CATveri1 ve CATveri2 olarak tanımlanmıştır. Son durumda CATveri1 ve CATveri2'de 259 hasta 256 kontrol grupları oluşturulmuştur. UCL verisi için ise orijinal ham veri kullanılacaktır. UCL verisinde ise 3 sınıflı tiroit 150, hipotiroit 35 ve hipertiroit 30 olmak üzere toplamda 215 gözlem vardır. Uygulama aşamaları oluşturulan 3 model ile yapılmıştır. Üç veri setinin de %80'i eğitim için %20'si de test için ayırılmıştır. Yöntemlerin uygulanma aşamasında fitcontrol adı altında kontrol grubu oluşturulmuştur. Veri setlerine uygulanacak olan sınıflandırma teknikleri CTREE, CART, C5.0, rastgele orman ve XGBOOST yöntemleridir. Eğitim veri setleri ile modeller eğitilip doğruluk ve kappa değerlerine göre en iyi modeller seçilecektir. Test veri setleri ile ise performansları ölçülecektir. Ardından performansların karşılaştırılması yapılacaktır.

4.1. CATveri1 Verisinin Eğitilmesi

CATveri1 veri setinin %80'i olan eğitim veri setini oluşturan 413 gözlem 15 nitelik üzerinden yöntemler uygulandığında doğruluk, kappa ve diğer parametrelerin istatistikleri göz önüne alınarak en uygun modeller seçilir.

Çizelge 2. CATveri1 koşullu ağaç modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)		
Örnek boyutlarının özeti: 371, 372, 373, 372, 371, 371, ...		
Mincriterion	Doğruluk	Kappa
0.5783492	0.995119	0.9902381
0.6741079	0.995119	0.9902381
0.7099023	0.995119	0.9902381

Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır.
Model için kullanılan son değer mincriterion = 0,7099023.

Çizelge 2'de eğitim verisi için oluşturulan 3 model için doğruluk değerleri yaklaşık 0,995 olarak bulunmuştur. Kappa istatistikleri ise yaklaşık olarak 0,99 olarak bulunmuş ve mincriterion kriterine göre en iyi model 3. modeldir.

Çizelge 3. CATveri1 rastgele orman modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)		
Örnek boyutlarının özeti: 371, 371, 372, 372, 373, 371, ...		
Mtry	**Doğruluk**	**Kappa**
6	0.995180	0.9903513
12	0.997561	0.9951132

Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır.
Model için kullanılan son değer mtry = 12'dir.

Çizelge 3'de eğitim verisi için oluşturulan 2 modelde doğruluk değerleri yaklaşık 0.995 ve 0.998 olarak bulunmuştur. Kappa istatistikleri ise yaklaşık olarak 0.990 ve 0.995 olarak bulunmuştur.

Çizelge 4. CATveri1 XGBOOST modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)								
Örnek büyüklüklerinin özeti: 371, 371, 372, 372, 372, 371, ...								
Eta	Max Depth	Gamma	Colsample Bytree	Min Child Weigh	Subsample	nrounds	Doğruluk	Kappa
0.161	**5**	**1.680**	**0.693**	**7**	**0.928**	**165**	**0.998**	**0.996**
0.205	5	3.884	0.476	17	0.331	702	0.945	0.889
0.224	6	0.270	0.601	14	0.767	594	0.998	0.996

Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır.
Model için kullanılan son değerler nrounds = 165, max_depth = 5, eta =0, 1606484, gamma=1,6791672 colsample_bytree= 0,6925749, min_child_weight = 7 ve subsample = 0,9278271.

Çizelge 4'de eğitim verisi için 3 model oluşmuştur. Doğruluk değeri, kappa istatistiği, Gamma istatistiği ve diğer parametrelere göre en uygun model 1. Model seçilmiştir.

Çizelge 5. CATveri1 C5.0 karar ağacı algoritması modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)				
Örnek büyüklüklerinin özeti: 372, 371, 372, 372, 372, 371, ...				
Model	**Winnow**	**Denemeler**	**Doğruluk**	**Kappa**
Rules	Yanlış	44	0.995122	0.9902381
Rules	Yanlış	78	0.995122	0.9902381
Tree	Doğru	55	0.995122	0.9902381

Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır.
Model için kullanılan son değerler denemeler = 44, model = Rules ve winnow = doğru.

Çizelge 5'de eğitim verisi için 3 model oluşmuştur. C5.0'e göre 44 deneme, 0,995 doğruluk oranı ve 0,990 kappa istatistiğine göre rules modeli seçilmiştir.

Çizelge 6. CATveri1 CART karar ağacı algoritması modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)		
Örnek boyutlarının özeti: 372, 371, 372, 371, 371, 372, ...		
cp	**Doğruluk**	**Kappa**
0.0000000	0.997561	0.99511323
0.995122	0.550000	0.09511323

Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır.
Model için kullanılan son değer cp = 0'dır.

Çizelge 6'da eğitim verisi için yapılan CART karar ağacı algoritmasına göre 2 model oluşturulmuştur. Doğruluk değeri yaklaşık 0,998 ve cp=0 olduğu durumda en uygun model 1. model seçilmiştir.

Eğitim veri setlerinde en uygun modeller seçildikten sonra test veri seti ile performans değerlendirmeleri yapılacaktır.

4.2. CATveri1 Verisinin Test Edilmesi

Test verisi için kalan 102 gözlem 7 nitelik üzerinden test veri setine uygulanan modellerin performanslarına ilişkin karmaşıklık matris parametrelerinin istatistikleri üzerinden en iyi yöntem seçilir.

Çizelge 7. CATveri1 test verisi için uygulanan yöntemlerin performans indikatörleri.

Model		CAT	Kontrol
CTREE	CAT	51	1
	Kontrol	0	50
Rastgele Orman	CAT	51	1
	Kontrol	0	50
XGBOOST	CAT	51	1
	Kontrol	0	50
C5.0	CAT	51	1
	Kontrol	0	50
CART	CAT	51	1
	Kontrol	0	50

Çizelge 7. devamı

Model/Parametre	Doğruluk	%95 Güven Aralığı	Kappa	Hassasiyet	Özgüllük	F1 Değeri
CTREE	0,9902	0,947 – 0,999	0,9804	1,0000	0.9804	0.9903
Rastgele Orman	0,9902	0,947 – 0,999	0,9804	1,0000	0.9804	0.9903
XGBOOST	0,9902	0,947 – 0,999	0,9804	1,0000	0.9804	0.9903
C5.0	0,9902	0,947 – 0,999	0,9804	1,0000	0.9804	0.9903
CART	0,9902	0,947 – 0,999	0,9804	1,0000	0.9804	0.9903

Çizelge 7'de test veri setine uygulanan tekniklerin performans indikatörleri verilmiştir.

CTREE yöntemine göre test veri seti için oluşturulan modelde 51 CAT grubunun tamamını CAT grubuna atamıştır. 51 kontrol olgusunun 50'sini kontrol grubuna 1 tanesinin CAT grubuna atamıştır. Oluşturulan modelin kappa istatistiği 0,9804, hassasiyeti 1, Özgüllüğü 0,9804, F1 değeri 0,9903, doğruluk oranı 0,9902 ve %95 güven aralığı (0,9466 ile 0,9998) arasındadır.

Rastgele orman yöntemine göre test veri seti için oluşturulan modelde 51 CAT grubunun tamamını CAT grubuna atamıştır. 51 kontrol olgusunun 50'sini kontrol grubuna 1 tanesinin CAT grubuna atamıştır. Oluşturulan modelin kappa istatistiği 0,9804, hassasiyeti 1, Özgüllüğü 0,9804, F1 değeri 0,9903, doğruluk oranı 0,9902 ve %95 güven aralığı (0,9466 ile 0,9998) arasındadır.

XGBOOST yöntemine göre test veri seti için oluşturulan modelde 51 CAT grubunun tamamını CAT grubuna atamıştır. 51 kontrol olgusunun 50'sini kontrol grubuna 1

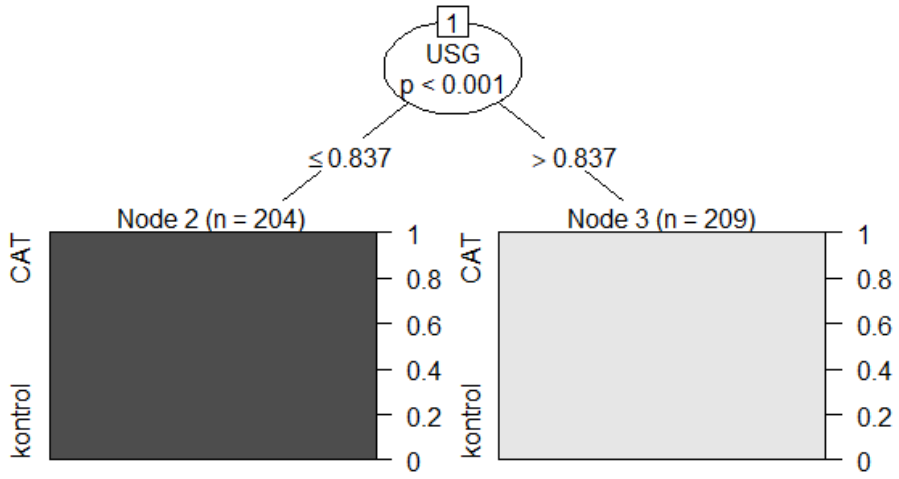
tanesinin CAT grubuna atamıştır. Oluşturulan modelin kappa istatistiği 0,9804, hassasiyeti 1, Özgüllüğü 0,9804, F1 değeri 0,9903, doğruluk oranı 0,9902 ve %95 güven aralığı (0,9466 ile 0,9998) arasındadır.

C5.0 yöntemine göre test veri seti için oluşturulan modelde 51 CAT grubunun tamamını CAT grubuna atamıştır. 51 kontrol olgusunun 50'sini kontrol grubuna 1 tanesinin CAT grubuna atamıştır. Oluşturulan modelin kappa istatistiği 0,9804, hassasiyeti 1, Özgüllüğü 0,9804, F1 değeri 0,9903, doğruluk oranı 0,9902 ve %95 güven aralığı (0,9466 ile 0,9998) arasındadır.

CART yöntemine göre test veri seti için oluşturulan modelde 51 CAT grubunun tamamını CAT grubuna atamıştır. 51 kontrol olgusunun 50'sini kontrol grubuna 1 tanesinin CAT grubuna atamıştır. Oluşturulan modelin kappa istatistiği 0,9804, hassasiyeti 1, Özgüllüğü 0,9804, F1 değeri 0,9903, doğruluk oranı 0,9902 ve %95 güven aralığı (0,9466 ile0,9998) arasındadır.

Bütün modeller iyi sonuç vermiştir. Tüm modellerin iyi sonuç vermesini etkileyen değişkenler ise ultrason (USG) görüntüleridir.

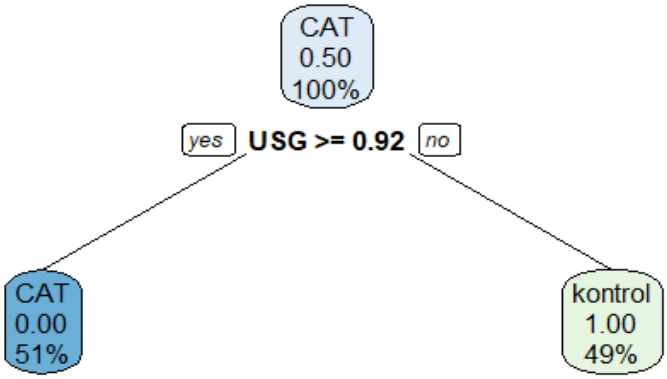
CTREE ve CART algoritmalarımızın karar ağacı grafikleri aşağıdaki gibi verilmiştir.



Şekil 4. CATveri1 için koşullu çıkarım ağacı

Şekil 4'deki koşullu çıkarım ağacı (CTREE) için oluşturulan karar ağacında sınıflandırmayı Ultrasona (USG) göre yapmıştır. Grafiğe göre USG değeri 0.837'ye

eşit ya da küçük olması durumda 2. düğüme 0.837'den büyük olması durumunda ise 3. düğüme atamıştır 2. Düğümde bulunan 204 kişinin hepsi kronik otoimmün tiroit tanısı konulmuştur. Düğüm 3'de bulunana 209 kişinin hepsi kontrol grubuna dahil edilmiştir.



Şekil 5. CATveri1 için CART karar ağacı

Şekil 5'deki CART için oluşturulan karar ağacında sınıflandırmayı Ultrasona (USG) göre yapmıştır. Grafiğe göre USG değeri 0.92'ye eşit ya da büyük olması durumda 0 kontrol olma olasılığı ile kronik otoimmün tiroit (CAT) tanısı konulmuştur. 0.92'den küçük olması durumunda ise 1 kontrol olma olasılığı ile kontrol grubuna atama yapılır.

4.3. CATveri2 Verisinin Eğitilmesi

CATveri2 veri setinde bazı yöntemlerdeki problemleri çözmek için ATPO ve sT3 değişkenlerinin 10 tabanında logaritmaları alınmıştır. Logaritmalar alındıktan sonra ATPO değişkeninde gözlemlerden biri –sonsuz değerini aldığı için o gözlemin olduğu satır silinmiştir. "CATveri2"veri setinin %80'ni olan eğitim veri setini oluşturan 413 gözlem 15 nitelik üzerinden yöntemler uygulandığında doğruluk, kappa ve diğer parametrelerin istatistiklerine göz önüne alınarak en uygun modeller seçilmiştir.

Çizelge 8. CATveri2 koşullu ağaç modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)		
Örnek boyutlarının özeti: 372, 371, 370, 370, 371, 370, ...		
Mincriterion	Doğruluk	Kappa
0.3875749	0.9414518	0.8827523
0.6064771	0.9414518	0.8827523
0.9732761	0.9317538	0.8634929
Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır. Model için kullanılan son değer mincriterion = 0,6064771.		

Çizelge 8'de eğitim verisi için oluşturulan 3 modelin 2 tanesi için doğruluk değerleri yaklaşık 0.942 ve 3. model için 0.932 olarak bulunmuştur. Kappa istatistikleri ise yaklaşık 0.883 ile 0.864 olarak bulunmuştur. En iyi model 2. Modeldir.

Çizelge 9. CATveri2 rastgele orman modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)		
Örnek boyutlarının özeti: 371, 371, 370, 371, 370, 371, ...		
Ayarlama parametrelerinde sonuçları yeniden örnekleme:		
Mtry	Doğruluk	Kappa
3	0.9780430	0.9560539
7	0.9781039	0.9561439
Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır. Model için kullanılan son değer mtry =7'dir.		

Çizelge 9'da eğitim verisi için oluşturulan 2 model için doğruluk değerleri yaklaşık 0.978 bulunmuştur. Kappa istatistikleri ise yaklaşık olarak 0.956 bulunmuştur. Rastgele orman yönteminde Mtry göre en iyi model 2. modeldir.

Çizelge 10. CATveri2 XGBOOST modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)								
Örnek büyüklüklerinin özeti: 372, 371, 370, 371, 370, 372, ...								
eta	Max Depth	Gamma	Colsample Bytree	Min Child Weigh	subsample	Nrounds	Doğruluk	Kappa
0.199	9	9.922	0.610	7	0.985	363	0.938	0.875
0.420	7	5.643	0.678	4	0.941	919	0.962	0.923
0.508	8	6.687	0.37	16	0.822	781	0.923	0.846
Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır.								
The final values used for the model were nrounds = 919, max_depth = 7, eta = 0.4199452, gamma = 5.642197, colsample_bytree = 0.6777757, min_child_weight =4 and subsample = 0.9402708.								

Çizelge 10'da eğitim veri seti için oluşturulan XGBOOST yöntemine göre üç modele ulaşılmıştır doğruluk, kappa, gamma, eta, nrounds ve diğer değerlere göre en iyi sonucu veren model 2. modeldir.

Çizelge 11. CATveri2 C5.0 karar ağacı modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)				
Örnek büyüklüklerinin özeti: 371, 371, 371, 370, 371, 370, ...				
Model	**Winnow**	**Denemeler**	**Doğruluk**	**Kappa**
Rules	Doğru	28	0.9661411	0.9323025
Tree	Yanlış	76	0.9782201	0.9564286
Tree	Yanlış	78	0.9782201	0.9564286
Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır.				
Model için kullanılan son değerler denemeler = 76, model = Tree ve winnow = yanlış.				

Çizelge 11'de eğitim verisi için yapılan C5.0 karar ağacı algoritmasına göre 3 model oluşturulmuştur. Doğruluk değeri ve diğer durumlara göre en uygun model 2. Model seçilmiştir.

Çizelge 12. CATveri2 CART algoritması modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)		
Örnek boyutlarının özeti: 370, 371, 370, 370, 370, 372, ...		
cp	Doğruluk	Kappa
0.000000000	0.9246748	0.8492967
0.001626016	0.9246748	0.8492967
0.082926829	0.8954559	0.7909298
Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır.		
Model için kullanılan son değer cp = 0'dır.		

Çizelge 12'de eğitim verisi için yapılan CART karar ağacı algoritmasına göre 3 model oluşturulmuştur. Doğruluk değeri ve CP=0,00 olduğu durumda en uygun model 1. model seçilmiştir.

4.4. CATveri2 Verisinin Test Edilmesi

Test verisi için kalan 102 gözlem 7 nitelik üzerinden test veri setine uygulanan modellerin performanslarına ilişkin karmaşıklık matris parametrelerinin istatistikleri üzerinden en iyi yöntem seçilmiştir.

Çizelge 13. CATveri2 test verisi için uygulanan yöntemlerin performans indikatörleri.

Model		CAT	Kontrol
CTREE	CAT	47	3
	Kontrol	4	48
Rastgele Orman	CAT	50	1
	Kontrol	1	50
XGBOOST	CAT	46	1
	Kontrol	5	50
C5.0	CAT	49	1
	Kontrol	2	50
CART	CAT	47	1
	Kontrol	4	50

Çizelge 13 devamı

Model/Parametre	Doğruluk	%95 Güven Aralığı	Kappa	Hassasiyet	Özgüllük	F1 Değeri
CTREE	0,9314	0,8637 - 0,9720	0,8627	0,9216	0,9412	0,9307
Rastgele Orman	0,9706	0,9310 - 0,9976	0,9608	0,9804	0,9804	0,9804
XGBOOST	0,9412	0,8764 - 0,9781	0,8824	0,9020	0,9804	0,9388
C5.0	0,9706	0,9164 - 0,9939	0.9412	0,9608	0,9804	0,9703
CART	0,9510	0,8893 - 0,9839	0,9020	0,9216	0,9804	0.9495

Çizelge 13'de test veri setine uygulanan tekniklerin performans indikatörleri verilmiştir.

CTREE yöntemine göre test veri seti için oluşturulan modelde 51 CAT grubunun 47'sini CAT grubuna 4 tanesini kontrol grubuna atamıştır. 51 kontrol olgusunun 48'ini kontrol grubuna 3 tanesini CAT grubuna atamıştır. Oluşturulan modelin kappa istatistiği 0,8627, hassasiyeti 0,9216, Özgüllüğü 0,9412, F1 değeri 0,9307, doğruluk oranı 0,9314 ve %95 güven aralığı (0,8637 ile 0,9720) arasındadır.

Rastgele orman yöntemine göre test veri seti için oluşturulan modelde 51 CAT grubunun 50'sini CAT grubuna 1 tanesini kontrol grubuna atamıştır. 51 kontrol olgusunun 50'sini kontrol grubuna 1 tanesini CAT grubuna atamıştır. Oluşturulan modelin kappa istatistiği 0,9608, hassasiyeti 0,9804, Özgüllüğü 0,9804, F1 değeri 0,9804, doğruluk oranı 0,9706 ve %95 güven aralığı (0,9310 ile 0,9976) arasındadır.

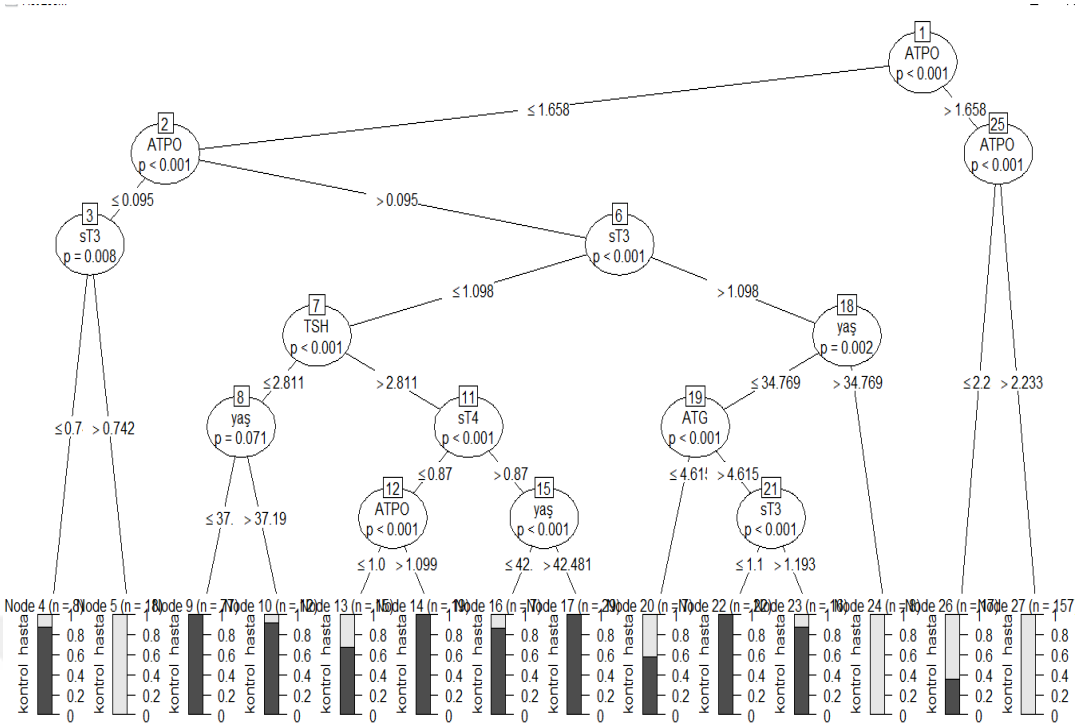
XGBOOST yöntemine göre test veri seti için oluşturulan modelde 51 CAT grubunun 46'sını CAT grubuna 5 tanesini kontrol grubuna atamıştır. 51 kontrol olgusunun 50'sini kontrol grubuna 1 tanesini CAT grubuna atamıştır. Oluşturulan modelin kappa istatistiği 0,8824, hassasiyeti 0,9216, Özgüllüğü 0,9608, F1 değeri 0,9400, doğruluk oranı 0,9412 ve %95 güven aralığı (0,8764 ile 0,9781) arasındadır.

C5.0 yöntemine göre test veri seti için oluşturulan modelde 51 CAT grubunun 49'unu CAT grubuna 2 tanesini kontrol grubuna atamıştır. 51 kontrol olgusunun 50'sini kontrol grubuna 1 tanesini CAT grubuna atamıştır. Oluşturulan modelin kappa istatistiği 0,9412, hassasiyeti 0,9608, Özgüllüğü 0,9804, F1 değeri 0,9703, doğruluk oranı 0,9706 ve %95 güven aralığı (0,9164 ile0,9939) arasındadır.

CART yöntemine göre test veri seti için oluşturulan modelde 51 CAT grubunun 47'sini CAT grubuna 4 tanesini kontrol grubuna atamıştır. 51 kontrol olgusunun 50'sini kontrol grubuna 1 tanesini CAT grubuna atamıştır. Oluşturulan modelin kappa istatistiği 0,8824, hassasiyeti 0,9216, Özgüllüğü 0,9804, F1 değeri 0,9495, doğruluk oranı 0,9510 ve %95 güven aralığı (0,8893 ile 0,9839) arasındadır.

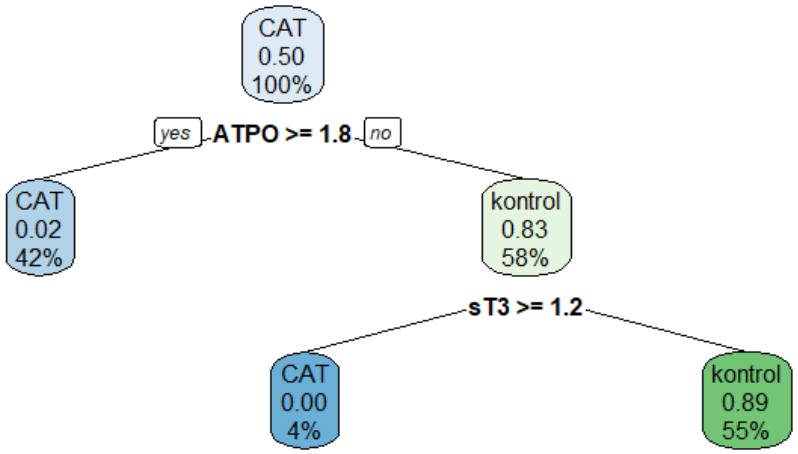
Test grubu için oluşturulan modellerde doğruluk değerleri ve diğer parametrelere göre en iyi sonucu veren yöntemler rastgele orman ve C5.0 karar ağacı algoritmasıdır. Yeni gelen bir hastanın sınıflandırma işlevi bu iki yöntem ile gerçekleştirilebilir. Modeller arasında en kötü performansı veren model ise 0,9314 doğruluk ile CTREE modelidir.

CTREE ve CART algoritmalarımızın karar ağacı grafikleri aşağıdaki gibi verilmiştir.



Şekil 6. CATveri2 için koşullu çıkarım ağacı

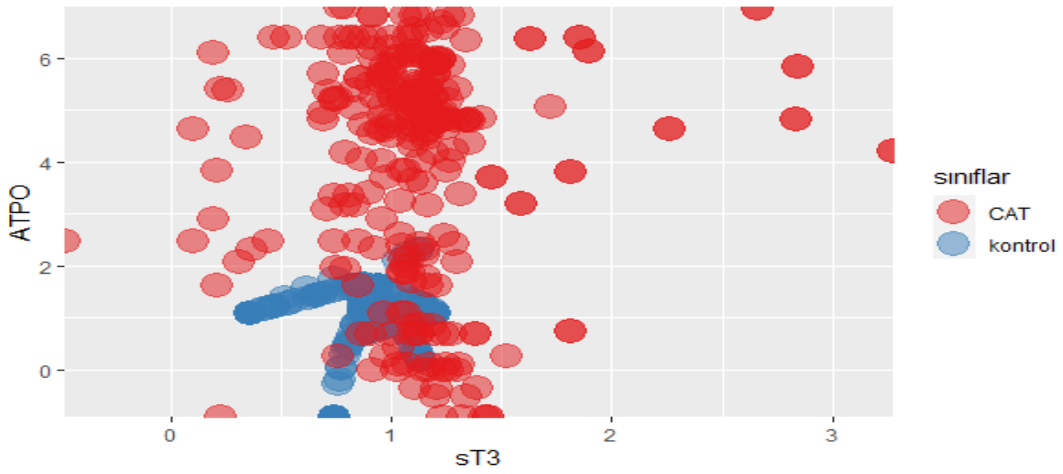
Şekil 6'daki koşullu çıkarım ağacı (CTREE) grafiği incelendiğinde ATPO değeri 1,658'den büyük olduğunda ATPO'nun bir başka belirleyici değerine bakılır. Bir sonraki adımda ATPO değeri 2,2'den küçük ya da eşit olduğunda deneklerin yaklaşık %40'ı CAT grubuna atanır. %60'ı ise kontrol grubuna atanır. ATPO değeri 2,2'den büyük olduğunda deneklerin tamamı kontrol grubuna atanır. ATPO değeri 1,658'den küçük ya da eşit olduğunda ATPO'nun bir başka belirleyici değerine bakılır. Bir sonraki adımda ATPO değeri 0,095'den küçük ya da eşit olduğunda sT3 değerine bakılır. sT3 değeri 0,7'den küçük ya da eşit olduğunda deneklerin %80'i CAT grubuna atanır. %20'si ise kontrol grubuna atanır. sT3 değeri 0,7'den büyük olduğunda deneklerin tamamı kontrol grubuna atanır.



Şekil 7. CATveri2 için CART karar ağacı

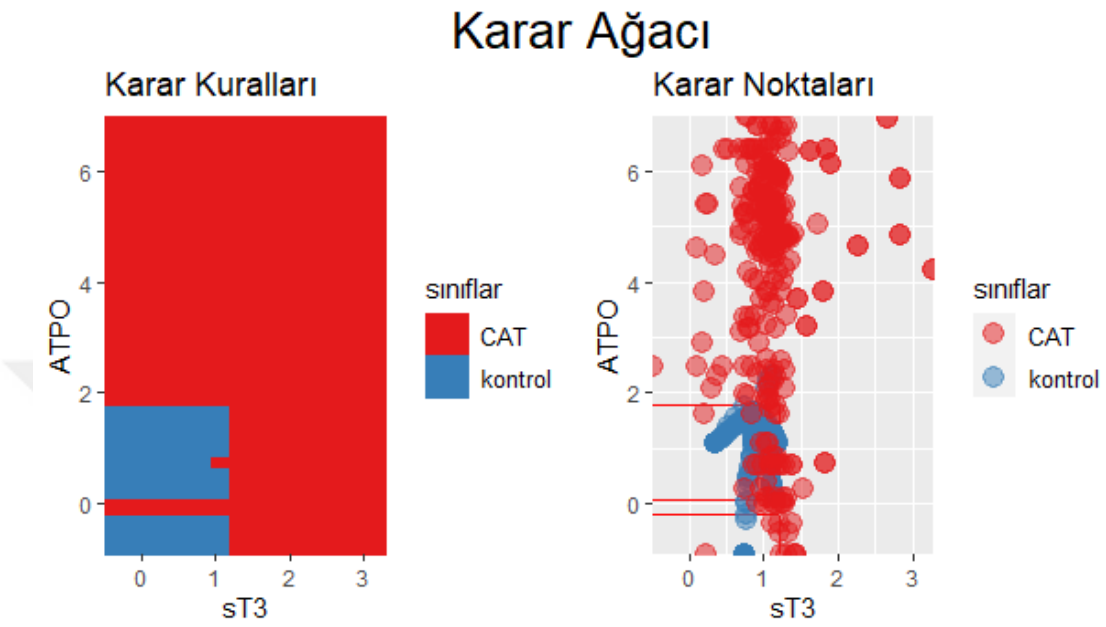
Şekil 7'deki CART karar ağacı incelendiğinde ATPO değeri 1,8'den büyük ya da eşit olunca 0,02 olasılık ile kontrol grubu olma ihtimaline karşın %42'si CAT olarak sınıflanır. 1,8'den küçük olunca 0,83 olasılık ile %58'i kontrol grubuna atanır. Daha sonra sınıflandırma sT3'e göre yapılır. sT3 değeri 1,2'ye eşit ya da büyük olunca 0 olasılık ile kontrol grubu olma ihtimaline karşın %4'ü CAT olarak tanımlar. 1,2'den küçük olunca 0,89 olasılık ile %55'ini kontrol grubuna sınıflandırır.

Kullanılan yöntemlerin karar analizlerinin görselleştirilmesi için CATveri2'de etkili olan "ATPO" ve "sT3" değişkenleri kullanılmıştır.



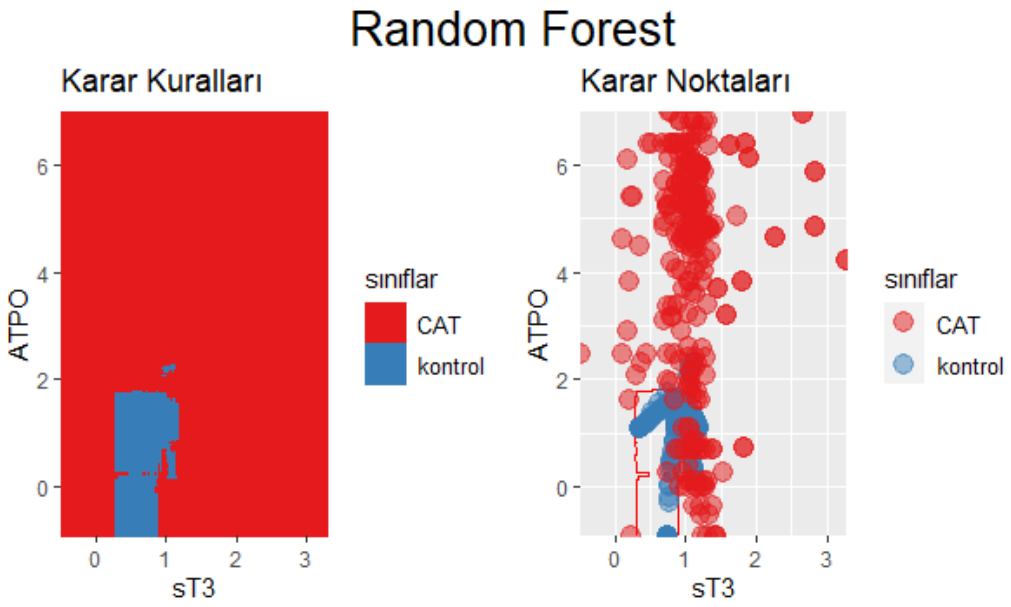
Şekil 8. CATveri2 önemli değişkenleri sınıflar üzerindeki dağılımları

Hekim şekil 8'i incelediğinde ATPO değişkeni 0 ile 2 arasında ve sT3 değişkeni 0 ile 1 arasında değer aldığında kontrol ve CAT sınıflarını ayırt edebilir. Yöntemlerin karar kuralları ve karar noktaları aşağıdaki gibidir.



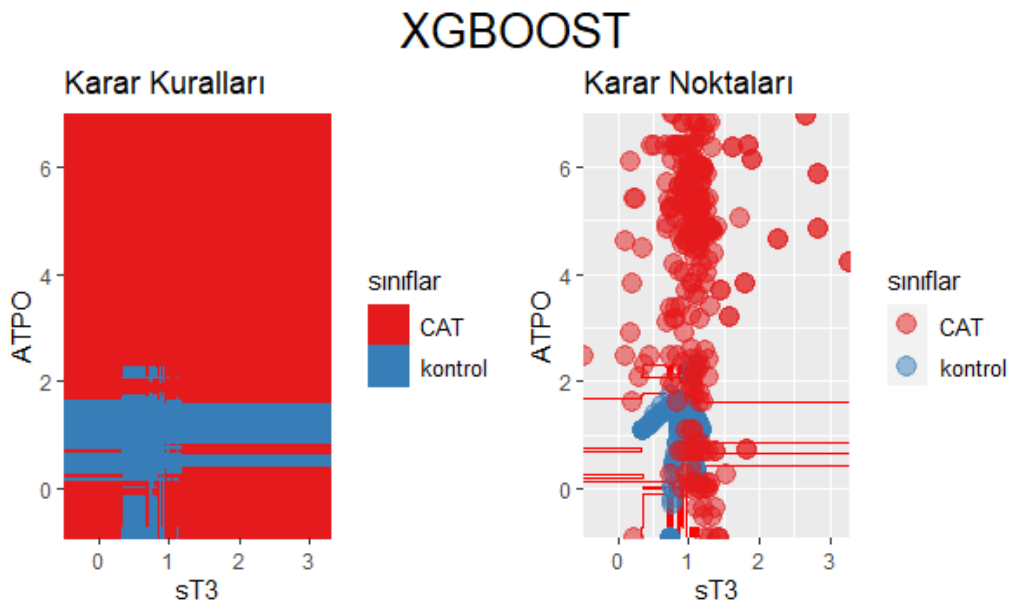
Şekil 9. CATveri2 karar ağacı için karar noktaları ve karar kuralları

Şekil 9'da koşullu çıkarım ağacı (CTREE) yöntemine göre oluşturulan karar kuralı ve karar noktaları incelendiğinde hekim sınıflandırma adımını basit bir şekilde belirleyebilir. ATPO değişkeni 0 ile 2 arasında ve sT3 değişkeni 0 ile 1 arasında değer aldığında kontrol ve CAT sınıfları rahatlıkla ayırt edilebilir. Yöntemin performansı iyi bir sonuç vermemiş olsa da karar kuralı basittir.



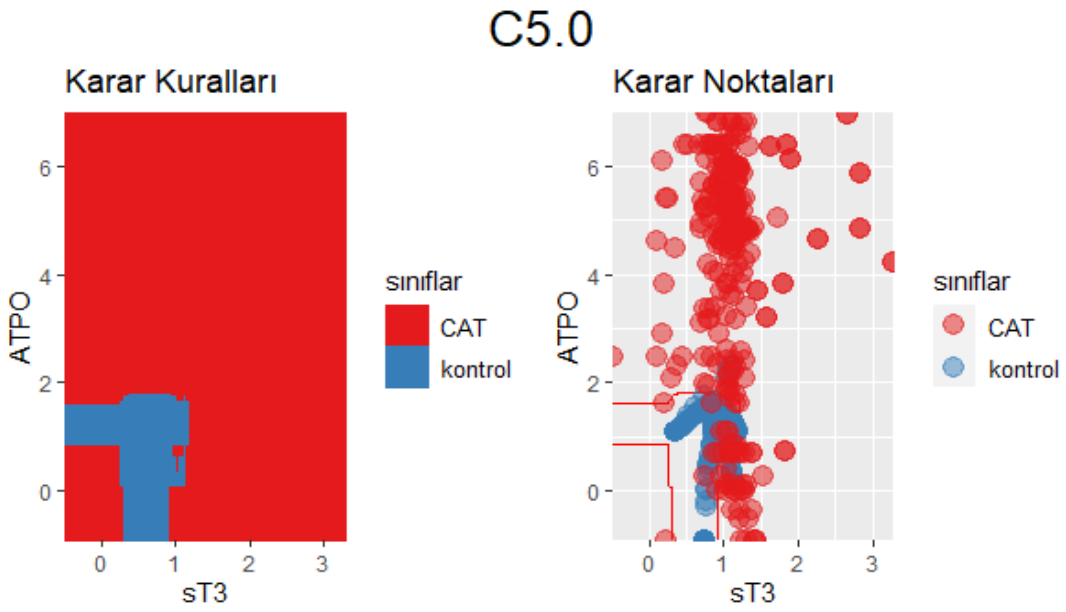
Şekil 10. CATveri2 rastgele orman için karar noktaları ve karar kuralları

Şekil 10'da rastgele orman tekniğine göre oluşturulan karar kuralı ve karar noktaları verilmiştir. Rastgele orman tekniğine göre ATPO değişkeni 0 ile 2 arasında ve sT3 değişkeni yaklaşık 0,5 ile 1 arasında değer aldığında kontrol grubu ayırt edilebilmektedir. Ancak karar kuralında kontrol grubu ve CAT grubunun iç içe olduğu durumlardan dolayı karmaşıklık söz konusudur. Bu karmaşık durumdan dolayı hekim rastgele orman tekniği ile karar vermekte zorluk çeker. Yöntemin performansı iyi bir sonuç vermiş olsa da karar kuralı karmaşıktır.



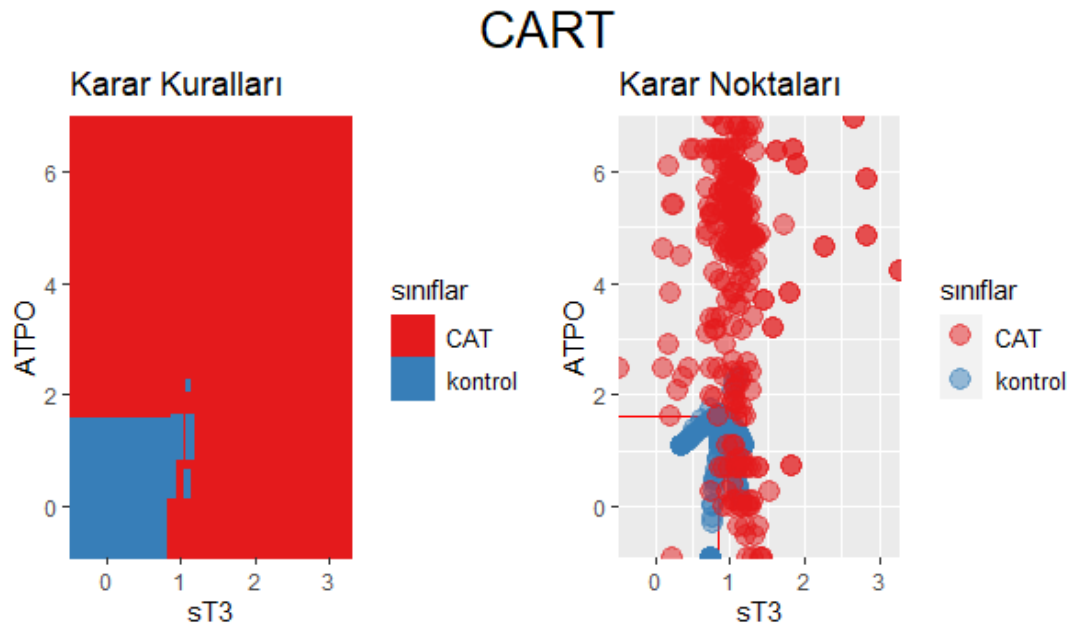
Şekil 11. CATveri2 xgboost için karar noktaları ve karar kuralları

Şekil 11'de XGBOOST tekniğine göre oluşturulan karar kuralı ve karar noktaları verilmiştir. XGBOOST tekniğine göre ATPO değişkeni 0 ile 2 arasında ve sT3 değişkeni tüm değerleri kontrol grubuna dahil edilmektedir. Bu durumda gözlem yapılmayan değerlerin kontrol olarak tanımı yapılmıştır. Bu karmaşık durumdan dolayı hekim XGBOOST tekniği ile karar vermekte zorluk çekebilir. Yöntemin performansı iyi bir sonuç vermiş olsa da karar kuralı diğer yöntemlerinkine göre karmaşıktır.



Şekil 12. CATveri2 C5.0 için karar noktaları ve karar kuralları

Şekil 12'de C5.0 karar ağacı algoritmasına göre oluşturulan karar kuralı ve karar noktaları incelendiğinde hekim sınıflandırma adımını basit bir şekilde belirleyebilir. ATPO değişkeni yaklaşık 1 ile 2 arasında ve sT3 değişkeni yaklaşık 0,5 ile 1 arasında değer aldığında kontrol ve CAT sınıfları rahatlıkla ayırt edilebilir. Yöntemin hem karar kuralı basittir hem de performansı iyi bir sonuç vermiştir. C5.0 karar ağacı algoritması önceliklidir.



Şekil 13. CATveri2 CART karar ağacı algoritması için karar noktaları ve karar kuralları

Şekil 13'de sınıflandırma ve regresyon ağaçları (CART) yöntemine göre oluşturulan karar kuralı ve karar noktaları incelendiğinde hekim sınıflandırma adımını basit bir şekilde belirleyebilir. ATPO değişkeni 0 ile 2 arasında ve sT3 değişkeni 0 ile 1 arasında değer aldığında kontrol ve CAT sınıfları rahatlıkla ayırt edilebilir. Yöntemin performansı iyi bir sonuç vermemiş olsa da karar kuralı basittir.

4.5. UCL Verisinin Eğitilmesi

UCL veri setinin %80'ni olan eğitim veri setini oluşturan 172 gözlem 5 nitelik ve 3 sınıf (tiroit, hipotiroit ve hipertiroit) üzerinden yöntemler uygulandığında doğruluk, kappa ve diğer parametrelerin istatistiklerine göz önüne alınarak her yöntem için en uygun modeller seçilmiştir.

Çizelge 14. UCL veri koşullu ağaç modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)		
Örnek boyutlarının özeti: 154, 155, 155, 154, 155, 155, ...		
Mincriterion	**Doğruluk**	**Kappa**
0.6576039	0.9175245	0.8176911
0.7601054	0.9175245	0.8176911
0.8535267	0.9175245	0.8176911
Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır.		
Model için kullanılan son değer mincriterion = 0,8535267.		

Çizelge 14'de eğitim verisi için oluşturulan 3 modelin doğruluk değerleri yaklaşık 0.918 olarak bulunmuştur. Kappa istatistikleri ise yaklaşık 0.818 olarak bulunmuştur. En iyi model 3. modeldir.

Çizelge 15. UCL veri rastgele orman modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)		
Örnek boyutlarının özeti: 155, 154, 155, 155, 156, 155, ...		
Ayarlama parametrelerinde sonuçları yeniden örnekleme:		
Mtry	**Doğruluk**	**Kappa**
4	0.9597631	0.9081142
5	0.9365196	0.8604013
Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır.		
Model için kullanılan son değer mtry =4'dir.		

Çizelge 15'de eğitim verisi için oluşturulan 2 model için doğruluk değerleri yaklaşık 0.960 ve 0.937 olarak bulunmuştur. Kappa istatistikleri ise yaklaşık olarak 0.909 ve 0.861 olarak bulunmuştur. Rastgele orman yönteminde Mtry göre en iyi model 1. modeldir.

Çizelge 16. UCL veri xgboost modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)								
Örnek büyüklüklerinin özeti: 156, 155, 155, 155, 155, 154, ...								
eta	Max Depth	Gamma	Colsample Bytree	Min Child Weigh	Subsample	nrounds	Doğruluk	Kappa
0.163	1	7.139	0.682	20	0.387	530	0.699	0.000
0.278	9	0.571	0.650	8	0.919	789	0.959	0.906
0.502	5	1.399	0.407	9	0.999	747	0.960	0.903
Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır.								
The final values used for the model were nrounds = 747, max_depth = 5, eta= 0.5016154, gamma = 1.399753, colsample_bytree = 0.4070044, min_child_weight =9 and subsample = 0.9999195.								

Çizelge 16'da eğitim veri seti için oluşturulan XGBOOST yöntemine göre üç modele ulaşılmıştır doğruluk, kappa, gamma, eta, nrounds ve diğer değerlere göre en iyi sonucu veren model 3. modeldir.

Çizelge 17. UCL veri C5.0 karar ağacı algoritması modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)				
Örnek büyüklüklerinin özeti: 155, 154, 155, 156, 154, 154, ...				
Model	**Winnow**	**Denemeler**	**Doğruluk**	**Kappa**
Rules	Doğru	76	0.8890114	0.7338529
Tree	Yanlış	28	0.9649510	0.9213094
Tree	Doğru	58	0.8890114	0.7338529
Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır.				
Model için kullanılan son değerler denemeler = 28, model = Tree ve winnow = yanlış.				

Çizelge 17'de eğitim verisi için yapılan C5.0 karar ağacı algoritmasına göre 3 model oluşturulmuştur. Doğruluk değeri ve diğer durumlara göre en uygun model 2. model seçilmiştir.

Çizelge 18. UCL veri CART karar ağacı algoritması modelinin eğitim verisi performans indikatörleri.

Yeniden Örnekleme: Çapraz Doğrulanmış (10 kat)		
Örnek boyutlarının özeti: 154, 156, 155, 155, 155, 155, ...		
cp	**Doğruluk**	**Kappa**
0.00000000	0.9012663	0.7712939
0.01923077	0.9012663	0.7712939
0.40384615	0.7515114	0.2963785
Doğruluk, en büyük değeri kullanarak en uygun modeli seçmek için kullanılmıştır.		
Model için kullanılan son değer cp =0,01923077'dır.		

Çizelge 18'de eğitim verisi için yapılan CART karar ağacı algoritmasına göre 3 model oluşturulmuştur. Doğruluk değeri ve CP=0,01923077 olduğu durumda en uygun model 2. model seçilmiştir.

4.6. UCL Verisinin Test Edilmesi

Test verisi için kalan 102 gözlem 7 nitelik üzerinden test veri setine uygulanan modellerin performanslarına ilişkin karmaşıklık matris parametrelerinin istatistikleri üzerinden en iyi yöntem seçilir.

Çizelge 19. UCL veri test verisi için uygulanan yöntemlerin performans indikatörleri.

Model		Tiroit	Hipertiroit	Hipotiroit
CTREE	Tiroit	29	2	1
	Hipertiroit	1	5	0
	Hipotiroit	0	0	5
Rastgele Orman	Tiroit	30	0	3
	Hipertiroit	0	7	0
	Hipotiroit	0	0	3
XGBOOST	Tiroit	30	2	2
	Hipertiroit	0	5	0
	Hipotiroit	0	0	4
C5.0	Tiroit	30	2	2
	Hipertiroit	0	5	0
	Hipotiroit	0	0	4
CART	Tiroit	28	0	3
	Hipertiroit	2	7	0
	Hipotiroit	0	0	3

Çizelge 19. Devamı.

Model/Parametre	Doğruluk	%95 Güven Aralığı	Kappa
CTREE	0,9070	0,7786 - 0,9741	0,7895
Rastgele Orman	0,9302	0,8094 - 0,9854	0,8371
XGBOOST	0,9070	0,7786 - 0,9741	0,7766
C5.0	0,9070	0,7786 - 0,9741	0,7766
CART	0,8837	0,7492 - 0,9611	0,7434

Çizelge 19. Devamı.

Model	Parametre/Sınıf	Tiroit	Hipertiroit	Hipotiroit
CTREE	Hassasiyet	0.9667	0.7143	0.8333
	Özgüllük	0.7692	0.9722	1.0000
	F1	0.9355	0.7692	0.9091
	Dengelenmiş Doğruluk	0.8679	0.8433	0.9167
Rastgele Orman	Hassasiyet	1.000	1.000	0.5000
	Özgüllük	0.7692	1.000	1.0000
	F1	0.9524	1.0000	0.66667
	Dengelenmiş Doğruluk	0.8846	1.0000	0.75000
XGBOOST	Hassasiyet	1.0000	0.7143	0.66667
	Özgüllük	0.6923	1.0000	1.00000
	F1	0.9375	0.8333	0.80000
	Dengelenmiş Doğruluk	0.8462	0.8571	0.83333
C5.0	Hassasiyet	1.0000	0.7143	0.66667
	Özgüllük	0.6923	1.0000	1.00000
	F1	0.9375	0.8333	0.80000
	Dengelenmiş Doğruluk	0.8462	0.8571	0.83333
CART	Hassasiyet	0.9333	1.0000	0.50000
	Özgüllük	0.7692	0.9444	1.00000
	F1	0.9180	0.8750	0.66667
	Dengelenmiş Doğruluk	0.8513	0.9722	0.75000

Çizelge 19'da test verisine uygulanan tekniklerin performans indikatörleri verilmiştir. CTREE yöntemine göre test veri seti için oluşturulan modelde 30 tiroit grubunun 29'u tiroit grubuna ve 1 tanesi hipertiroit grubuna atanmıştır. 7 hipertiroit grubunun 5 tanesi hipertiroit ve 2 tanesi tiroit grubuna atanmıştır. 6 hipotiroit grubunun 1 tanesi tiroit grubuna ve 5 tanesi hipotiroit grubuna atanmıştır. Oluşturulan modelin kappa istatistiği 0,7895, doğruluk oranı 0,9070 ve %95 güven aralığı (0,7786 ile 0,9741) arasındadır. Tiroit için hassasiyet 0,9667, özgüllük 0,7692, F1 değeri 0,9355 ve dengelenmiş doğruluk oranı 0,8679'dur. Hipertiroit için hassasiyet 0,7143, özgüllük 0,9722,F1 değeri 0,7692 ve dengeli doğruluk oranı 0,8433'dur. Hipotiroit için hassasiyet 0,8333, özgüllük 1,000, F1 değeri 0,9091 ve dengeli doğruluk oranı 0,9167'dir.

Rastgele orman yöntemine göre test veri seti için oluşturulan modelde 30 tiroit grubunun tamamı tiroit grubuna atanmıştır. 7 hipertiroit grubunun tamamı hipertiroit grubuna atanmıştır. 6 hipotiroit grubunun 3 tanesi tiroit grubuna ve 3 tanesi hipotiroit grubuna atanmıştır. Oluşturulan modelin kappa istatistiği 0,8371, doğruluk oranı 0,9302 ve %95 güven aralığı (0,8094 ile 0,9854) arasındadır. Tiroit için hassasiyet 1,000, özgüllük 0,7692, F1 değeri 0, 9524 ve dengelenmiş doğruluk oranı 0,8846'dır. Hipertiroit için hassasiyet 1,000, özgüllük 1,000, F1 değeri 1,000 ve dengeli doğruluk

oranı 1,000'dir. Hipotiroit için hassasiyet 0,5000, özgüllük 1,000, F1 değeri 0,6667 ve dengeli doğruluk oranı 0,7500'dür.

XGBOOST yöntemine göre test veri seti için oluşturulan modelde 30 tiroit grubunun tamamı tiroit grubuna atanmıştır. 7 hipertiroit grubunun 5 tanesi hipertiroit ve 2 tanesi tiroit grubuna atanmıştır. 6 hipotiroit grubunun 2 tanesi tiroit grubuna ve 4 tanesi hipotiroit grubuna atanmıştır. Oluşturulan modelin kappa istatistiği 0,8371, doğruluk oranı 0,9302 ve %95 güven aralığı (0,8094 ile 0,9854) arasındadır. Tiroit için hassasiyet 1,000, özgüllük 0,7692, F1 değeri 0, 9524 ve dengelenmiş doğruluk oranı 0,8846'dır. Hipertiroit için hassasiyet 1,000, özgüllük 1,000, F1 değeri 1,000 ve dengeli doğruluk oranı 1,000'dir. Hipotiroit için hassasiyet 0,5000, özgüllük 1,000, F1 değeri 0,6667 ve dengeli doğruluk oranı 0,7500'dür.

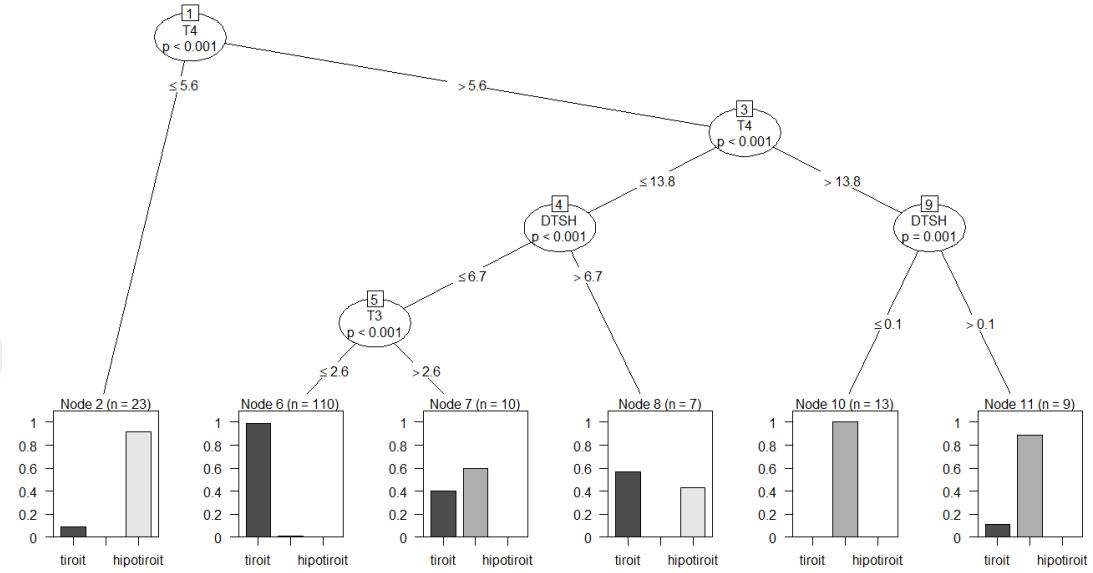
C5.0 yöntemine göre test veri seti için oluşturulan modelde 30 tiroit grubunun tamamı tiroit grubuna atanmıştır. 7 hipertiroit grubunun 5 tanesi hipertiroit ve 2 tanesi tiroit grubuna atanmıştır. 6 hipotiroit grubunun 2 tanesi tiroit grubuna ve 4 tanesi hipotiroit grubuna atanmıştır. Oluşturulan modelin kappa istatistiği 0,8371, doğruluk oranı 0,9302 ve %95 güven aralığı (0,8094 ile 0,9854) arasındadır. Tiroit için hassasiyet 1,000, özgüllük 0,7692, F1 değeri 0, 9524 ve dengelenmiş doğruluk oranı 0,8846'dır. Hipertiroit için hassasiyet 1,000, özgüllük 1,000, F1 değeri 1,000 ve dengeli doğruluk oranı 1,000'dir. Hipotiroit için hassasiyet 0,5000, özgüllük 1,000, F1 değeri 0,6667 ve dengeli doğruluk oranı 0,7500'dür.

CART yöntemine göre test veri seti için oluşturulan modelde 30 tiroit grubunun 28 tanesi tiroit grubuna ve 2 tanesi hipertiroit grubuna atanmıştır. 7 hipertiroit grubunun tamamı hipertiroit grubuna atanmıştır. 6 hipotiroit grubunun 3 tanesi tiroit grubuna ve 3 tanesi hipotiroit grubuna atanmıştır. Oluşturulan modelin kappa istatistiği 0,7434, doğruluk oranı 0,8837 ve %95 güven aralığı (0,7492 ile 0,9611) arasındadır. Tiroit için hassasiyet 0, 9333, özgüllük 0,7692, F1 değeri 0,9180 ve dengelenmiş doğruluk oranı 0,8513'dür. Hipertiroit için hassasiyet 1,000, özgüllük 0,9444 F1 değeri 0,8450 ve dengeli doğruluk oranı 0,9722'dir. Hipotiroit için hassasiyet 0,5000, özgüllük 1,000, F1 değeri 0,6667 ve dengeli doğruluk oranı 0,7500'dür.

Test grubu için oluşturulan modellerde doğruluk değerleri ve kappa istatistiklerine göre en iyi sonucu veren yöntemler rastgele orman ve CTREE karar ağacı algoritmasıdır. Yeni gelen bir hastanın sınıflandırma işlevi bu iki yöntem ile

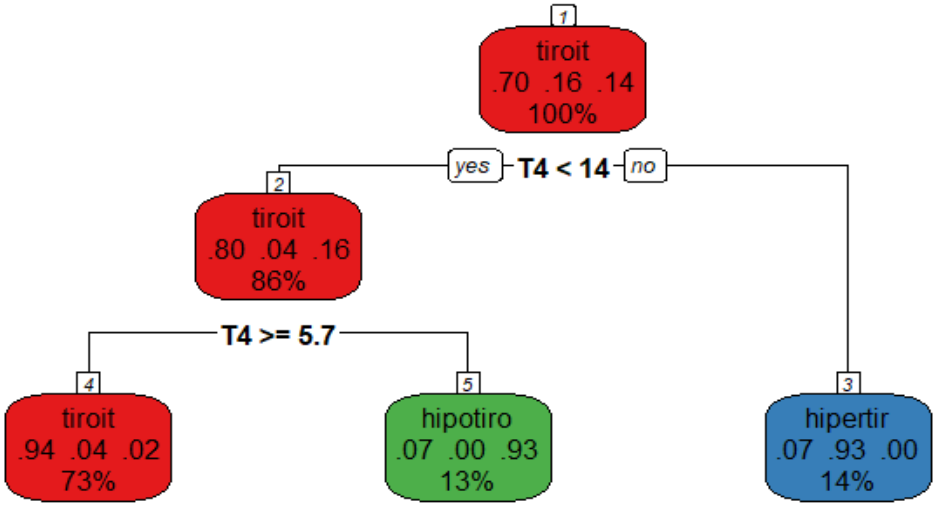
gerçekleştirilebilir. Modeller arasında en kötü performansı veren model ise 0,8837 doğruluk oranı ile CART modelidir.

CTREE ve CART algoritmalarımızın karar ağacı grafikleri aşağıdaki gibi verilmiştir.



Şekil 14. UCL verisi için koşullu çıkarım ağacı.

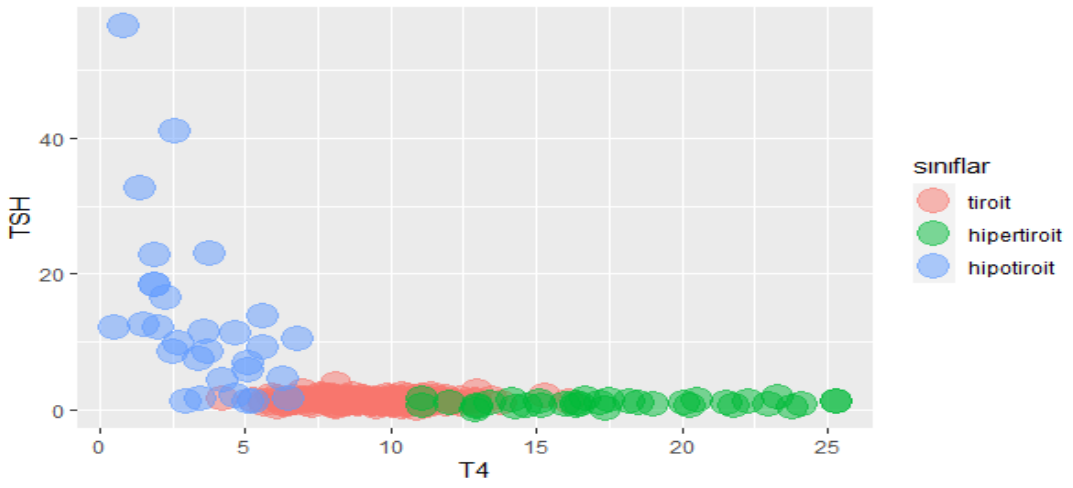
Şekil 14'deki koşullu çıkarım ağacı (CTREE) incelendiğinde hastanın T4 hormonu 5,6'dan küçük ya da eşit olduğunda deneklerin %90'ı hipotiroit grubuna atanmıştır. %10'u ise tiroit grubuna atanmıştır. T4 değeri 5,6'dan büyük olduğu zaman tekrardan T4'ün bir başka değerine bakılır. T4 değeri 13,8'e eşit ya da küçük olduğu zaman DTSH değerine bakılır. DTSH değeri 6,7'den büyük ise deneklerin %60'ı tiroit grubuna atanmıştır. %40'ı ise hipotiroit grubuna atanmıştır. DTSH değeri 6,7'den küçük ya da eşit ise T3 değerine bakılır. T3 değeri 2,6'dan küçük ya da eşit ise deneklerin yaklaşık %99'u tiroit grubuna atanmıştır. Yaklaşık %1'i ise hipertiroit grubuna atanmıştır. T3 değeri 2,6'dan büyük ise %40'ı tiroit grubuna atanmıştır. %60'ı ise hipertiroit grubuna atanmıştır.



Şekil 15. UCL verisi için CART karar ağacı.

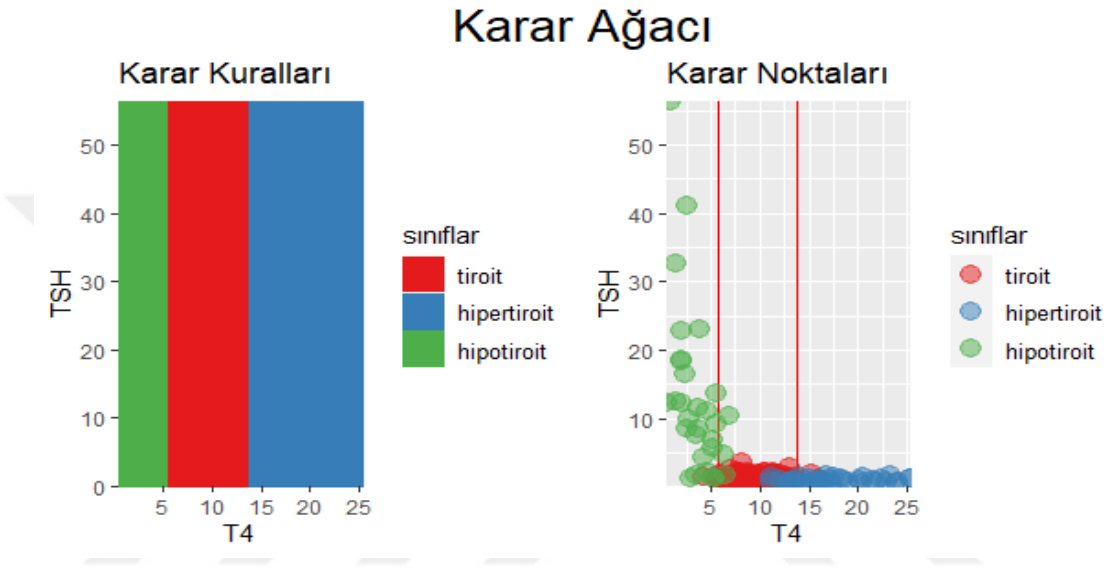
Şekil 15'deki CART karar ağacı incelendiğinde deneğin T4 hormonu 14'den büyük olursa %14 olasılık ile hipertiroit, 14'den küçük olursa tiroit olur ve tekrar T4 hormonun bir başka değerine bakılır. T4 hormonu 5,7'den büyük olursa denek %73 olasılık ile tiroit. T4 hormonu 5,7'den küçük olursa %13 olasılık ile hipotiroit olarak sınıflanır.

Kullanılan yöntemlerin karar analizlerinin görselleştirilmesi için UCL veride etkili olan "TSH" ve "T4" değişkenlerini kullanılmıştır.



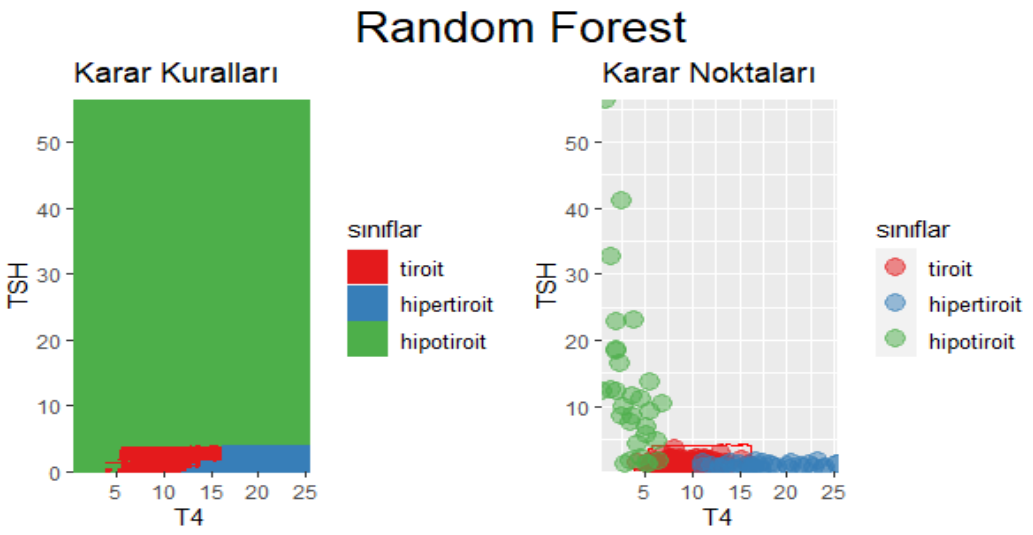
Şekil 16. UCL verisi önemli değişkenleri sınıflar üzerindeki dağılımları.

Hekim şekil 16'yı incelediğinde T4 hormonu yaklaşık 0 ile 5 arasında değer aldığında denek hipotiroit, yaklaşık 6 ile 12 arası değer aldığında denek tiroit ve yaklaşık 13'den büyük bir değer aldığında hipertiroit olarak karar verebilir. TSH'a göre bu ayrımı sadece hipotiroit sınıfı için yapabilir çünkü yaklaşık 0 ile 10 arası hariç diğer aralıklarda sınıf yoğunluğu fazladır. Yöntemlerin karar kuralları ve karar noktaları aşağıdaki gibidir.



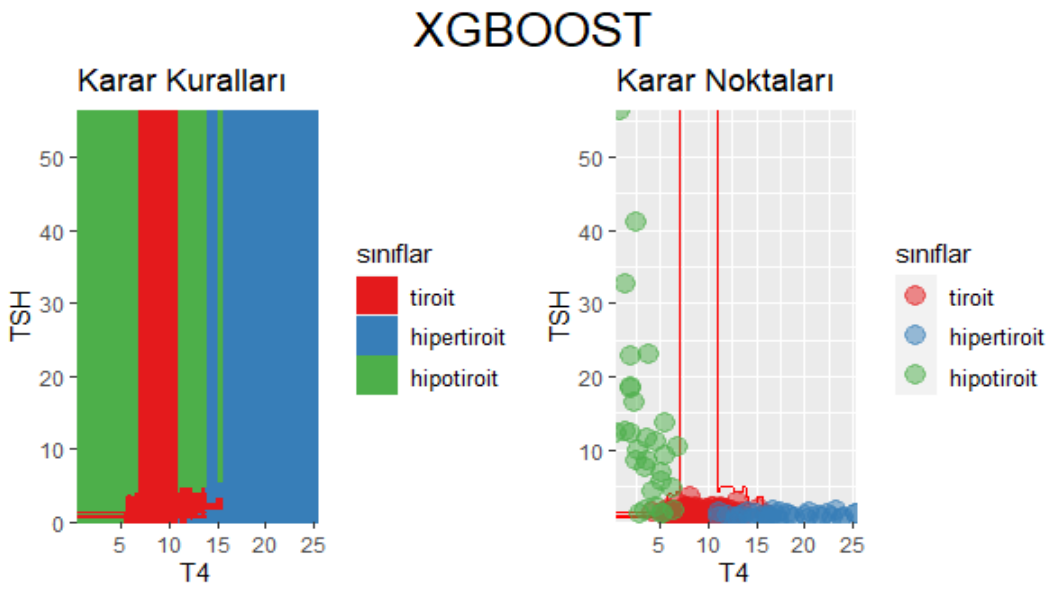
Şekil 17. UCL verisi koşullu ağaca göre karar analizi.

Şekil 17'de koşullu çıkarım ağacına (CTREE) göre oluşturulan karar kuralı ve karar noktaları incelendiğinde hekim sınıflandırma adımını basit bir şekilde belirleyebilir. T4 değeri yaklaşık 0 ile 5 arasında değer aldığında hipotiroit, T4 değeri yaklaşık 6 ile 13 arası değer aldığında tiroit ve T4 değeri yaklaşık 14 ile 25 arası değer aldığında hipertiroit olarak sınıflanır. TSH'a göre incelendiğinde tüm sınıflar karar kuralında gözlemlenmektedir. Ancak karar noktaları incelendiğinde sadece hipotiroit tüm aralıklarda yer alırken, tiroit ve hipertiroit 0 ile 10 arasında yer almaktadır. Yöntemin performansı iyi bir sonuç vermemiş olsa da karar kuralı basittir.



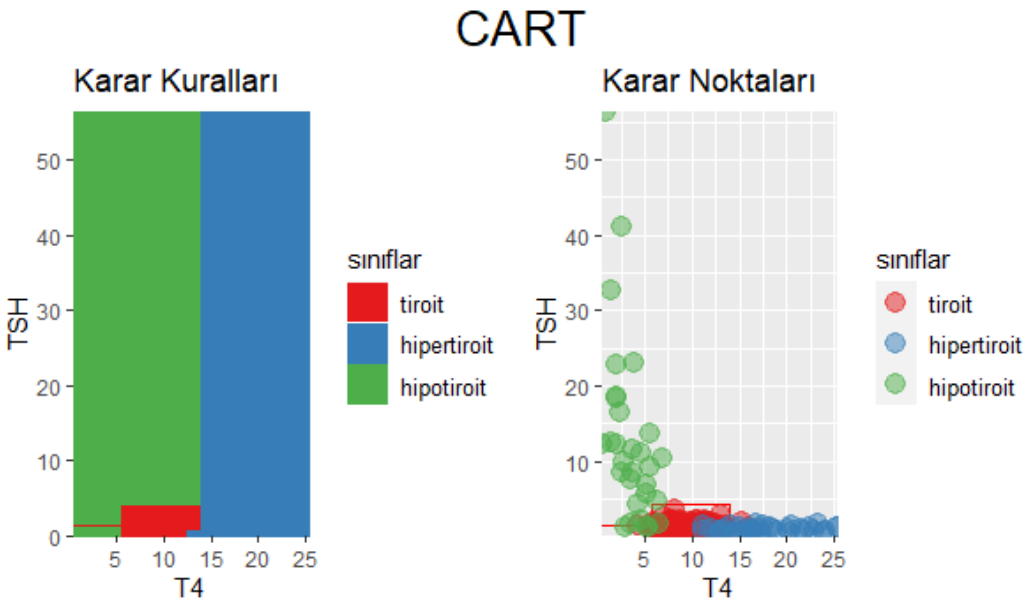
Şekil 18. UCL verisi rastgele orman karar analizi.

Şekil 18'de rastgele orman yöntemine göre oluşturulan karar kuralı ve karar noktaları verilmiştir. Rastgele orman tekniğine göre T4 değeri yaklaşık 0 ile 4 arasında değer aldığında hipotiroit, T4 değeri yaklaşık 5 ile 15 arası değer aldığında tiroit ve T4 değeri yaklaşık 11 ile 25 arası değer aldığında hipertiroit olarak sınıflanır. TSH'a göre incelendiğinde tüm sınıflar karar kuralında gözlemlenmektedir. Ancak karar noktaları incelendiğinde sadece hipotiroit tüm aralıklarda yer alırken, tiroit ve hipertiroit yaklaşık 0 ile 5 arasında yer almaktadır. Hipotiroit olarak sınıflandırılan olguların gözlemlenmediği bölge hipotiroit olarak belirlenmiştir. Yöntemin performansı iyi bir sonuç vermiş olsa da karar kuralı karmaşıktır. Bu karmaşıklıktan dolayı hekim karar vermekte zorluk çekebilir.



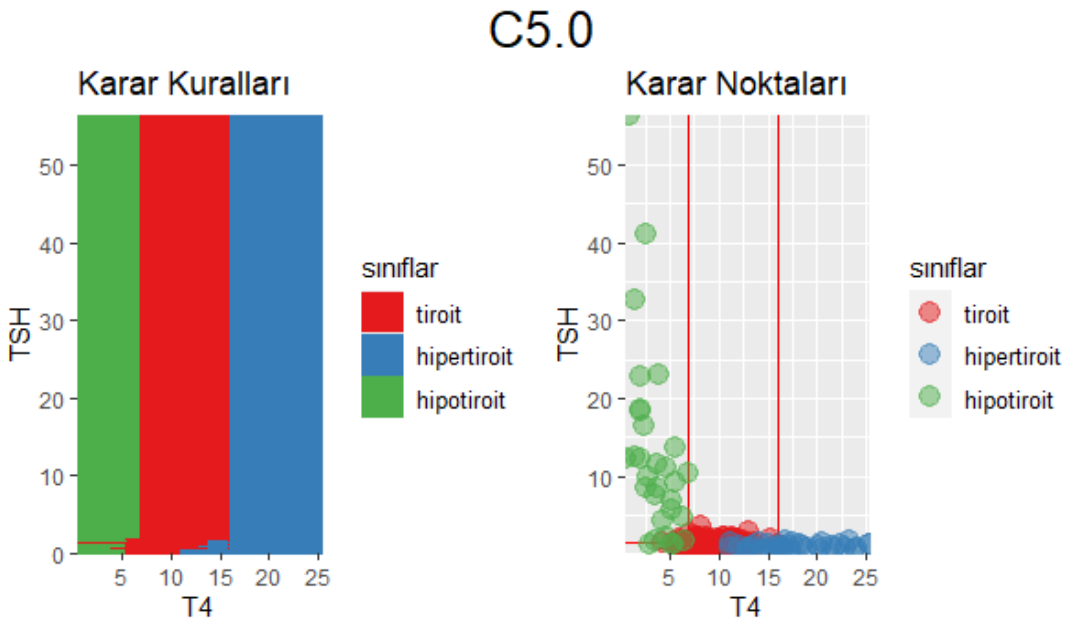
Şekil 19. UCL verisi xgboost karar analizi.

Şekil 19'da xgboost yöntemine göre oluşturulan karar kuralı ve karar noktaları verilmiştir. Xgboost tekniğine göre T4 değeri yaklaşık 0 ile 5 arasında değer aldığında hipotiroit, T4 değeri yaklaşık 6 ile 15 arası değer aldığında tiroit ve T4 değeri yaklaşık 11 ile 25 arası değer aldığında hipertiroit olarak sınıflanır. TSH'a göre incelendiğinde tüm sınıflar karar kuralında gözlemlenmektedir. Ancak karar noktaları incelendiğinde sadece hipotiroit tüm aralıklarda yer alırken, tiroit ve hipertiroit yaklaşık 0 ile 5 arasında yer almaktadır. Hipotiroit olarak sınıflandırılan olguların gözlemlenmediği bölge hipotiroit olarak belirlenmiştir. Yöntemin performansı iyi bir sonuç vermiş olsa da karar kuralı karmaşıktır. Bu karmaşıklıktan dolayı hekim karar vermekte zorluk çekebilir.



Şekil 20. UCL verisi CART karar ağacı karar analizi.

Şekil 20'de sınıflandırma ve regresyon ağacına (CART) göre oluşturulan karar kuralı ve karar noktaları incelendiğinde hekim sınıflandırma adımını basit bir şekilde belirleyebilir. T4 değeri yaklaşık 0 ile 5 arasında değer aldığında hipotiroit, T4 değeri yaklaşık 6 ile 12 arası değer aldığında tiroit ve T4 değeri yaklaşık 10 ile 25 arası değer aldığında hipertiroit olarak sınıflanır. TSH'a göre incelendiğinde tüm sınıflar karar kuralında gözlemlenmektedir. Ancak karar noktaları incelendiğinde sadece hipotiroit tüm aralıklarda yer alırken, tiroit ve hipertiroit yaklaşık 0 ile 5 arasında yer almaktadır. Yöntemin performansı iyi bir sonuç vermemiş olsa da karar kuralı basittir.



Şekil 21. UCL verisi C5.0 karar ağacı karar analizi.

Şekil 21'de C5.0 karar ağacı algoritmasına göre oluşturulan karar kuralı ve karar noktaları incelendiğinde hekim sınıflandırma adımını basit bir şekilde belirleyebilir. T4 değeri yaklaşık 0 ile 5 arasında değer aldığında hipotiroit, T4 değeri yaklaşık 6 ile 15 arası değer aldığında tiroit ve T4 değeri yaklaşık 11 ile 25 arası değer aldığında hipertiroit olarak sınıflanır. TSH'a göre incelendiğinde tüm sınıflar karar kuralında gözlemlenmektedir. Ancak karar noktaları incelendiğinde sadece hipotiroit tüm aralıklarda yer alırken, tiroit ve hipertiroit yaklaşık 0 ile 3 arasında yer almaktadır. Yöntemin performansı iyi bir sonuç vermemiş olsa da karar kuralı basittir.

5. SONUÇLAR

Sağlık alanında oluşan büyük veri setleri teknolojinin de gelişmesi ile birlikte makine öğrenmesi yöntemleri ile birlikte anlaşılabilir ve yorumlanabilir hale getirilebilmektedir. Tıbbi tanımın konması, hastaya ait doğru veriler, doğru tıp literatür bilgisi ve klinik tecrübeler gerektiren, uzmanlar açısından oldukça kritik bir süreçtir. Sınıf dengesizliği problemi bulunan veri setlerine doğrudan makine öğrenmesi yöntemlerinin uygulanması sınıflandırma performansı üzerinde oldukça önemli etkilere sahiptir.

Bu tez çalışmasında Muğla Sıtkı Koçman Üniversitesi Eğitim ve Araştırma hastanesinde toplanan kronik otoimmün tiroit veri setine sentetik azınlık yüksek hızla öğrenme tekniği (SMOTE) uygulanmıştır. Daha sonra elde edilen veri ile birlikte University of California Irvine Machine Learning Repository sitesinden çekilen açık veri setine makine öğrenmesi yöntemleri uygulanmıştır. Araştırmada kullanılan makine öğrenmesi yöntemleri sınıflandırma ve regresyon ağaçları (CART), C5.0 karar ağacı algoritması, koşullu çıkarım ağacı (CTREE), rastgele orman ve aşırı gradyan arttırma (XGBOOST) yöntemleri uygulanmıştır. Uygulamalar açık kaynak kodlu R program dili kütüphaneleri ile gerçekleştirilmiştir (Dogu E., 2019). Kronik otoimmün tiroit veri setinden iki farklı veri seti oluşturulmuştur. On altı nitelikten oluşan kronik otoimmün tiroit ham veri seti ve sekiz nitelikten oluşan kronik otoimmün tiroit veri seti iki olarak tanımlanmıştır. Açık veri seti ise UCL verisi olarak tanımlanmıştır. Üç veri kümesine de uygulanan yöntemlerin sonuçları verilmiştir. Kronik otoimmün tiroit ham veri setine yapılan uygulama sonucunda ultrason görüntülerinin olduğu niteliklerden dolayı bütün yöntemler iyi sonuç vermiştir. Değişken elemesi yapılarak oluşturulan kronik otoimmün tiroit veri seti 2'de yapılan uygulamalarda en iyi sonucu veren yöntemler rastgele orman ve C5.0 karar ağacı algoritmasıdır. UCL veri setine yapılan uygulama sonucunda C5.0 karar ağacı algoritması, koşullu çıkarım ağacı (CTREE), aşırı gradyan arttırma (XGBOOST) ve rastgele orman teknikleri iyi sonuç vermiştir ancak regresyon ve sınıflandırma ağaçları (CART) diğer tekniklere göre daha düşük sonuç vermiştir.

CATveri2 veri seti ve UCL veri setinde sınıflandırmada etkili olan iki değişken ile karar analizleri yapılmıştır. CATveri2 veri setine uygulanan karar analizi sonuçlarına

göre C5.0 karar ağacı algoritması, koşullu çıkarım ağacı (CTREE) ile sınıflandırma ve regresyon ağacı (CART) tekniklerinin karar kuralları basit olduğu için sınıflandırmada tercih edilebilirler.

UCL veri setine uygulanan karar analizi sonuçlarına göre C5.0 karar ağacı algoritması, koşullu çıkarım ağacı (CTREE) ile sınıflandırma ve regresyon ağacı (CART) tekniklerinin karar kuralları basit olduğu için sınıflandırmada tercih edilebilirler.

KAYNAKLAR

Adak, M. F., ve Yurtay, N. (2013). Gini algoritmasını kullanarak karar ağacı oluşturmayı sağlayan bir yazılımın geliştirilmesi. *Bilişim Teknolojileri Dergisi*, *6*(3), 1-6.

Akgül, G., Çelik, A. A., Aydın, Z. E., ve Öztürk, Z. K. (2020). Hipotiroidi Hastalığı Teşhisinde Sınıflandırma Algoritmalarının Kullanımı. *Bilişim Teknolojileri Dergisi*, *13*(3), 255-268.

Alan A. ve Kabatak M. (2020). Veri Seti-Sınıflandırma İlişkisinde Performansa Etki Eden Faktörlerin Değerlendirilmesi. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, *32*(2), 531-540.

Alshari H. Saleh A. ve Odabas A. (2021). CPU Performansı için Gradyan Artırıcı Karar Ağacı Algoritmalarının Karşılaştırılması. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi*, *37*(1), 157-168.

Banu G. R. (2016). A Role of decision Tree classification data Mining Technique in Diagnosing Thyroid disease. *International Journal of Computer Sciences and Engineering*, *4*(11), 64-70.

Bao, W. Lianju, N. ve Yue K. (2019). Integration of unsupervised and supervised machine learning algorithms for credit risk assessment. *Expert Systems with Applications*, *128*, 301–315.

Barnes, J. (2015). Azure Machine Learning Microsoft Azure Essentials Livro. In *Microsoft Press*.

Bayer H. ve Çoban T., (2015). *Web İstatistiklerinde Makine Öğrenmesi Algoritmaları İle Kritik Parametre Tespiti*.

Begum A. ve Parkavi A. (2019, March). Prediction of thyroid disease using data mining techniques. In *2019 5th International Conference on Advanced Computing and Communication Systems (ICACCS)* (pp. 342-345).

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Budholiya K. Shrivastava S. K., ve Sharma V. (2020). An optimized XGBoost based diagnostic system for effective prediction of heart disease. *Journal of King Saud University-Computer and Information Sciences*.

Burdur Z. ve Atay E. C., (2018). An Analysis Of Pesticide Use For Cotton Production Through Data Mining: The Case Of Nazilli. *Anadolu Üniversitesi Bilim Ve Teknoloji Dergisi A-Uygulamalı Bilimler ve Mühendislik*, 2018, 19.3: 732-747.

Chawla N. V. Bowyer K. W. Hall L. O. ve Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

Dogu E., (2019). *Classification With Caret*. 8374.

Emir Ş. (2013). Yapay sinir ağları ve destek vektör makineleri yöntemlerinin sınıflandırma performanslarının karşılaştırılması: borsa endeks yönünün tahmini üzerine bir uygulama. *İstanbul Üniversitesi.*

Farboudi S. (2009). Tıp Bilişiminde İstatistiksel Veri Madenciliği, *Yüksek Lisans Tezi, Hacette Üniversitesi, Ankara*.

Gordon, A. D., Breiman, L., Friedman, J. H., Olshen, R. A., ve Stone, C. J. (1984). Classification and Regression Trees.

Gottipati S., Shankararaman V., ve Lin, J. R. (2018). Text analytics approach to extract course improvement suggestions from students' feedback. *Research and Practice in Technology Enhanced Learning*, *13*(1), 1-19.

Gül S. S., (2020). *Tiroit Hastalıklarında Multidisipliner Yaklaşım*. *9*(January), 222–234.

Gümüşçü A., Taşlatın R., ve Aydilek İ. B. C4. 5 Decision Tree Pruning Using Genetic Algorithm. *Dicle Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *5*(2), 77-80.

Güngör O. ve Akar Ö., (2012). Rastgele orman algoritması kullanılarak çok bantlı görüntülerin sınıflandırılması. *Jeodezi ve Jeoinformasyon Dergisi*, (106), 139-146.

Haciefendioğlu Ş. (2012). *Makine öğrenmesi yöntemleri ile glokom hastalığının*

teşhisi (Doctoral dissertation, Selçuk Üniversitesi Fen Bilimleri Enstitüsü).

İğci E. ve Göktay Y., (1996). *Tiroid Bezi Ultrasonografisi*. *Türkiye Ekopatoloji Dergisi; Cilt: 2 Sayı: 1-2*.

Kaba M., (2013). Gebelik ve Tiroid Hormonları. *Kocatepe Tıp Dergisi*, *14*(3), 160-166.

Kaya, T. (2015). Makine Öğrenme Yöntemleri İle Trafik Kazaları İçin Risk Tahmini Yapabilen Web Tabanlı Bir Yazılım. *Yüksek Lisans Tezi Bilgisayar Eğitimi Anabilim Dalı*.

Koçdağ H. M. Y., (2013). *Mutabakat Fonksiyonu Kullanılarak İş Akışları Optimizasyonu*.

Korkem E. (2013). Mikroarray Gen Ekspresyon Veri Setlerinde Random Forest Ve Naive Bayes Sınıflama Yöntemleri Yaklaşım.

Kousarrizi M. N., Seiti F., ve Teshnehlab M. (2012). An experimental comparative study on thyroid disease diagnosis based on feature subset selection and classification. *International Journal of Electrical ve Computer Sciences IJECS-IJENS*, *12*(01), 13-20.

Lezki, Ş. (2014). Çok Kriterli Karar Verme Problemlerinde Karar Ağacı Kullanımı. *İktisadi Yenilik Dergisi*, *2*(1), 16–31.

Malinin A., Prokhorenkova L., ve Ustimenko, A. (2020). Uncertainty in gradient boosting via ensembles.

Margret J., Lakshmipathi B., ve Kumar S. A. (2012). Diagnosis of thyroid disorders using decision tree splitting rules. *International Journal of Computer Applications*, *44*(8), 43-46.

Namazkhan M., Albers C., ve Steg L. (2020). A decision tree method for explaining household gas consumption: The role of building characteristics, socio-demographic variables, psychological factors and household behaviour. *Renewable and Sustainable Energy Reviews*, *119*, 109542.

Özdemir S. (2018). Potential distribution modelling and mapping using Random Forest method: an example of Yukarıgökdere district. *Turkish Journal of Forestry*, *19*(1), 51-56.

Özer D. Z., Güngör S. N., ve Şimşekli Y. (2011). Sınıf öğretmenliği öğrencilerinin biyoloji deneylerini uygulayabilme ve bilimsel süreç becerilerini analiz edebilme yeterlilikleri. *Uludağ Üniversitesi Eğitim Fakültesi Dergisi*, *24*(2), 563-580.

Özlüer B. B., Yangın M., ve Sarıdaş E. S. (2021). Makine Öğrenmesi Teknikleriyle Diyabet Hastalığının Sınıflandırılması. *Journal of Natural and Applied Sciences*, *25*(1).

Pandya R., ve Pandya J. (2015). C5. 0 algorithm to improved decision tree with feature selection and reduced error pruning. *International Journal of Computer Applications*, *117*(16), 18-21.

Peng W., Chen J., ve Zhou H. (2009). An implementation of ID3-decision tree learning algorithm. *From web. arch. usyd. edu. au/wpeng/DecisionTree2. pdf Retrieved date: May*, *13*.

Rezapour M., Molan A. M., ve Ksaibati K. (2020). Analyzing injury severity of motorcycle at-fault crashes using machine learning techniques, decision tree and logistic regression models. *International journal of transportation science and technology*, *9*(2), 89-99.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, *3*(3), 210-229.

Sevli O. (2019). Göğüs Kanseri Teşhisinde Farklı Makine Öğrenmesi Tekniklerinin Performans Karşılaştırması. *Avrupa Bilim ve Teknoloji Dergisi*, (16), 176-185.

Sharma N., Sharma R., ve Jindal N. (2021). Machine Learning and Deep Learning Applications-A Vision. *Global Transitions Proceedings*, *2*(1), 24-28.

Sidiq U. D., Aaqib S. M. ve Khan R. A. (2019). Veri madenciliği sınıflandırma teknikleri kullanılarak çeşitli tiroid rahatsızlıklarının teşhisi. *Int J Sci Res Coput Sci Inf Technol* , *5* , 131-6.

Siknun G. P. ve Sitanggang I. S. (2016). Web-based classification application for forest fire data using the shiny framework and the C5. 0 algorithm. *Procedia Environmental Sciences*, *33*, 332-339.

Singh S., ve Gupta P. (2014). Comparative study ID3, cart and C4. 5 decision tree algorithm: a survey. *International Journal of Advanced Information Science and*

Technology (IJAIST), *27*(27), 97-103.

Solmaz R., Günay M., ve Alkan A. (2013). Uzman sistemlerin tiroit teşhisinde kullanılması. *XV. Akademik Bilişim Konferansı Bildirileri*, 23-25.

Şanlıtürk E., (2018). *Makine Öğrenme Algoritmalarıyla Hatalı Ürün Tahmini. 1*, 43.

Thongsuwan S., Jaiyen S., Padcharoen A., ve Agarwal P. (2021). ConvXGB: A new deep learning model for classification problems based on CNN and XGBoost. *Nuclear Engineering and Technology*, *53*(2), 522-531.

Tosun C. F., (2013). Tiroit Sintigrafisi. *Journal of Experimental and Clinical Medicine*, *29*(4S), 289–300.

Uzunlu B. Y., ve Hussain S. M. (2020). Employing Machine Learning Algorithms to build Trading Strategies with higher than Risk-Free Returns. *International Econometric Review*, *12*(2), 112-138.

Üstüner M., Abdikan S., Bilgin G., ve Şanlı F. B. (2020). Hafif Gradyan Artırma Makineleri ile Tarımsal Ürünlerin Sınıflandırılması. *Türk Uzaktan Algılama ve CBS Dergisi*, *1*(2), 97-105.

Yabacı A. (2017). *Sağkalım verilerinde kullanılan ağaç tabanlı yöntemlerin karşılaştırılması* (Master's thesis, Uludağ Üniversitesi).

Yakut E. ve Gemici E. (2017). LR, C5. 0, CART, DVM Yöntemlerini kullanarak hisse senedi getiri sınıflandırma tahmini yapılması ve kullanılan yöntemlerin karşılaştırılması: Türkiye'de BIST'de bir uygulama.

Yangın G. (2019). *Xgboost ve Karar Ağacı Tabanlı Algoritmaların Diyabet Veri Setleri Üzerine Uygulanması*.

Yavaş M., Güran A., ve Uysal M. Covid-19 Veri Kümesinin SMOTE Tabanlı Örnekleme Yöntemi Uygulanarak Sınıflandırılması. *Avrupa Bilim ve Teknoloji Dergisi*, 258-264.

Yıldız A., 2019. (2019). Makine Öğrenmesi Yöntemleri İle Tiroit Hastalığının Teşhisi.

Yüce T. ve Kabak M. (2021). Makine Öğrenmesi Algoritmaları ile Detay Üretim Alanları İçin İş Merkezi Kırılımında Üretim Süresi Tahminleme. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi*, *37*(1), 47-60.

Zeybek M. (2021). Classification of Uav Point Clouds By Random Forest Machine Learning Algorithm. *Turkish Journal of Engineering*, *5*(2), 48–57.

EKLER

Ek. A: Kronik Otoimmün Veri Seti İçin Kaynak Kodları

```
install.packages("plyr")
install.packages("dplyr")
install.packages("caret")
install.packages("ggplot2")
install.packages("ggraph")
install.packages("grid")
install.packages("gridExtra")
install.packages("doFuture")
install.packages("caTools")
install.packages("bitops")
install.packages("h2o")
install.packages("RWeka")
install.packages("rJava")
install.packages("mlbench")
install.packages("DiagrammeR")
install.packages("pROC")
install.packages("xgboost")
install.packages("rpart")
install.packages("rpart.plot")
library(rpart)
library(rpart.plot)
library("xgboost")
library("pROC")
library("DiagrammeR")
library("mlbench")
library("plyr")
library("dplyr")
library("ggplot2")
library("grid")
library("gridExtra")
```

```
library("caret")
library("h2o")
library("doFuture")
registerDoFuture()
plan(multiprocess)
library(AppliedPredictiveModeling)
library(RColorBrewer)
library(AppliedPredictiveModeling)
library(RColorBrewer)
library("RWeka")
install.packages("RANN")
library("RANN")
install.packages(c('caret', 'skimr', 'RANN', 'randomForest', 'fastAdaboost', 'gbm', 'xgboost', 'caretEnsemble', 'C50', 'earth'))
install.packages("imbalance")
install.packages("smotefamily")
library(imbalance)
library(smotefamily)
getwd()
setwd("C:/Users/YUNUS EMRE/Desktop")
list.files()
veri1 <- read.csv("MSKU tiroit/veri1.csv")
veri2 <- read.csv("MSKU tiroit/veri2.csv")
UCLveri<-"https://archive.ics.uci.edu/ml/machine-learning-databases/thyroid-disease/new-thyroid.data"
UCLveri<-as.data.frame(read.table(UCLveri,header=FALSE,sep=',',dec="."))
rownames(UCLveri)<-paste0("CAT",1:dim(UCLveri)[1])
colnames(UCLveri)<-c("sınıflar","RT3U","T4","T3","TSH","DTSH")
hedef_UCLveri <- as.factor(x = UCLveri[[1]])
UCLveri$sınıflar <-revalue(hedef_UCLveri,c("1"="tiroit", "2"="hipertiroit", "3"="hipotiroit"))
hedef_veri1 <- as.factor(x = veri1[[1]])
veri1$GRUPLAR <-revalue(hedef_veri1,c("1"="CAT", "2"="kontrol"))
hedef_veri2 <- as.factor(x = veri2[[1]])
veri2$GRUPLAR <-revalue(hedef_veri2,c("1"="CAT", "2"="kontrol"))
```

```
summary(veri1)
summary(veri2)
summary(UCLveri)
veri1<-veri1[complete.cases(veri1),]
veri2<-veri2[complete.cases(veri2),]
smotedata<- veri1[, c(1:16)]
CATveri1<-SMOTE(veri1[,-1],veri1[,1])
CATveri1<-CATveri1$data
CATveri1$class<-as.factor(CATveri1$class)
table(CATveri1$class)
names(CATveri1$class)<-c(CATveri1$class)
trainRowNumbers <- createDataPartition(CATveri1$class, p=0.8, list=FALSE)
eğitimdata <- CATveri1[trainRowNumbers,]
testdata <- CATveri1[-trainRowNumbers,]
fitControl <- trainControl(   method = 'cv',number =10,search = "random" )
model_ctree<-train(class~ ., data=eğitimdata, method='ctree',trControl=fitControl)
model_ctree
test_ctree <- predict(model_ctree, testdata)
confusionMatrix(reference = testdata$class, data = test_ctree, mode='everything', positive='CAT')plot(model_ctree$finalModel, fallen.leaves = FALSE)
model_rf<-train(class~ ., data=eğitimdata, method='rf',trControl=fitControl)
model_rf
test_rf <- predict(model_rf, testdata)
confusionMatrix(reference = testdata$class, data = test_rf, mode='everything', positive='CAT')
model_xgbost<-train(class~.,data=eğitimdata,
method='xgbTree',trControl=fitControl)
model_xgbost
test_xgbost <- predict(model_xgbost, testdata)
confusionMatrix(reference = testdata$class, data = test_xgbost, mode='everything', positive='CAT')
model_C5.0<-train(class~ ., data=eğitimdata, method='C5.0',trControl=fitControl)
model_C5.0
test_C5.0 <- predict(model_C5.0, testdata)
```

```
confusionMatrix(reference = testdata$class, data = test_C5.0, mode='everything', positive='CAT')

model_rpart<-train(class~ ., data=eğitimdata, method='rpart',trControl=fitControl)

model_rpart

test_rpart <- predict(model_rpart, testdata)

confusionMatrix(reference = testdata$class, data = test_rpart, mode='everything', positive='CAT')

plot(model_rpart$finalModel)

text(model_rpart$finalModel)

rpart.plot(model_rpart$finalModel, fallen.leaves = FALSE)

CATveri2<-veri2[, c(1:8)]

CATveri2<-SMOTE(veri2[,-1],veri2[,1])

CATveri2<-CATveri2$data

CATveri2$class=as.factor(CATveri2$class)

table(CATveri2$class)

CATveri2<-mutate(CATveri2,log(CATveri2$sT3))

CATveri2<-mutate(CATveri2,log(CATveri2$ATPO))

CATveri2<-CATveri2[,c(1,2,3,9,5,10,7,8)]

names(CATveri2)<-c("yaş","cinsiyet","TSH","sT3","sT4","ATPO","ATG","class")

summary(CATveri2)

CATveri2<-CATveri2[-348,]

trainRowNumbers2<- createDataPartition(CATveri2$class, p=0.8, list=FALSE)

eğitimdata2<- CATveri2[trainRowNumbers2,]

testdata2<- CATveri2[-trainRowNumbers2,]

model_ctree2<-train(class~ ., data=eğitimdata2, method='ctree',trControl=fitControl)

model_ctree2

test_ctree2 <- predict(model_ctree2, testdata2)

confusionMatrix(reference = testdata2$class, data = test_ctree2, mode='everything', positive='CAT')

plot(model_ctree2$finalModel, fallen.leaves = FALSE)

model_rf2<-train(class~ ., data=eğitimdata2, method='rf',trControl=fitControl)

model_rf2

test_rf2 <- predict(model_rf2, testdata2)

confusionMatrix(reference = testdata2$class, data = test_rf2, mode='everything', positive='CAT')
```

```
model_xgbost2<-train(class~.,data=eğitimdata2,
method='xgbTree',trControl=fitControl)

model_xgbost2

test_xgbost2 <- predict(model_xgbost2, testdata2)

confusionMatrix(reference = testdata2$class, data = test_xgbost2, mode='everything',
positive='CAT')

model_C5.0_2<-train(class~data=eğitimdata2, method='C5.0',trControl=fitControl)

model_C5.0_2

test_C5.0_2 <- predict(model_C5.0_2, testdata2)

confusionMatrix(reference = testdata2$class, data = test_C5.0_2, mode='everything',
positive='CAT')

model_rpart2<-train(class~ ., data=eğitimdata2, method='rpart',trControl=fitControl)

model_rpart2

test_rpart2 <- predict(model_rpart2, testdata2)

confusionMatrix(reference = testdata2$class, data = test_rpart2, mode='everything',
positive='CAT')

plot(model_rpart2$finalModel)

text(model_rpart2$finalModel)

rpart.plot(model_rpart2$finalModel, fallen.leaves = FALSE)

karar.kuralları<-CATveri2[,c(4,6,8)]

karar.kurallarıColor <- brewer.pal(3,'Set1')[1:2]

names(karar.kuralları)<-c("sT3","ATPO","sınıflar")

names(karar.kurallarıColor) <- c('CAT','kontrol')

ggplot(data = karar.kuralları,aes(x =sT3, y = ATPO)) +

geom_point(aes(color = sınıflar), size = 6, alpha = .5) +

scale_colour_manual(name = 'sınıflar', values = karar.kurallarıColor) +

scale_x_continuous(expand = c(0,0)) +

scale_y_continuous(expand = c(0,0))

nbp <- 250;

PredA <- seq(min(karar.kuralları$sT3), max(karar.kuralları$sT3), length = nbp)

PredB <- seq(min(karar.kuralları$ATPO), max(karar.kuralları$ATPO), length = nbp)

Grid <- expand.grid(sT3 = PredA, ATPO = PredB)

PlotGrid <- function(pred,title) {

surf <- (ggplot(data = karar.kuralları, aes(x = sT3, y = ATPO, color = sınıflar))
+geom_tile(data = cbind(Grid, sınıflar = pred), aes(fill = sınıflar))
```

```
+scale_fill_manual(name = 'sınıflar', values = karar.kurallarıColor) +ggtitle("Karar Kuralları") + theme(legend.text = element_text(size = 10)) +
scale_colour_manual(name = 'sınıflar', values = karar.kurallarıColor)) +
scale_x_continuous(expand = c(0,0)) + scale_y_continuous(expand = c(0,0))
pts <- (ggplot(data = karar.kuralları, aes(x = sT3, y = ATPO, color = sınıflar)) +
geom_contour(data = cbind(Grid, sınıflar = pred), aes(z = as.numeric(sınıflar)),
color = "red", breaks = c(1.5)) +geom_point(size = 4, alpha = .5) +
ggtitle("Karar Noktaları") +theme(legend.text = element_text(size = 10)) +
scale_colour_manual(name = 'sınıflar', values = karar.kurallarıColor)) +
scale_x_continuous(expand = c(0,0)) +scale_y_continuous(expand = c(0,0))
grid.arrange(surf, pts, top = textGrob(title, gp = gpar(fontsize = 20)), ncol = 2)}
V <- 10
T <- 4
TrControl <- trainControl(method = "repeatedcv",number = V, repeats = T)
Seed <- 345 ErrsCaret <- function(Model, Name) {
Errs <- data.frame(t(postResample(predict(Model, newdata = karar.kuralları), karar.kuralları[["sınıflar"]])),Resample = "None", model = Name)rbind(Errs, data.frame(Model$resample, model = Name))}
Errs <- data.frame()
CaretLearnAndDisplay <- function (Errs, Name, Formula, Method, ...) {
set.seed(Seed)
Model <- train(as.formula(Formula), data = karar.kuralları, method = Method, trControl = TrControl, ...)
Pred <- predict(Model, newdata = Grid)
PlotGrid(Pred, Name)
Errs <- rbind(Errs, ErrsCaret(Model, Name))}
Errs <- CaretLearnAndDisplay(Errs, "CART", "sınıflar ~ .", "rpart",
control = rpart::rpart.control(minsplit = 5, cp = 0.005), tuneGrid = data.frame(cp = 0))
Tree <- train(sınıflar ~ ., data = karar.kuralları, method = "rpart", control = rpart::rpart.control(minsplit = 5, cp = 0),
tuneGrid = data.frame(cp = 0.005), trControl = TrControl)
Tree$finalModel
rpart.plot(Tree$finalModel)
prp(Tree$finalModel, type = 2, extra = 104, nn = TRUE, fallen.leaves = TRUE,
box.col = karar.kurallarıColor[Tree$finalModel$frame$yval])
```

```
Errs <- CaretLearnAndDisplay(Errs, "Random Forest", "sınıflar ~ .", "rf",
tuneLength = 1,
control = rpart.control(minsplit = 5))
Errs <- CaretLearnAndDisplay(Errs, "C5.0", "sınıflar ~ .", "C5.0")
Errs <- CaretLearnAndDisplay(Errs, "XGBOOST", "sınıflar ~ .", "xgbTree")
Errs <- CaretLearnAndDisplay(Errs, "Karar Ağacı", "sınıflar ~ .", "ctree")
```

Ek. B: UCL Veri Seti İçin Kaynak Kodları

```
trainRowNumbers3 <- createDataPartition(UCLveri$sınıflar, p=0.8, list=FALSE)
eğitimdata3<- UCLveri[trainRowNumbers3,]
testdata3 <- UCLveri[-trainRowNumbers3,]
fitControl <- trainControl(   method = 'cv',number =10,search = "random" )
model_ctree3<-train(sınıflar~ data=eğitimdata3, method='ctree',trControl=fitControl)
model_ctree3
test_ctree3 <- predict(model_ctree3, testdata3)
confusionMatrix(reference = testdata3$sınıflar, data = test_ctree3, mode='everything',
positive='CAT')
plot(model_ctree3$finalModel, fallen.leaves = FALSE)
model_rf3<-train(sınıflar~ ., data=eğitimdata3, method='rf',trControl=fitControl)
model_rf3
test_rf3 <- predict(model_rf3, testdata3)
confusionMatrix(reference = testdata3$sınıflar, data = test_rf3, mode='everything',
positive='CAT')
model_xgbost3<-train(sınıflar~.,data=eğitimdata3,
method='xgbTree',trControl=fitControl)
model_xgbost3
test_xgbost3 <- predict(model_xgbost3, testdata3)
confusionMatrix(reference = testdata3$sınıflar, data = test_xgbost3,
mode='everything', positive='CAT')
model_C5.03<-train(sınıflar~ ., data=eğitimdata3,
method='C5.0',trControl=fitControl)
model_C5.03
test_C5.03 <- predict(model_C5.03, testdata3)
confusionMatrix(reference = testdata3$sınıflar, data = test_C5.03, mode='everything',
positive='CAT')
model_rpart3<-train(sınıflar~ ., data=eğitimdata3,
method='rpart',trControl=fitControl)
model_rpart3
test_rpart3 <- predict(model_rpart3, testdata3)
confusionMatrix(reference = testdata3$sınıflar, data = test_rpart3, mode='everything',
positive='CAT')
plot(model_rpart3$finalModel)
text(model_rpart3$finalModel)
rpart.plot(model_rpart3$finalModel, fallen.leaves = FALSE)
```

Ek B.(devam)

```
karar.kuralları3<-UCLveri[,c(1,3,5)]
karar.kurallarıColor2 <- brewer.pal(3,'Set1')[1:3]
names(karar.kuralları3)<-c("sınıflar","T4","TSH")
names(karar.kurallarıColor2) <- c('tiroit','hipertiroit','hipotiroit')
ggplot(data = karar.kuralları3,aes(x =T4, y = TSH)) +
geom_point(aes(color = sınıflar), size = 6, alpha = .5)
nbp <- 250;
PredA <- seq(min(karar.kuralları3$T4), max(karar.kuralları3$T4), length = nbp)
PredB <- seq(min(karar.kuralları3$TSH), max(karar.kuralları3$TSH), length = nbp)
Grid <- expand.grid(T4 = PredA, TSH = PredB)
PlotGrid <- function(pred,title) {
  surf <- (ggplot(data = karar.kuralları3, aes(x = T4, y = TSH, color = sınıflar)) +
geom_tile(data = cbind(Grid, sınıflar = pred), aes(fill = sınıflar)) +
scale_fill_manual(name = 'sınıflar', values = karar.kurallarıColor2) +
ggtitle("Karar Kuralları") + theme(legend.text = element_text(size = 10)) +
scale_colour_manual(name = 'sınıflar', values = karar.kurallarıColor2)) +
scale_x_continuous(expand = c(0,0)) +
scale_y_continuous(expand = c(0,0))
pts <- (ggplot(data = karar.kuralları3, aes(x = T4, y = TSH,
color = sınıflar)) +geom_contour(data = cbind(Grid, sınıflar = pred), aes(z =
as.numeric(sınıflar)),
color = "red", breaks = c(1.5)) +
geom_point(size = 4, alpha = .5) +
ggtitle("Karar Noktaları") +
theme(legend.text = element_text(size = 10)) +
scale_colour_manual(name = 'sınıflar', values = karar.kurallarıColor2)) +
scale_x_continuous(expand = c(0,0)) +
scale_y_continuous(expand = c(0,0))
grid.arrange(surf, pts, top = textGrob(title, gp = gpar(fontsize = 20)), ncol = 2)}
V <- 10
T <- 4
TrControl <- trainControl(method = "repeatedcv", number = V, repeats = T)
Seed <- 345
ErrsCaret2 <- function(Model, Name) {
```

Ek B.(devam)

```
Errs2 <- data.frame(t(postResample(predict(Model, newdata = karar.kuralları3), karar.kuralları3[["sınıflar"]])),
Resample = "None", model = Name)
rbind(Errs2, data.frame(Model$resample, model = Name))}
Errs2 <- data.frame()
CaretLearnAndDisplay <- function (Errs2, Name, Formula, Method, ...) {
set.seed(Seed)
Model <- train(as.formula(Formula), data = karar.kuralları3, method = Method, trControl = TrControl, ...)
Pred <- predict(Model, newdata = Grid)
PlotGrid(Pred, Name)
Errs2 <- rbind(Errs2, ErrsCaret2(Model, Name))}
Errs2 <- CaretLearnAndDisplay(Errs2, "CART", "sınıflar ~ .", "rpart",
control = rpart::rpart.control(minsplit = 5, cp = 0.005), tuneGrid = data.frame(cp = 0))
Tree <- train(sınıflar ~ ., data = karar.kuralları3, method = "rpart", control = rpart::rpart.control(minsplit = 5, cp = 0),
tuneGrid = data.frame(cp = 0.05), trControl = TrControl)
Tree$finalModel
rpart.plot(Tree$finalModel)
prp(Tree$finalModel, type = 2, extra = 104, nn = TRUE, fallen.leaves = TRUE,
box.col = karar.kurallarıColor2[Tree$finalModel$frame$yval])
Errs2 <- CaretLearnAndDisplay(Errs2, "Random Forest", "sınıflar ~ .", "rf",
tuneLength = 1,
control = rpart.control(minsplit = 5))
Errs2 <- CaretLearnAndDisplay(Errs2, "C5.0", "sınıflar ~ .", "C5.0")
Errs2 <- CaretLearnAndDisplay(Errs2, "XGBOOST", "sınıflar ~ .", "xgbTree")
Errs2 <- CaretLearnAndDisplay(Errs2, "Karar Ağacı", "sınıflar ~ .", "ctree")
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı: Y***s E**e C****N

Uyruk: Türkiye Cumhuriyeti

Doğum Yeri ve Tarihi: İstanbul / 0*.0*.1**4

E-posta: 29******9@gmail.com

Eğitim

Derece	Kurum	Bölüm	Giriş-Mezuniyet
Lisans	Muğla Sıtkı Koçman Üniversitesi	İstatistik	2013-2018
Lise	Bayrampaşa Tuna Lisesi	Fen Bilimleri	2008-2012

İş Tecrübesi

Yıl	Yer	Kurum	Pozisyon/Görev
2018-	İstanbul	Milli Eğitim Bakanlığı	Matematik Öğretmeni

Sertifika

Uluslararası Lisansüstü Çalışmalar Kongresi Katılım Sertifikası, (Haziran 2021).

4006-Tübitak Bilim Fuarları Destekleme Programı Katılım Sertifikası, (Haziran 2021).