

T.C.
HACETTEPE ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ

TIPTA KARAR VERMEDE BULANIK KARAR AĞAÇLARI,
LOJİSTİK REGRESYON VE CART (CLASSIFICATION AND
REGRESION TREES) ANALİZ YÖNTEMLERİNİN
KARŞILAŞTIRILMASI

Uzm. Dr. Didem TURGUT

Biyoistatistik Programı
YÜKSEK LİSANS TEZİ

ANKARA

2025

**T.C.
HACETTEPE ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ**

**TIPTA KARAR VERMEDE BULANIK KARAR AĞAÇLARI,
LOJİSTİK REGRESYON VE CART (CLASSIFICATION AND
REGRESION TREES) ANALİZ YÖNTEMLERİNİN
KARŞILAŞTIRILMASI**

Uzm. Dr. Didem TURGUT

**Biyoistatistik Programı
YÜKSEK LİSANS TEZİ**

**TEZ DANIŞMANI
Prof. Dr. Pınar ÖZDEMİR**

**ANKARA
2025**

TIPTA KARAR VERMEDE BULANIK KARAR AĞAÇLARI, LOJİSTİK REGRESYON VE
CART(CLASSİFİCATION AND REGRESSION TREES) ANALİZ YÖNTEMLERİNİN
KARŞILAŞTIRILMASI

Didem TURGUT

Danışman: Prof. Dr. Pınar ÖZDEMİR

Bu tez çalışması 11/4/2025 tarihinde jürimiz tarafından “Biyoistatistik Programı” nda yüksek lisans tezi olarak kabul edilmiştir.

Jüri Başkanı: Dr. Öğr. Üyesi Hande ŞENOL

Pamukkale Üniversitesi

Tez Danışmanı: Prof. Dr. Pınar ÖZDEMİR

Hacettepe Üniversitesi

Üye: Dr. Öğr. Üyesi Ebru ÖZTÜRK

Hacettepe Üniversitesi

Bu tez Hacettepe Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca yukarıdaki jüri tarafından uygun bulunmuştur.

09 Mayıs 2025

Prof. Dr. Müge YEMİŞCİ ÖZKAN

Enstitü Müdürü

ÖZET

TURGUT, D., Tıpta Karar Vermede Bulanık Karar Ağaçları, Lojistik Regresyon ve CART (Classification And Regresion Trees) Analiz Yöntemlerinin Karşılaştırılması, Hacettepe Üniversitesi Sağlık Bilimleri Enstitüsü Biyoistatistik Programı Yüksek Lisans Tezi, Ankara, 2025. Tıpta karar vermede kullanılan veri setleri karmaşık olabilmektedir. Bu veri yapılarında değişkenlerin sayıca çok ve belirsizliğin etkisi altında olmaları da sonuç kararlarını etkiler. Bu nedenle farklı veri yapılarında değişkenler arasındaki ilişkiyi belirlemek için gelişmiş analiz yöntemlerine ihtiyaç duyulmaktadır. Sınıflama ve regresyon yöntemleri, bağımlı ve bağımsız değişkenler arasındaki ilişkiyi matematiksel olarak tanımlamaya yöneliktir. Bağımsız değişkenler için herhangi bir varsayım olmaksızın kategorik bağımlı değişkeni tahmin etmek için kullanılan bu modeller güvenilirliklerinin iyi olması ve veri setleri ile kolayca entegre edilebilmeleri gibi nedenlerle sınıflama modelleri içerisinde yaygın bir şekilde kullanılmaya başlanmıştır. Lojistik regresyon da bağımlı değişkenin kategorik olması durumunda kullanılan istatistiksel yöntemdir. Veri setindeki belirsizlikler ve kesin olmayan verilerle başa çıkmak için kullanılan diğer bir yöntem de bulanık mantıktır. Malnutrisyonu olan 155 hemodiyaliz hastasının değerlendirildiği bu tezde bağımsız değişken olarak serum albümin, BMI, nPCR ve serum GDF15 kullanılmıştır. Tip-1 bulanık karar ağaçları bu veri yapısı için uygun olabilir. Sınıflama ve Karar Ağacı (CART) algoritması, Tip-1 bulanık karar ağacı algoritması ve lojistik regresyon analizinin değerlendirildiği bu tezde modellerin sınıflandırma performans ölçütü olarak doğruluk (accuracy) ve eğri altında kalan alan yöntemleri kullanılmıştır. Bu veri seti için CART ve lojistik regresyonun doğruluk yüzdeleri benzer (0,652 ve 0,760); ancak Tip-1 bulanık karar ağacı modelinin diğerlerine göre daha düşük doğruluk yüzdesi olduğu (0,456) görüldü. Yine ROC eğrisi ve eğri altında kalan değerlendirildiğinde bulanık karar ağacı modelinin performansı düşük olarak değerlendirildi. Her iki karar ağaçlarının oluşumunda ilk dallanma kriteri olarak en önemli bağımsız değişken serum albümin değerinin alındığı görüldü. Bu sonuç lojistik regresyon ile belirlenen bağımsız değişkenlerin katsayıları yorumları ile benzer olarak değerlendirildi.

Anahtar Kelimeler: Bulanık mantık, CART, karar ağacı, lojistik regresyon

ABSTRACT

TURGUT, D., Comparison of Fuzzy Decision Trees, Logistic Regression and CART (Classification and Regression Decision Trees) Analysis Methods in Medical Decision Making, Hacettepe University Graduate School of Health Sciences Biostatistics Program Master's Thesis, Ankara, 2025. Data sets used in medical decision making can be complex and limited. The large number of variables in these data structures and the uncertainty may change the final decisions. Therefore, advanced analysis methods are needed to determine the relationship between variables. Classification and regression methods are an important part of statistical analysis to mathematically describe the relationship between dependent and independent variables. These models have been widely used in classification models due to their good reliability and easy integration with data sets. Logistic regression is another statistical method used when the dependent variable is categorical. Fuzzy logic is used to deal with imprecise data when there are uncertainties. In this thesis where 155 hemodialysis patients with malnutrition were evaluated, serum albumin, BMI, nPCR and serum GDF15 were used as independent variables. Classification and Decision Tree (CART) algorithm, Type-1 fuzzy decision tree algorithm and logistic regression analysis were evaluated and accuracy and area under the curve methods were used as classification performance measures of the models. For this dataset, the accuracy percentages of CART and logistic regression were similar (0.652 and 0.760); however, the Type-1 fuzzy decision tree model had a lower accuracy percentage (0.456) than the others. Again, when the ROC curve and the residual under the curve were evaluated, the performance of the fuzzy decision tree model was evaluated as low. It was seen that the most important independent variable serum albumin value was taken as the first branching criterion in the formation of both decision trees. This result was similar to the interpretation of the coefficients of the independent variables determined by logistic regression. As a result, CART and logistic regression classification performances were similar, while fuzzy decision tree classification performances were low, which may be related to the linear structure of the data set and low uncertainty.

Key Words: Fuzzy, CART, decision tree, logistic regression

YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kâğıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe Üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alınarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayınlanan “**Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge**” kapsamında tezim aşağıda belirtilen koşullar haricince YÖK Ulusal Tez Merkezi / H.Ü. Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

- o Enstitü / Fakülte yönetim kurulu kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren 2 yıl ertelenmiştir.¹
- o Enstitü / Fakülte yönetim kurulunun gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren ... ay ertelenmiştir.²
- o Tezimle ilgili gizlilik kararı verilmiştir.³

...../...../2025

Didem TURGUT

1 “Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge”

(1) Madde 6. 1. Lisansüstü teze ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir.

(2) Madde 6. 2. Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış ve internetten paylaşılması durumunda 3. şahıslara veya kurumlara haksız kazanç imkânı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulunun gerekçeli kararı ile altı ayı aşmamak üzere tezin erişime açılması engellenebilir.

(3) Madde 7. 1. Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir. Kurum ve kuruluşlarla yapılan iş birliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.

Madde 7.2. Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir

* Tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu tarafından karar verilir.

ETİK BEYAN

Bu çalışmadaki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi, görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu, kullandığım verilerde herhangi bir tahrifat yapmadığımı, yararlandığım kaynaklara bilimsel normlara uygun olarak atıfta bulunduğumu, tezimin kaynak gösterilen durumlar dışında özgün olduğunu, Prof. Dr. Pınar ÖZDEMİR danışmanlığında tarafımdan üretildiğini ve Hacettepe Üniversitesi Sağlık Bilimleri Enstitüsü Tez Yazım Yönergesine göre yazıldığını beyan ederim.

Uzm. Dr. Didem TURGUT

TEŞEKKÜR

Tezimin planlanmasından tamamlanmasına kadar geçen süreçte ve biyoistatistik eğitimimin her aşamasında verdikleri sonsuz destek ve yardımları için tez danışmanım Prof. Dr. Pınar ÖZDEMİR ve Biyoistatistik Anabilim Dalı öğretim üyelerine ve asistanlarına teşekkürü bir borç bilirim. Sayın emekli bölüm Başkanımız Prof. Ergun KARAAĞAOĞLU'na böyle bir eğitim alabilmemde sağlamış olduğu fırsat için yürekten minnettarım.

Tez izleme komitemde bilgi ve deneyimleriyle bana yol gösteren Sayın Dr. Öğretim Üyesi Hande Şenol'a ve Sayın Dr. Öğretim Üyesi Ebru Öztürk'e ayrıca teşekkür ederim. Tez ile ilgili analizler kısmında verdiği sınırsız destekler için Sevgili Gökalep YENTÜR'e de teşekkürü bir borç bilirim.

Sevgili Menekşe TARLA'ya verdiği motivasyon destekleri, işleri kolaylaştırmada, yoluna koymadaki inanılmaz çabaları ve etkinliği için canıgönülden teşekkür ederim.

Tıp eğitimimde olduğu gibi Yüksek lisans eğitimimde de benden desteğini esirgemeyen sevgili aileme sonsuz minnet ve şükranlarımı sunarım.

Sayın Prof. Dr. Pınar ÖZDEMİR, siz olmasaydınız bunu başaramazdım. Hayatımdaki yeriniz tez danışmanlığının çok ötesinde. Yüksek eğitimci kimliğiniz ve güzel kalbinizle bana kattıklarınızı ömrüm oldukça hatırlayacağım.

İÇİNDEKİLER

ONAY SAYFASI	iii
ÖZET	iv
ABSTRACT	v
YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI	vi
ETİK BEYAN	vii
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ŞEKİLLER	xi
TABLolar	xii
1. GİRİŞ	1
2. GENEL BİLGİLER	3
2.1. Lojistik Regresyon	3
2.1.1. Lojistik Regresyon Modelinin Matematiksel Gösterimi	4
2.1.2. Lojistik Regresyon Modelinin Uygulanması	5
2.1.3. Model Seçim Yöntemleri ve Model Uyum İyiiliğinin Değerlendirilmesi	7
2.2. Sınıflandırma ve Regresyon Karar Ağaçları (Classification And Regression Trees (CART))	9
2.2.1. Karar Ağaçlarının Yapısı	10
2.2.2. Sınıflandırma ve Regresyon Karar Ağaçları (classification and regression trees (CART)) Algoritması	13
2.3. Bulanık Karar Ağaçları (Fuzzy Decision Trees)	15
2.3.1. Bulanık Mantık ve Bulanık Kümeleme	15
2.3.2. Bulanık Sistem Kuralları	17
2.3.3. Bulanık Karar Ağaçları	18
2.4. Sınıflandırma ve Regresyon Karar Ağaçları, Bulanık Karar Ağaçları ve Lojistik Regresyon Yöntemlerinin Karşılaştırılması	19

3. UYGULAMA	22
3.1. Tanımlayıcı İstatistikler	22
3.2. Sınıflandırma ve Regresyon Karar Ağacı Uygulaması	24
3.3. Bulanık Karar Ağacı Uygulaması	25
3.4. Lojistik Regresyon Analizi Sonuçları	28
3.5. CART, Bulanık Karar Ağacı ve Lojistik Regresyon Analiz Sonuçlarının Karşılaştırılması	28
4. TARTIŞMA	30
5. SONUÇ VE ÖNERİLER	33
7. KAYNAKLAR	34
8. EKLER	38
EK-1: Turnitin Raporu Ekran Görüntüsü	38
EK-2: Turnitin Dijital Makbuz	39
EK-3: Tezin Hazırlanması Aşamasında Kullanılan RStudio Bilgisayar Yazılımı Kodları	40
9. ÖZGEÇMİŞ	

ŞEKİLLER

Şekil	Sayfa
2.1. Lojistik regresyon modelinin grafiksel gösterimi	5
2.2. Örnek karar ağacı	11
2.3. Bulanık sistem akış şeması	17
3.1. Bağımsız değişkenlerin korelasyon dağılımı	23
3.2. CART algoritması ile malnutrisyonun değerlendirilmesi (Orange datamining)	24
3.3. CART algoritması ile malnutrisyonun değerlendirilmesi (Rstudio)	25
3.4. Bulanıklaştırma katsayısı $m=2$ olduğunda bağımsız değişkenler için kümeler	26
3.5. Bağımsız değişken serum albümin için bulanıklaştırma katsayısı $m=2$ için kümeler	26
3.6. Bağımsız değişken serum albümin için bulanıklaştırma katsayısı $m=3$ için kümeler	27
3.7. Tip-1 bulanık karar ağacı algoritması ile malnutrisyonun değerlendirilmesi	27
3.8. Lojistik regresyon, CART ve tip-1 bulanık karar ağacı ROC eğrisi grafikleri	29

TABLolar

Tablo	Şekil
2.1. Bulanık küme işlemleri	16
2.2. Sınıflandırma ve regresyon karar ağaçları, bulanık karar ağaçları ve lojistik regresyon analizinin karşılaştırılması	21
3.1 Bağımsız değişkenlerin tüm veri setinde ve gruplar arasında dağılımları	23
3.2. Lojistik regresyon sınıflama tablosu	28
3.3. Lojistik regresyon bağımsız değişkenlerin katsayı tablosu	28
3.4. Modellerin tüm veri seti için doğruluk oranlarının karşılaştırılması	28

1. GİRİŞ

Tüm bilimsel ve akademik çalışmalarda veri analizinde tanımlayıcı istatistikler, regresyon analizleri, kümeleme yöntemleri gibi teknikler sıklıkla kullanılmaktadır. Özellikle tıp alanında yapılan bilimsel çalışmalarda, veri setleri oldukça düzensiz ve karışık olabilmektedir. Tıbbi kararlar da çoğu zaman karmaşık, belirsiz ve sınırlı veriye dayalı olan bu çalışmalarla belirlenmektedir. Bu veri yapılarında değişkenlerin sayıca çok olması ve belirsizliğin etkisi altında olmaları nedenleriyle kararların etkinliğini artırabilmek ve değişkenler arasındaki ilişkiyi belirlemek için gelişmiş analiz yöntemlerine ihtiyaç duyulmaktadır.

İstatistiksel çalışmalarda değişkenler arasındaki ilişkiyi matematiksel olarak modellemek için farklı yöntemler kullanılmaktadır. Sınıflama ve regresyon yöntemleri, bağımlı ve bağımsız değişkenler arasındaki ilişkiyi matematiksel olarak belirtmeye yönelik istatistiksel analizlerin önemli bir parçasıdır. Modelleme örneklerinin en yaygın olanı bağımlı değişkenin sürekli olduğu doğrusal regresyon modelidir. Bağımlı değişkenin kategorik olması halinde tipik doğrusal modelin uygulanamadığı ve normallik varsayımının bozulması durumlarında lojistik regresyon modeli kullanılmaktadır. Lojistik regresyon analizinde amaç en az değişkenin kullanılmasıyla bağımlı ve bağımsız değişkenler arasındaki ilişkiyi tanımlayabilmek, en iyi uyuma sahip olacak şekilde ve istatistiksel olarak uygun bir model kurmaktır (1). Varsayımlardan bağımsız kategorik bağımlı değişkeni tahmin etmek için karar ağaçları da kullanılabilir. Sınıflandırma ve Regresyon Karar Ağaçları (Classification and Regression Trees (CART)), güvenilirliklerinin iyi olması ve veri setleri ile kolayca entegre edilebilmeleri, yorumlanmalarının kolay olması gibi nedenlerle sınıflama modelleri içerisinde yaygın kullanıma sahip olan yöntemlerdir (2). Bu yöntemler tanımlayıcı ve tahmin edici özelliklere sahiptir. Sınıflandırma ve Regresyon Karar Ağaçları her ne kadar oluşturulma ve yorumlama sürecinde kolaylık sağlasa da ve çoğu zaman başarılı performans gösterse de karar verme sürecinde, kesin olmayan bilgiler olması durumunda ve belirsizlikte yetersiz kalmaktadır. Bu kapsamda, temelini bulanık mantık/bulanık kümelemenin oluşturduğu bulanık karar ağaçlarının kullanımı daha uygun olmaktadır (3).

Bu tezde farklı istatistiksel modellerin aynı veri seti ile performanslarının değerlendirilmesi amaçlanmıştır. Tezin genel bilgiler kısmında uygulanacak istatistiksel yöntemler olan sınıflama ve karar ağaçları yaklaşımı, bulanık karar ağacı yaklaşımı ve lojistik regresyon analizine genel bir giriş yapılmıştır. Genel tanım ve kavramlara yer verilmiştir ve çeşitli karar ağacı algoritmaları çıkış yıllarına göre incelenmiştir. Her üç yöntem için sınıflama ve tahmin modellerinin oluşum yapısı incelenmiştir. Oluşturulmuş olan modeller içerisinde en uygun modellerin seçiminin yapılabilmesi için gerekli olan seçim kriterlerinden bahsedilmiştir. Ve her üç yöntemin avantaj ve dezavantajlarına bakılarak yöntemler karşılaştırılmıştır.

Tezin uygulama bölümünde; sınıflama ve karar ağacı, bulanık tip karar ağacı ve lojistik regreasyon analizleri aynı gerçek hasta veri setinde uygulamalı olarak ele alınmıştır. Her üç algoritmanın avantajları ve dezavantajları değerlendirilmiştir. Analiz sonucunda kurulan modeller karşılaştırılmış yapılmış ve her üç yöntemden elde edilen bulgular yorumlanmıştır.

Tezin tartışma kısmında literatürde istatistiksel yöntemlerin karşılaştırıldığı tıbbi alanda yapılmış çalışmalr değerlendirilerek uygulama bölümünde elde edilen sonuçlar yorumlanmıştır. Tezin katkıları ve kısıtlılıkları değerlendirilerek sonuç ve önerilerde bulunulmuştur.

2. GENEL BİLGİLER

Tıpta karar vermede kullanılacak istatistiksel yöntemler, araştırılan konunun gereksinimlerine ve veri setinin yapısına bağlı olarak değişir. CART hem sınıflandırma hem de regresyon problemleri için esnek bir yöntem sunar. Bulanık karar ağaçları, belirsiz ve karmaşık karar süreçlerinde faydalı olabilirken, lojistik regresyon daha net ilişkileri göstermede etkili bir seçimdir. Her bir yöntemin avantajlarını ve dezavantajlarını değerlendirmek, doğru karar verme ve analiz süreci için önemlidir.

2.1. Lojistik Regresyon

Regresyon analizi, iki veya daha fazla değişken arasındaki bağımlılık yapısını incelemek ve aralarındaki ilişkiyi kullanarak veriyi özetlemek, bağımlı değişkeni etkileyen önemli değişkenleri belirlemek veya o konu ile ilgili kestirimler yapabilmek veya tahminlerde bulunabilmek gibi amaçları hedefler. Bu analizde bağımlı değişken ve bağımsız değişken (ler) arasında ilişki matematiksel modellerle açıklanır ve iki (basit regresyon) veya daha fazla değişken (çoklu regresyon) modellerde kullanılabilir. En sık kullanılan doğrusal regresyon yöntemlerinde bağımlı değişken sayısal veri türündedir. Ancak, özellikle tıpta kullanılan veri türlerinde bağımlı değişkenin kategorik/niteliksel olduğu durumlar da sıktır. Bu durumda kestirim yapmak için lojistik regresyon yöntemi ile çözümleme yapılmaktadır (4). Bağımlı değişkenin iki kategorili/durumlu niteliksel değişken olması durumunda iki kategorili (binary) lojistik regresyon yöntemi kullanılmaktadır. Bu tez kapsamında bağımlı değişkenin iki kategorili nitel değişken olması durumu temelinde çalışılacaktır.

Lojistik model ilk defa 1800 yıllarının başında tanımlanmış ve toplumların nüfus olarak büyümelerini değerlendirmede kullanılmaya başlanmıştır (1). Probit modelin geliştirilmesi ve biyolojide çalışılmaya başlanmasını logit modelin biyoloji alanında probit modele alternatif olarak kullanılmasına yol açmıştır (5). Yıllar içerisinde Cox'un çalışmaları ile model ile ilgili birçok iyileştirme yapılmış ve lojistik regresyon modellerinin yaygın bir şekilde kullanılır hale gelmesi, katsayı tahmin yöntemlerinin geliştirilmesine neden olmuştur (6). İlk defa 1962 yılında

Cornfiell, lojistik regresyondaki katsayı tahmin işlemlerinde diskriminant fonksiyonu yaklaşımını kullanmıştır (7). Takip eden yıllarda, Robert ve ark. en çok olabilirlik ve hipotez testleri üzerine, Hsu ve Leonard ise Monte Carlo dönüşümlerinin uygulanabilirliği üzerine çalışmalar yapmıştır (8). Ancak lojistik regresyon analizine en kapsamlı katkıyı Hosmer ve Lemeshow sunmuştur (9).

2.1.1. Lojistik Regresyon Modelinin Matematiksel Gösterimi

Lojistik regresyon modellerinde bağımsız değişkenlerin sürekli veya kesikli değer alması koşulu aranmaz, ancak aralarında yüksek düzeyde bağlantı bulunması ve yeterli örneklem büyüklüğünün olmaması modelin test edilebilirliğini etkilemektedir. İki kategorili Lojistik regresyonda bağımlı/yanıt değişkenin değerinin kestirilmesi ile ilgilenilmez. Bunun yerine, bağımlı değişkenin 1 değerini alması olasılığı (riskli durum 1 olarak belirlendiğinde) kestirilmeye çalışılır. Elde edilen sonuç bir olasılık değeri olduğu için sadece 0 ile 1 arasında değer alabilir (4). Bu nedenle, lojistik regresyon, hastaların bir hastalığa sahip olup olmadıklarını, tedaviye yanıt verip vermediklerini tahmin etmeye yarar. Bağımlı değişken, iki sınıflı olarak kullanılan 1 (evet/doğru) veya 0 (hayır/yanlış) olarak kodlanmış verileri içerir. Olasılığın modellenmesi için ise lojistik regresyon modelinden yararlanılır (Eşitlik-1).

$$\pi(x) = P(Y = 1|X = x) = \frac{e^{\beta_0 + \beta_1(x)}}{1 + e^{\beta_0 + \beta_1(x)}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1(x))}} \text{-----Eşitlik-1}$$

Lojistik regresyon modelinde odds oranı (odds ratio-OR) ise iki farklı olasılık değerinin oranını göstermektedir. Bir olayın olma olasılığının (π) olmama olasılığına ($1-\pi$) oranı olarak tanımlanır. $OR > 1$, $OR < 1$ ve $OR = 1$ değerlerini alabilir. Bağımlı değişkenin bağımsız değişkenin etkisi ile kaç kat daha fazla gözlemlendiğini belirten OR'nin 1 olması, bağımsız değişkenin odds üzerinde bir etkisi olmadığı anlamına gelmektedir (10).

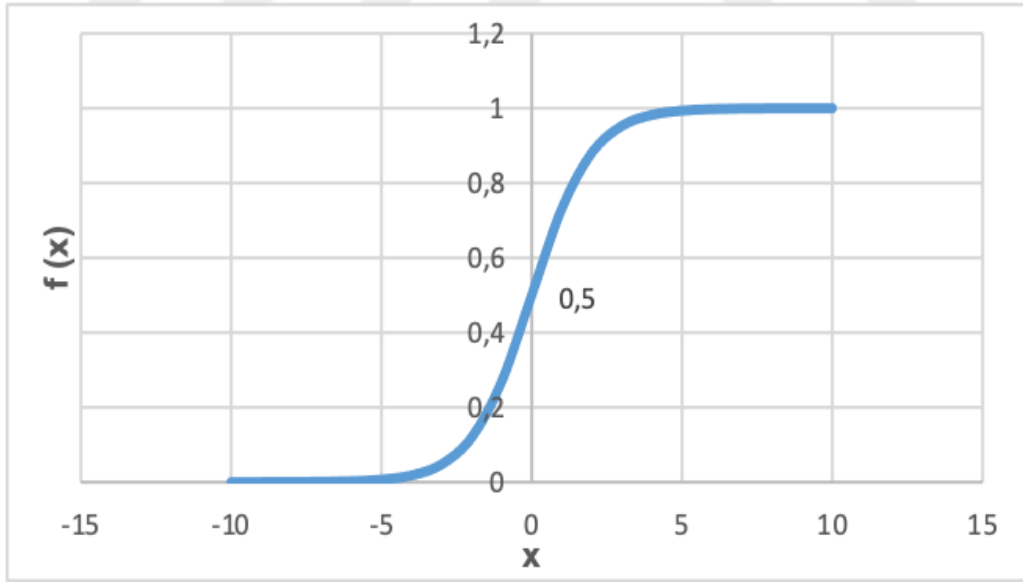
Doğrusal olmayan lojistik regresyon fonksiyonu logit dönüşümü uygulandığında doğrusallaştırılabilmektedir. Logit ile bağımsız değişkenler arasındaki ilişkinin doğrusal olması gerekliliği modelin tek varsayımını oluşturur (Eşitlik-2). Lojistik regresyon modelini doğrusal modele dönüştürmek için yanıt değişkeninin odds türünden belirtildiği modelin doğal logaritması alınır (Eşitlik-3).

$$\text{logit}_{\pi(x)} = g(x) = \ln\left(\frac{\pi(x)}{1-\pi(x)}\right) \text{-----Eşitlik-2}$$

$$g(x) = \ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = \ln e^{\beta_0 + \beta_1(x)} = \beta_0 + \beta_1(x) \text{-----Eşitlik-3}$$

Eşitlikten de anlaşıldığı gibi lojit ile bağımsız değişkenler arasındaki ilişkinin doğrusal olması modelin varsayımdır. Doğrusal modelde β_1 bağımsız değişken x 'teki 1 birim değişimin bağımlı değişkende ne kadarlık değişime neden olduğunu gösterirken, lojistik modelde x 'teki 1 birim değişimin lojit'te ne kadar değişime neden olduğunu gösterir. Lojistik regresyonda artıklar sıfır ortalama ve $\pi(1 - \pi)$ varyansla binom dağılım gösterir (4).

Lojistik regresyon modeli yanıt/cevap fonksiyonu ile gösterilir ve grafiksel gösterimde eğri S şeklindedir (Şekil. 2.3). Bu eğride olasılıkları artan ve azalan x değerlerine karşılık fonksiyon değerleri (π) 0 ile 1 arasında kalmaktadır. $x = -\infty$ olduğunda $f(x) \approx 0$ değerini alırken, $x = \infty$ olduğunda $f(x) \approx 1$ değerini almaktadır (11).



Şekil 2.1. Lojistik regresyon modelinin grafiksel gösterimi

2.1.2. Lojistik Regresyon Modelinin Uygulanması

Lojistik regresyon modelinin avantajları ve dezavantajları vardır. Lojistik regresyon analizi yalnızca bir tahmin edicinin katsayı büyüklüğünü vermekle kalmaz,

aynı zamanda ilişkinin pozitif veya negatif yönünü de gösterir. Büyük veri setlerini de yüksek hızda işleme özelliği vardır. Ayrıca Doğrusal regresyon analizinde bağımsız değişkenin çoklu normal dağılım göstermesi şartı aranırken, Lojistik Regresyon Analizinde böyle bir şart yoktur. En büyük dezavantajları ise modelin tam ayırma sorunu yaşayabileceği ve yüksek boyutlu veri kümelerinde aşırı uyum sorunlarına yol açabilmesidir.

Regresyon Katsayılarının Kestirimi

Lojistik regresyon modelinde regresyon katsayılarının kestirimi genellikle iteratif bir yöntem olan *En Çok Olabilirlik* (maximum likelihood method) kullanılarak hesaplanır. Bu fonksiyonu en büyük yapan değerler, bilinmeyen parametrelerin en çok olabilirlik yani en iyi uyuşan kestiricileridir (4, 12). Başka fonksiyonlar da katsayı kestiriminde kullanılabilirken, en çok olabilirlik kestiricisi hem örneklem büyüklüğü sonsuza yaklaştıkça hem de küçük olduğunda tutarlı olabilmektedir.

Regresyon Katsayılarının Önemliliğinin Test Edilmesi

Regresyon katsayıları kestirimi yapıldıktan sonra modeldeki bağımlı değişken ile bağımsız değişkenler arasındaki ilişkinin önemliliği test edilir. Bunun için kullanan üç test vardır. Bunlar *Olabilirlik Oran (Likelihood Ratio-LR)*, *Wald* ve *Skor* testleridir. Analizlerin temeli bağımsız değişken modelde olduğunda ve modelde olmadığında elde edilen kestirim değerlerinin gözlenen değerlerle karşılaştırılmasına dayanır.

Olabilirlik oran testinde modellerin olabilirlik fonksiyonlarının maksimum değerleri karşılaştırılır (Eşitlik-4) G istatistiği olarak adlandırılan bu fonksiyon, doğrusal regresyondaki kısmi F istatistiği ile aynı rolü oynamaktadır (12).

$$G = -2 \ln \frac{\text{Sadece sabit içeren fonksiyonun maksimum değeri}}{\text{Tüm modelin olabilirlik fonksiyonunun maksimum değeri}} \text{-----Eşitlik-4}$$

Test istatistiği LR 'de sıfır hipotezi altında 1 serbestlik dereceli ki-kare dağılımına sahiptir. LR testi, sapma (deviance) testi olarak da bilinmektedir. Sapma,

doğrusal regresyondaki hata kareler toplamına karşılık gelmektedir. G test istatistiği, bağımsız değişkenlerin modele olan katkılarının araştırılmasının yanında, modelin genel geçerliliğini de test eder (13).

Wald test istatistiği yokluk hipotezi ve örneklem büyüklüğü varsayımları sağlandığında normal dağılım gösterir. Wald testi de olabilirlik fonksiyonunu kullanır. Eğim parametresinin en çok olabilirlik kestiriminin standart hatasına bölünmesi ile elde edilir (Eşitlik-5) (9).

$$w = \frac{\hat{\beta}_j}{s(\hat{\beta}_j)} \text{-----Eşitlik-5}$$

Wald istatistiği için alternatif olarak ki-kare dağılımı da kullanılabilir. Bu durumda en çok olabilirlik kestiriminin karesi $(\hat{\beta}_j)^2$ standart hatasının karesine $(s(\hat{\beta}_j))^2$ bölünür.

Skor testinin diğer testlere avantajı en çok olabilirlik kestiriminin hesaplanmasına gerek olmamasıdır. Genel olarak matris hesaplamaları gerektiren çok değişkenli bir testtir. Hesaplama işlemlerini çok kısaltmakla beraber istatistik yazılımlarda bulunmaması kullanımını kısıtlamaktadır. Skor test istatistiği normal dağılıma uymaktadır (Eşitlik-6) (9).

$$ST = \frac{\sum_{i=1}^n x_i (y_i - \bar{y})}{\sqrt{y_i (1 - \bar{y}) \sum_{i=1}^n (x_i - \bar{x})^2}} \text{-----Eşitlik-6}$$

2.1.3. Model Seçim Yöntemleri ve Model Uyum İyiliğinin Değerlendirilmesi

Lojistik regresyonda model seçim yöntemleri belirlenirken çalışmanın amacı ve bağımlı değişkeni etkileyebilecek uygun bağımsız değişkenlerin belirlenmesi esastır. Uygulamada birden fazla uygun model olabilir, ancak tüm olası regresyon modeller değerlendirilmeye alınmalıdır. Modelin kararlılığını artırmak ve kullanılabilir olmasını sağlamak için bağımsız değişkenlerin özellikleri iyi bilinmeli ve mümkün olduğunca az sayıda değişken modele dahil edilmelidir.

Değişken seçimi için Hosmer ve arkadaşları, amaca yönelik yaklaşımı önerir. Tek değişkenli analiz ile anlamlı değişkenler belirlenir ve bu değişkenler Wald testi ile çok değişkenli analizde değerlendirilir. Burada p-değeri 0,20 ya da 0,25 olarak alınmaktadır (12). Önemli değişkenlerin belirlenmesi için sıklıkla başvurulacak yöntemlerden diğeri de adımsal seçim (stepwise) yöntemleridir. Adımsal seçim yöntemleri, değişkenleri ekleyerek veya çıkararak en iyi modeli bulmayı hedefler. İleriye doğru seçimde (forward selection) değişkenler eklenirken, geriye doğru elemeye (backward elimination) karmaşık modelden anlamsız değişkenler çıkarılır. Ancak bu yöntemler model belirsizliklerini göz ardı edebilir ve yanlış sonuçlara yol açabilir. Adım adım seçim yönteminde ise ileriye ve geriye doğru seçim yöntemleri birleştirilir. Her adımda regresyon modeline bir değişken eklenir ya da çıkarılır.

Alternatif olarak, Rastlantısal Orman (RF) yöntemi, değişken seçiminde doğrudan sonuçlar sunar ve değişkenlerin önemini belirlemek için Gini ve permütasyon ölçütleri kullanır (14). Model belirsizliğini azaltmak için Bayesyen Model Ortalaması (BMA) yöntemi önerilmektedir; bu yöntem, tüm modellerin olasılıklarını dikkate alarak daha güvenilir sonuçlar sağlayabilir (15).

Lojistik regresyon analizi uygulamalarında ikiden fazla bağımsız değişken olduğunda değişkenler arasındaki ilişkinin de değerlendirilmesi gerekir. Mesela hem bağımlı hem bağımsız değişkenle ilişkili olan bir değişken etki karıştırıcı olarak tanımlanır. Etki karışımını gidermek için olası değişken modele eklenerek analizlerin değerlendirilmesi önerilir (4). Yine modelde bir bağımsız değişkenin bağımlı değişken ile ilişkisi, diğer bir bağımsız değişkenin düzeyine göre değişiyorsa etkileşim olduğu söylenir. Etkileşimi incelemek için öncelikle bağımsız değişkenlerin ana etkilerini içeren model oluşturulur ve bu modele etkileşim terimi eklenerek önemlilikleri incelenir (4). Bağımsız değişkenlerin birbiriyle ilişkili olması ise çoklu bağlantı sorunu oluşturabilir. Bu durumda regresyon katsayılarının standart hataları olduğundan çok büyük kestirilir. Çoklu bağlantı sorunundan şüphelenildiğinde değişkenlerden sadece bir tanesinin modele alınması veya farklı tek değişken oluşturulması sorunu çözebilmektedir (4).

Modelin uyum iyiliği, lojistik regresyon analizinde elde edilen modelin sonuç değişkenini tanımlamakta ne kadar etkili olduğunu incelemekle belirlenir. Bunun için

pearson ki-kare ve sapma istatistikleri, Hosmer-Lemeshow testi, sınıflandırma tabloları ve işlem karakteristiği eğrisi (ROC eğrisi) altında kalan alan gibi farklı yöntemler kullanılabilir (12).

Lojistik regresyon analizinde doğrusal regresyon analizindeki R^2 istatistiğine benzeyen ve geniş kabul gören bir istatistik bulunmamaktadır. Ancak model uyum iyiliğini değerlendirmek için pseudo R^2 istatistikleri geliştirilmiştir. McFaden R^2 , Cox-Snell ve Nagelkerke R^2 en sık kullanılanlardır. Ancak bu değerler model uygunluğunun değerlendirilmesinden çok farklı modellerin performansını değerlendirmede kullanılmaktadır.

Model yeterliliğinin değerlendirilmesinde genelde gerçek olasılıklar ile tahmin edilen olasılıklar arasındaki standart farklara bakılmaktadır. Bu amaçla katsayıların yorumlanması, artıkların incelenmesi (etkili, uzak ve aykırı gözlemlerin incelenmesi), grafiksel değerlendirmelerin yapılması (artık grafiklerinin incelenmesi, ROC eğrisi altında kalan alan) gibi işlemlerin yapılması önerilmektedir (4).

2.2. Sınıflandırma ve Regresyon Karar Ağaçları (Classification And Regression Trees (CART))

Günümüzde hemen hemen her alanda bilgi (veri) hacminin giderek artması bu bilgilerin anlamlı şekilde işletilmesini ve günlük hayatta kullanılmasını gündeme getirmiştir. Bu geçerli ve uygulanabilir bilgileri kullanarak karar verme sürecinde birçok sektörde verilerin değerlendirilmesini veri madenciliği sağlamaktadır (2). Veri madenciliği istatistiksel yöntemlerin, matematik algoritmalarının, modelleme tekniklerinin, veri tabanı ve bilgisayar teknolojilerinin kullanıldığı bir alandır. Bu alanda kullanılan modeller tanımlayıcı veya tahmin edici modeller olabilir. Tanımlayıcı modeller ilişki analizi ve kümeleme analizinden oluşurken, tahmin edici modeller sınıflandırma ve istatistiksel tahmin yöntemlerinden oluşur. Sınıflandırma yöntemlerinden sık kullanılan algoritmalar karar ağaçları, yapay sinir ağları ve genetik algoritmalarıdır. İstatistiksel tahmin modellerinde ise regresyon analizi, diskriminant analizi ve lojistik regresyon analizi en yaygın kullanılan yöntemlerdir (16).

İstatistiksel tahmin yöntemlerinde ve sınıflandırma yöntemlerinde amaç, bağımlı ve bağımsız değişken arasındaki ilişkiyi en iyi tanımlayan, kabul edilebilir bir model oluşturmaktır. Lojistik regresyon ve karar ağaçları bu alanda bağımlı değişkenin kategorik olduğu durumlarda sıkça kullanılan yöntemlerdir (17). Karar ağaçları, istatistiksel varsayımları taşıma koşulu olmaması, uygulanmasının ve yorumlanmasının kolay olması, görsel sunumunun ön planda olması ve güvenilirliklerinin yüksek olması nedenleriyle sıkça tercih edilen yöntemlerdir.

2.2.1. Karar Ağaçlarının Yapısı

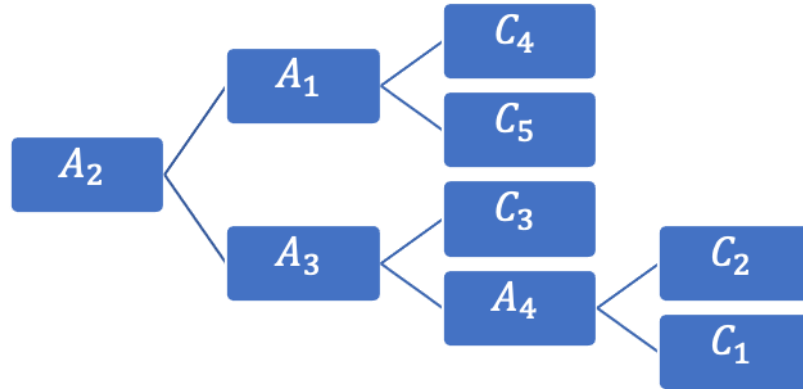
Karar ağaçları, verilerin sistematik bir şekilde sınıflandırılmasını sağlayarak, karmaşık verilerin daha anlaşılır hale gelmesine yardımcı olan bir sınıflama yöntemidir. Karar ağaçları, temel olarak iki aşamadan oluşan bir süreçle çalışır. Eğitim aşamasında model oluşturma için önceden belirlenmiş veriler kullanılır. Eğitim verisi, bir sınıflama algoritması tarafından analiz edilir ve bu süreçte bir model oluşturulur. Bu model, sınıflama kuralları veya bir karar ağacı şeklinde temsil edilir. Burada amaç, verinin içerdiği örüntülerin ve ilişkilerin belirlenmesi ve bu bilgilerin sistematik bir hale getirilmesidir. Test aşaması ise oluşturulan modelin doğruluğunu test etmeye yöneliktir. Eğitim verisi, elde edilen sınıflama kuralları veya karar ağacı ile karşılaştırılır. Eğer bu testler sonucunda elde edilen doğruluk oranı kabul edilebilir bir seviyede ise, model yeni verilerin sınıflanmasında kullanılmaya hazır hale gelir.

Karar ağaçları, bir ağaca benzer yapıdadır; kök, dallar ve yapraklardan oluşur. Bu yapı, başlangıçta tüm gözlemleri kapsayan bir kök ile başlar ve aşağıya doğru inildikçe veriyi alt gruplara ayıran dallara ayrılır. Alt gruplara ayıran her bir düğüm, bu ağaç yapısında önemli bir rol oynar. Düğümler, homojen olmayan yapılar için çocuk düğümü (*child node*) ve homojen düğümler için terminal düğüm (*parent node*) olarak adlandırılır (18). Düğümler üzerinde yapılan test işlemleri ağacın veri kaybı yaşamadan dallara ayrılmasını sağlar. Her bir düğümde test ve dallara ayrılma işlemleri ardışık olarak gerçekleşir ve bu süreç sonunda ağaç, belirli sınıflar ile sonlanır.

Karar ağacı oluşturulurken, belirlenen sorulara verilen yanıtlarla veri seti çeşitli bölümlere ayrılmakta ve bu sayede sonuçlara ulaşılmaktadır. Bu noktada, "böl ve elde et" aşaması, ağaç yapısının en hayati bileşenidir. Ağaç yapısına kök düğümden başlayarak yapraklara kadar dallanma yöntemiyle ilerlenir. Her bir düğümden, örnekleri en etkili şekilde sınıflandıracak nitelikler belirlenir. Eğer mevcut örnekler tamamıyla aynı sınıfa ait ise, yeni bir dallanma yapılmaz; bu durumda dal, yaprak düğüm olarak sonlanır. Dallanmalar, örneklerin homojenliği sağlanana kadar yani hepsi aynı sınıfa ait olana kadar veya örnekleri ayıracak nitelik kalmayana dek devam eder (19).

Örnek verecek olursak;

$D = \{t_1, \dots, t_n\}$ olmak üzere, her t_i , $t_i = \{t_{i1}, \dots, t_{in}\}$ 'den ve $\{A_1, A_2, \dots, A_n\}$ gruplarından oluşan bir veri seti olsun. Ayrıca veri setinde $C = \{C_1, \dots, C_n\}$ kadar da sınıf olsun. Her bir çocuk düğümün A_i ile tanımlandığı ve bu düğümlerden ayrılan ve bu alanla ilgili bir soruya yanıt veren yaprakların (kolların) sınıf olduğu (terminal düğümü) karar ağacı örneği Şekil. 2.2'de gösterilmiştir.



Şekil 2.2. Örnek karar ağacı (Kaynak 18'den uyarlanmıştır).

Karar ağaçları oluşturulurken verilerin hangi niteliklerine dayalı olarak bölüneceği konusunda çeşitli algoritmalar kullanılmaktadır. Bu algoritmalar, karar ağaçlarının etkinliğini ve doğruluğunu büyük ölçüde etkiler. Ayrıca kök denilen ilk düğümü oluşturan A_i 'nin farklı olması, en uçtaki yaprağa ulaşırken izlenecek yolu ve dolayısıyla sınıflandırmayı da değiştirecektir.

Karar ağacı oluşturmak için geliştirilen bu algoritmalar arasında; CHAID (Chi-Squared Automatic Interaction Detector: Otomatik Ki-Kare Etkileşim Belirleme), CART (Classification and Regression Trees: Sınıflama ve Regresyon Ağaçları), MARS (Multivariate Adaptive Regression Splines: Çok Değişkenli Uyumlu Regresyon Uzanımları), QUEST (Quick, Unbiased, Efficient Statistical Tree: Hızlı, Yansız, Etkin İstatistiksel Ağaç), SLIQ (Supervised Learning in Quest), SPRINT (Scalable Parallelizable Induction of Decision Trees), ID3, C4.5 ve C5.0 vardır. Bu algoritmalar kök ve düğüm seçimi, en iyi bölen nitelik seçimi kriterleri (dallanma kriterleri) için izledikleri yollar bakımından birbirlerinden ayrılırlar (20).

En iyi bölen nitelik seçimi için kullanılan en yaygın kriterler arasında entropi hesaplamalarına dayanan kriterler ve bunların türevleri olan bilgi kazancı (information gain) ve kazanç oranı kriterleri yer almaktadır. İkinci sıklıkta bölünme kriteri olarak genellikle safsızlık (impurity function) fonksiyonları kullanılır. Safsızlık fonksiyonunu kullanan algoritmaların başında CART algoritması yer alır. CART algoritması bölünme kriterleri "bölme uyum kriterleri" veya "goodness of split criteria" olarak da anılmaktadır. Sınıflama ağacı uygulamalarında, safsızlık fonksiyonu olarak kullanılan iki temel kriter Gini kriteri (Gini indeksi-IBM IntelligentMiner) ve Twoing bölme kriterleridir. Bu kriterler, sınıflandırma işlemlerinin etkinliğini artırmak için geliştirilmiştir. Regresyon ağaçları için ise, yine safsızlık fonksiyonuna dayanan bölünme kriterleri arasında en küçük kareli sapma (least squared deviation: LSD) ve en küçük mutlak sapma (least absolute deviation: LAD) yöntemleri yer alır. Bu yöntemler, regresyon analizi sürecinde hata payını minimize etmek amacıyla kullanılmaktadır (21).

Bu tezin uygulama aşamasında CART algoritması ve algoritmada en iyi bölen nitelik seçimi için de Gini indeksi kullanılacaktır. CART algoritmasının uygulanması bilgisayar yazılımlarında Orange DataMining ve RStudio progmanları ile hazır kodlarla yapılacaktır.

2.2.2. Sınıflandırma ve Regresyon Karar Ağaçları (Classification and regression trees (CART)) Algoritması

Karar Ağacı uygulamalarından olan CART, Breiman, Friedman, Olshen ve Stone'un çalışmaları sonucu 1980'lerde geliştirilen bir algoritma olup, ikili karar ağaçları oluşturarak, kullanıcıların verileri daha etkili bir şekilde analiz etmesine olanak tanır. Eğer bağımlı değişken kategorik bir yapıdaysa, bu yöntem Sınıflama Ağaçları (Classification Trees) olarak adlandırılırken, sürekli bir yapıdaysa Regresyon Ağaçları (Regression Trees) olarak ifade edilmektedir (22). CART, ikili bölünme (binary splitting) prensibine dayanır, yani her bir düğümde veri iki gruba ayrılır. CART'ın bu yapısı, tıbbi verilerin çeşitliliği ve belirsizliğini etkili bir şekilde modellemeye yarar. Tıbbi alanlarda, bir hastanın durumu üzerinde etkili olabilecek çok sayıda değişken bulunur. Bu değişkenler arasında tanı verileri, laboratuvar test sonuçları, yaş, cinsiyet ve genetik faktörler gibi özellikler yer alır. Her bir değişken, hastalığın varlığı veya yokluğuna dair önemli bilgiler sunar. CART'ın dinamik yapısı, tıbbi verilerin karmaşık doğasını ele alarak, klinik karar verme süreçlerini optimize etme potansiyeline sahiptir.

CART analizi üç temel adımdan oluşmaktadır. Birinci adım maksimum ağacın oluşturulmasıdır. Bu aşamada, verilerin tüm özellikleri dikkate alınarak, mümkün olan en karmaşık ağaç yapısı inşa edilir. Amaç, değişkenler arasındaki ilişkilerin en iyi şekilde temsil edildiği bir model elde etmektir. İkinci adım; oluşturulan ağaç yapılarından bağımlı değişkenle güçlü ilişkisi olanları belirlemek için yapılan budama işlemidir. Bu işlem, aşırı uyum (overfitting) riskini azaltarak, daha genel geçer ve anlamlı sonuçlar sunan daha küçük bir ağaç yapısını ortaya çıkarır. Budama, modelin daha iyi genelleme yapabilmesi için kritik bir aşamadır. Son adım ise en uygun ağaç yapısının seçilmesidir. Bu aşamada, farklı budanmış ağaçlar arasında karşılaştırmalar yapılarak, en iyi performansı gösteren ağaç belirlenir. Modelin doğruluğu ve geçerliliği, bu seçim sürecinde dikkate alınan temel kriterlerdir (23).

Ağaç oluşum sürecinde CART algoritması başlangıç veri setinde bağımsız değişkene sorduğu sorularla ikili alt gruplara ayırır. Soru sormadaki amaç veriyi kendi içlerinde en homojen iki alt gruba ayırmaktır. Bu işlem bölme kriterlerine göre

belirlenen her bir alt veri parçası için tekrarlanır ve işleme veri parçasında uç düğüme ulaşıncaya kadar devam edilir. Uç düğüm her durumda bağımlı değişken için aynı değere sahipse artık bölünme yapılmaz (24). Böylece sonunda veriyi en az hata ile sınıflayabilen CART algoritması oluşturulmuş olur.

Gini İndeksi

Sınıflama ağacı uygulamalarında en iyi bölen nitelik seçimi için kullanılan en sık yöntemlerden olan Gini indeksi, ayırma kuralı sağ ve sol olmak üzere nitelik değerini iki bölüme ayırma esasına dayanır.

Birinci adımda her bir nitelik için sağ ve sol bölünmelere karşılık gelen sınıf değerleri gruplandırılır. İkinci adımda sağ ve soldaki bölünmeler için $Gini_{sağ}$ ve $Gini_{sol}$ değerleri hesaplanır (Eşitlik 7, Eşitlik 8).

k : Sınıf sayısı

T_{sol} : Sol düğümdeki örnek sayısı

$T_{sağ}$: Sağ düğümdeki örnek sayısı

L_i : Sol taraftaki i kategorisindeki örneklerin sayısı

R_i : Sağ taraftaki i kategorisindeki örneklerin sayısı olmak üzere;

$$Gini_{sol} = 1 - \sum_{i=1}^k \left(\frac{L_i}{T_{sol}} \right)^2 \text{-----Eşitlik-7}$$

$$Gini_{sağ} = 1 - \sum_{i=1}^k \left(\frac{R_i}{T_{sağ}} \right)^2 \text{-----Eşitlik-8}$$

Üçüncü adımda, her j niteliği için, n eğitim kümesindeki satır sayısı olmak üzere $Gini_j$ değeri hesaplanır (Eşitlik-9).

$$Gini_j = \frac{1}{n} (|T_{sol}| Gini_{sol} + |T_{sağ}| Gini_{sağ}) \text{-----Eşitlik-9}$$

Dördüncü adımda her j niteliği için hesaplanan $Gini_j$ değerleri arasında en küçük olanı seçilir ve bölünme bu niteliğe göre yapılır (21).

Beşinci adımda bu basamaklar tekrar edilir. Ve sonunda uygun ağaç modeli oluşturulur.

2.3. Bulanık Karar Ağaçları (Fuzzy Decision Trees)

Bulanık karar ağaçları, sınıflama ve regresyon ağaçları algoritmalarında bulanık mantığın kullanılması ile ortaya çıkan matematiksel modellerdir.

2.3.1. Bulanık Mantık ve Bulanık Kümeleme

Bulanık mantık geleneksel ikili mantığın aksine, belirsizliğin ve kesin olmamanın 0 ile 1 arasında değişen doğruluk dereceleri kullanılarak matematiksel olarak modellenmesidir. Özellikle, tıbbi verilerin çoğu zaman subjektif ve karmaşık olduğu durumlarda verilerin daha iyi işlenmesine olanak sağlar. Bulanık mantık yaklaşımı aslında olgu ve olayların, karar verme mekanizmalarının kesintisiz süreçler olduğunu, kesin hatlar ile ayıramayacağını ifade eder. Bulanık mantıkta siyah-beyaz, doğru-yanlış, başarılı- başarısız şeklinde ayırım yoktur.

Belirsizlik kavramı ilk defa 1920li yıllarda Heisenberg tarafından Heisenberg Belirsizlik İlkesi ile bilim alanına sunulmuş, Polonya'lı mantıkçı Jan Lukasiewicz ise önermelerin sadece bir veya sıfır doğruluk değeri alabildiği klasik mantıktan farklı olarak bir ve sıfır arasında da kesirli doğruluk değeri alabildiği mantık ilkelerini oluşturmuştur (3, 25). 1937'de kuantum felsefecisi Max Black yayımlanan bir makalesinde liste ya da nesnelerden oluşan kümelerle ilk bulanık küme eğrilerini çizmiştir (26). Daha sonra 1965 yılında Zadeh bu alana adını veren "Bulanık Küme Teorisini" ortaya koymuştur (27). Zadehin teorisi önceleri kabul görmesede, 1970lerde özellikle mühendislik uygulamalarında bulanık mantık kuramının kullanmaya başlamasıyla kurama olan ilgi dünya çapında artmıştır. Günümüzde savaş teknolojileri, uzay araçları, yüz tanıma sistemleri, internet teknolojileri, yapay zekâ, elektron teknolojileri, robot teknolojileri, gibi farklı alanlarda da kullanılmaktadır (3).

Bulanık önermelerin doğruluğu ve yanlışlığı hakkında kesin bir şey söylenemeyeceğinden dolayı bunların doğruluk değeri, gerçel sayılar kümesinden, bir sayıyla derecelendirilir. Bir bulanık önerme derecesine göre hem doğru ve hem de

yanlış olabilir. Bulanık bir önerme için doğru değildir denilmiş ise bu yanlıştır anlamına gelmez. Bir önerme 0,8 derecesinde doğru ise aynı önerme 0,2 derecesinde de yanlıştır.

Bulanık kümelerin matematiksel gösterimi aşağıdaki gibidir (26):

U boş olmayan bir küme olduğunda, U 'da ki bir bulanık A kümesi;

$\forall x \in U$ için $\mu_A : U \rightarrow I = [0,1]$ fonksiyonu ile gösterilir.

μ_A : Bulanık kümeye karşılık gelen üyelik fonksiyonudur.

$\mu_A(x)$: x 'in A 'ya ait olma veya üyelik derecesidir.

Her üyelik fonksiyonu, bir bulanık evrensel kümenin elemanlarını $[0,1]$ aralığındaki bir sayıya karşılık getiren fonksiyonudur. Üyelik fonksiyonları üçgen, yamuk, sigmoidal veya çan eğrisi gibi farklı şekillerde olabilir. Çalışılan veri setinin alanına göre hangi fonksiyon şeklinin uygun olduğu belirlenebilir. Ancak birçok uygulamada şekil değişikliklerine karşı ciddi bir farklılık ve duyarlılık bulunmamaktadır.

Üyelik derecesi, çalışma yapılan alana ait her bir bireye bulanık kümede atanan değer olarak tanımlanır. Bu nedenle her bireyin kümeye ait olması farklılaşır. Üyelik dereceleri 0 ve 1 arasındaki reel sayılarla temsil edilir. Tam üye olma veya olmama durumu bulanık kümede sırasıyla 1 ve 0 değerleriyle temsil edilir.

Bulanık küme işlemleri, Tablo. 2. 1 'de özetlenmiştir (27, 28).

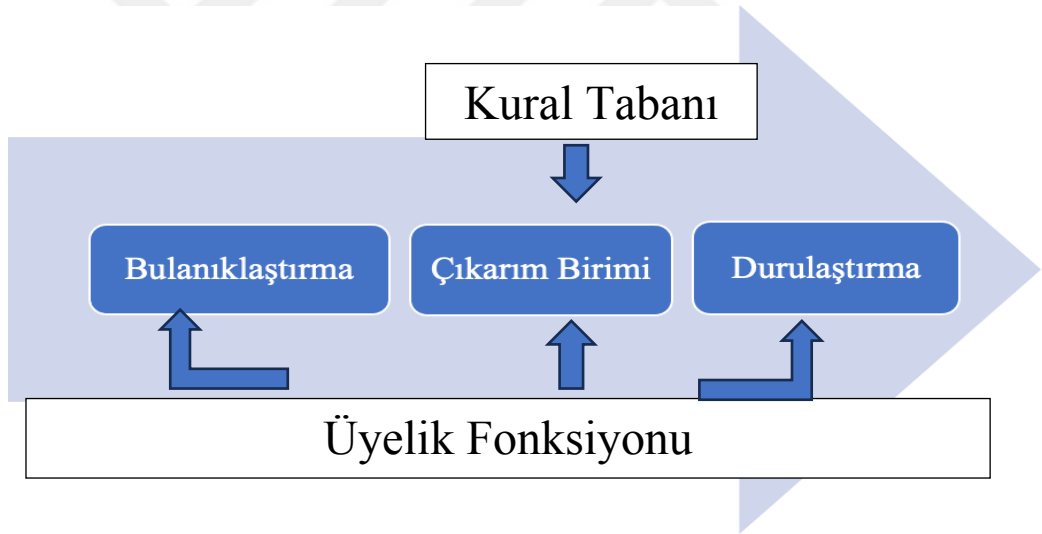
Tablo 2.1. Bulanık küme işlemleri

	Küme İşlemi	Açıklaması
$(A = B)$	$\mu_A(x) = \mu_B(x)$ $\forall x \in U$	İki kümede x 'in üyelik fonksiyonları eşittir.
$(A \cap B)$	$\min\{\mu_A(x), \mu_B(x)\}$ $\forall x \in U$	İki kümede x 'in üyelik fonksiyonları en küçük ortak değeri alır.
$(A \cup B)$	$\max\{\mu_A(x), \mu_B(x)\}$ $\forall x \in U$	İki kümede x 'in üyelik fonksiyonları en büyük ortak değeri alır.
$(A \subseteq B)$	$\mu_A(x) \leq \mu_B(x)$ $\forall x \in U$	x 'in B 'deki üyelik fonksiyonu A 'dan eşit ya da büyüktür.
(A')	$1 - \mu_A(x)$ $\forall x \in U$	

Bulanık küme sistemlerinde de diğer matematiksel modellerde olduğu gibi zamanla yetersizlikler yaşanmıştır. Bu yetersizlikleri çözmek amacıyla Tip-2 bulanık küme sistemi geliştirilmiştir (29). Geleneksel bulanık küme kavramı olarak bilinen Tip-1 bulanık küme sisteminin modelleyemediği durumlarda tercih edilen Tip-2 bulanık sistem, verinin gürültülü olduğu, sistemin aktif ölçümlerinin belirsiz olduğu durumlarda kullanılabilir (30).

2.3.2. Bulanık Sistem Kuralları

Bulanık mantık modeli oluşturulurken bulanık çıkarım sisteminin basamaklarına uyulmalıdır. Bulanık sistemin girdi ve çıktıları en baştan belirlenmelidir. Bu sistemin yapısı veri tabanı, kural tabanı, bulanıklaştırma, çıkarsama ve durulaştırma basamaklarından oluşmaktadır (28, 31). Bulanık sistem yapısı, Şekil. 2.2’de gösterilmiştir.



Şekil 2.3. Bulanık sistem akış şeması

Tüm üyelik fonksiyonlarının bulunduğu alan veri tabanıdır. Kural tabanı ise belirlenen girdi ve çıktılara göre çıkarım yapmak için kullanılacak kurallar kümesidir. Bulanıklaştırma işlemi üyelik fonksiyonlarını kullanarak veri tabanında bulunan değerleri bulanık değerlere (0 ve 1 arasında) dönüştürme işlemidir. Çıkarsama basamağında elde edilen bulanık değerlerden sonuçlar çıkarılır (28). Çıkarım sistemi bilgi tabanını kullanır ve neye göre nasıl çıkarım yapılacağı farklı yöntemlerle belirlenebilir. Takagi-Sugeno-Kang yöntemi, Mamdani yöntemi, Larsen yöntemi,

Tsukamoto yöntemi gibi farklı çıkarsama yöntemleri bulunmaktadır (32). Durulaştırma basamağında ise bulanık değerler aralığında gelen veriler gerçek değerlere dönüştürülür (28).

2.3.3. Bulanık Karar Ağaçları

Karar ağaçları, belirli kurallara dayanarak kararlar almak için kullanılırken, bulanık karar ağaçları, verilerin belirsizlik taşıdığı durumlarda tercih edilmektedir. Bu yöntemde, her bir karar düğümünde bulanık mantık kuralları kullanılarak seçim yapılır. Bulanık karar ağaçları, kesin olmayan gerçek dünya verilerinde birçok araştırmacı tarafından kullanılmış ve klasik karar ağaçlarına göre verinin ayrımının araştırmacı tarafından kesin olarak yapılamadığı, kayıp verinin olduğu veya gürültülü verilerde kullanışlı olduğu savunulmuştur (33).

Bulanık karar ağaçlarının oluşturulması veriyi bulanıklaştırma, bulanık karar ağacını oluşturma, bulanık karar ağacından sınıflandırma kurallarını oluşturma ve bu bulanık kuralları, sınıflandırmada kullanma basamaklarından oluşur. Süreçte üçgen veya yamuk bulanık üyelik fonksiyonları kullanılabilir (34). Bulanık karar ağacı algoritması, J.R. Quinlan tarafından popüler hale getirilen ID3 karar ağacı algoritmasına benzerdir (35).

Sınıflandırma ve karar verme süreçlerinde ID3 algoritması ile bulanık kümeleme yöntemi birleştirilerek Tip-1 Bulanık Karar Ağaçlarının kullanımı için Bulanık ID3 Algoritması geliştirilmiştir (36, 37). ID3 karar ağacı algoritmasında aşağıdan yukarı (kökten alt dallara doğru) ve *greedy search* (sonuca en yakın durum) tekniklerini kullanılır. Deneysel çalışmalarda, bulanık ID3 algoritmasının etkinlik, basitlik ve genelleştirici kapasitesinin yüksek olduğu gösterilmiştir. Bu algorithmada en iyi bölen kriteri olarak kazanç oranı, bilgi kazancı veya gini indeksi kullanılabilir (34).

Bilgi kazancı kriteri entropi hesaplamalarına dayanır. Entropi örneklerin özelliğini ölçer. Büyük entropi veri setinin saf olmadığını gösterir. Entropi 0 olması veri setindeki tüm örneklerin belirlenen değişken için aynı sınıfa ait olduğu anlamına gelir (38). Entropi ile en yüksek hesaplanan bilgi kazancı değeri seçilir. Dolayısıyla

bilgi kazancı entropideki azalmaya dayanır. Belirlenen değişken için bilgi kazancı hesaplaması Eşitlik-10 ile hesaplanır.

$$\text{Bilgi kazancı} = \text{Entropi (Veri seti)} - \text{Entropi (Değişken)} \text{-----Eşitlik-10}$$

$\text{Entropi (Değişken)}$ ise Eşitlik-11 ile hesaplanır.

$$\text{Entropi (Değişken)} = - \sum_{v \in \text{Değişken}(A)} \frac{|S_v|}{|S|} \times \text{Entropi (S}_v\text{)} \text{-----Eşitlik-11}$$

S veri setinde; $|S|$ tüm örneklem sayısı

S_v değişken için alt grup veri setinde; $|S_v|$ alt grup örneklem sayısı

Bir veri seti için bilgi kazancı aşağıdaki basamaklarda hesaplanır (39):

- 1- Tüm veri seti için entropi hesaplanır.
- 2- Tüm değişkenler için entropi hesaplanır.
- 3- Tüm değişkenler için bilgi kazancı hesaplanır.
- 4- En büyük bilgi kazancı (en küçük entropi) olan değişken seçilir.

Bu tezin uygulama Tip-1 bulanık sistem yaklaşımı ile oluşturulan Tip-1 Bulanık Karar Ağacı modeli oluşturulacaktır. Bulanık karar ağacı oluşturmada ID3 algoritması ve en iyi bölen kriteri olarak da bilgi kazancı kriteri kullanılacaktır. Tip-1 Bulanık karar ağacı algoritması RStudio bilgisayar yazılım programında tarafımızca oluşturulan kodlarla yapılacaktır.

2.4. Sınıflandırma ve Regresyon Karar Ağaçları, Bulanık Karar Ağaçları ve Lojistik Regresyon Yöntemlerinin Karşılaştırılması

Sınıflama ve regresyon karar ağaçları, bulanık karara ağaçları ve lojistik regresyon tıbbi durumlarda karar verme sistemlerinde yaygın olarak kullanılan yöntemlerdir, ancak her birinin belirli avantajları ve sınırlamaları bulunmaktadır. Bulanık karar ağaçları, belirsiz verilerle başa çıkma yetenekleri ile öne çıkarken, lojistik regresyon, modelin basitliği ve yorumlanabilirliği ile avantaj sağlar. Öte

yandan, CART, görselleştirme avantajı ve sınıflandırma doğruluğu ile dikkat çeker. Bununla birlikte, her bir yöntem, tıbbi verilerin doğasına ve analiz edilen problem tipine göre farklı performans sergileyebilir. Literatürde, bu üç yöntemin tıbbi veri analizi ve karar destek sistemlerinde karşılaştırılmasına yönelik çeşitli çalışmalar bulunmaktadır. Lojistik regresyon ile CART modelleri ile hastaların yoğun bakım ünitesine yatış kriterlerinin değerlendirildiği bir çalışmada, lojistik regresyonun istatistiksel olarak anlamlı değişkenler üzerine etkin olduğu, CART'ın ise daha çok sonlanım üzerine sonuç değerlendirmede etkin olduğu bulunmuştur. Her iki yöntemin de hastalık tahmininde yüksek doğruluk sağladığı belirtilmiştir (40). Yine hipertansiyon risk faktörlerinin değerlendirildiği bir çalışmada CART ile oluşturulan modelin hipertansiyon risk tespitinde lojistik regresyondan daha iyi olduğu belirtilmiştir (41). Hastalık tespitinde CART'ın yüksek duyarlılık ancak düşük özgüllükte çalıştığını gösteren çalışmalar da mevcuttur (42). Bunun yanı sıra, bulanık karar ağaçları ile yapılan çalışmalar, özellikle çok sayıda belirsiz ve eksik veri ile karşılaşılan durumlarda, daha esnek ve doğru sonuçlar elde edilmesini sağladığını göstermektedir (43). Üç modelin karşılaştırıldığı bir kanser çalışmasında ise kanser yayılımını öngörmeye her üç yöntemin de benzer olduğu ancak bulanık çıkarımın ve CART'ın lojistik regresyon analizine göre biraz daha iyi olduğu gösterilmiştir (16). Her 3 yöntemin karşılaştırılması Tablo. 2. 2.'de özetlenmiştir.

Tablo 2.2. Sınıflandırma ve regresyon karar ağaçları, bulanık karar ağaçları ve lojistik regresyon analizinin karşılaştırılması

Özellik	Sınıflandırma ve Regresyon Karar Ağaçları	Bulanık Karar Ağaçları	Lojistik Regresyon
Veri türü	Hem sınıflandırma hem regresyon için uygundur.	Belirsiz ve eksik verilerle iyi çalışır.	İkili sonuçlarla çalışır.
Modelin yorumlanabilirliği	Yüksek (Görsel ağaç yapıları)	Yüksek (Esnek kurallar)	Yüksek (Sade ve anlaşılır)
Karmaşıklık	Orta (Derinlik arttıkça karmaşıklık artar)	Orta (Hesaplama gereksinimi yüksek)	Düşük (Basit model)
Eğitim verisi gereksinimi	Orta (Genellikle büyük veri setleri gerektirir)	Yüksek (Büyük veri setleri gerekebilir)	Düşük (Daha az veri ile çalışabilir)
Hastalık türleriyle uyumluluk	Karmaşık hastalıklar, etkileşimli veriler	Belirsiz ve eksik verilerin bulunduğu hastalıklar	İkili sonuçlar, net sınıflandırmalar

Bu tez, tıbbi karar destek sistemlerinde CART, bulanık karar ağaçları ve lojistik regresyon modellerinin tıbbi bir veri setinde karşılaştırılması üzerine planlanmıştır. Özellikle, her bir yöntem ile elde edilen sonuçların doğruluğu, hesaplama verimliliği ve esnekliği analiz edilecektir. Literatür taramaları ve analiz sonuçları ile hangi yöntemin daha uygun olduğu belirlenmeye çalışılacaktır. Sonuç olarak, bu çalışma ile tıbbi karar süreçlerinde kullanılacak en uygun analitik yöntemi seçmeye yardımcı olmak amaçlanmaktadır.

3. UYGULAMA

Bu bölümde tıbbi alanda yapılmış bir çalışmanın veri setinde CART, bulanık karar ağacı ve lojistik regresyon analiz yöntemleri uygulamalı olarak karşılaştırılacaktır. Karşılaştırma yöntemleri bilgisayar yazılım programlarından IBM SPSS Statistics25 (44), RStudio 4.4.2 (45) ve Orange Datamining 3.38.1 (46) programları ile yapılmıştır.

3.1. Tanımlayıcı İstatistikler

Gerçek hasta verilerinin kullanıldığı çalışmanın amacı son dönem böbrek yetmezliği olan ve rutin programlı hemodiyalize giren hastalarda malnutriyonu gösteren özelliklerin belirlenmesiydi (Çalışma Başkent Üniversitesi Klinik Araştırmalar Etik Kurulu tarafından onaylanmıştır; No.19/29; Tarih: 20.02.2019; proje no. KA19/36). Toplam 155 hastanın değerlendirildiği çalışmada yaş, cinsiyet, normalized protein catabolic rate (nPCR), vücut kitle indeksi (BMI), son bakılan CRP (CRPSon), serum albümin (Alb) ve serum malnutriyon biobelirteçi olarak değerlendirilen growth differentiation factor-15 (GDF-15) değişkenleri kullanıldı. Malnutrisyonu tespit etmek için malnutrisyon-inflamasyon skoru (MIS) kullanıldı, MIS total değeri ≤ 8 olan hastalar malnutriyonu olan (Class=1), > 8 olan hastalar malnutrisyonu olmayan (Class=0) olarak değerlendirildi (47).

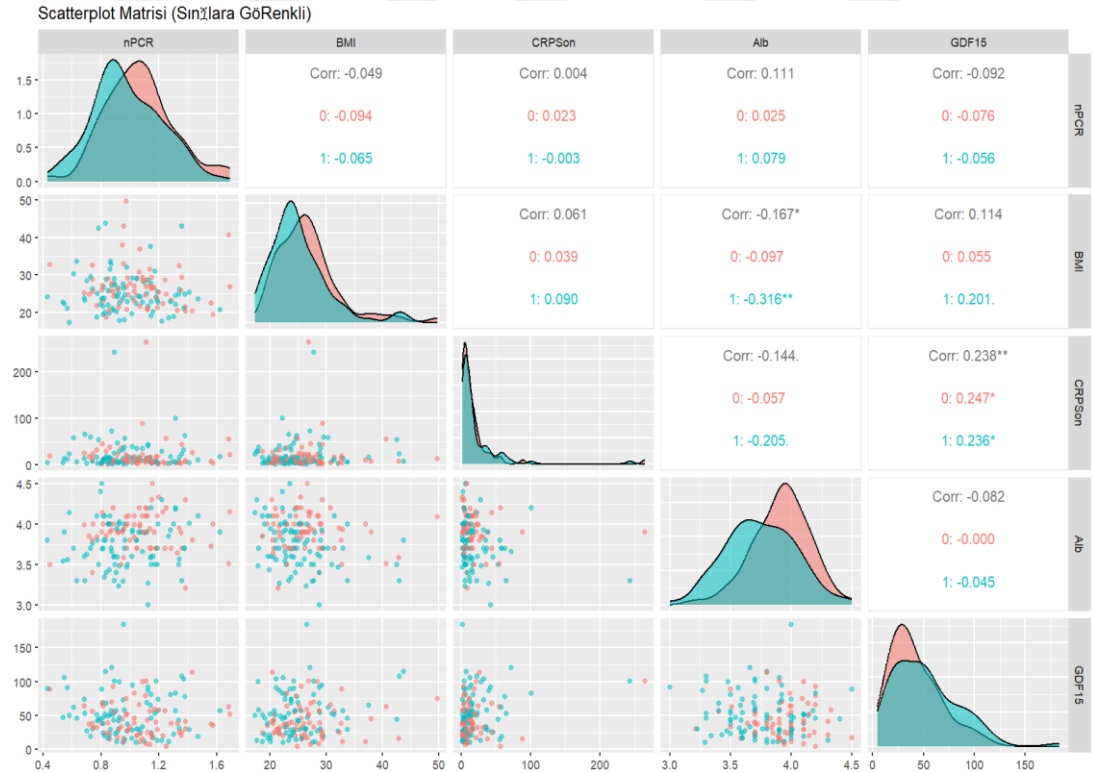
155 hastanın 87'sinde (%56) MIS ≤ 8 (Class1), 68'inde (%44) MIS > 8 (Class0) olarak tespit edildi. Hastaların 59'u (%38) kadındı. Hastaların yaş ortalaması $63,21 \pm 14,4$ iken malnutrisyonu olan ve olmayan hastalar arasında yaş ortalamaları ve cinsiyet sıklığı açılarından fark yoktu ($p=0,7$, $p=0,44$).

Bağımsız değişkenlerin dağılımları Tablo 3.1 'de özetlenmiştir. Gruplar arasında farka bakıldığında malnutriyonu olan grupta Alb değerinin düşük ($p=0,001$), GDF15 değerinin yüksek olduğu bulunmuştur ($p=0,033$).

Tablo 3.1 Bağımsız değişkenlerin tüm veri setinde ve gruplar arasında dağılımları

	Tüm hastalar (n=155)	MIS ≤8 (Class1) (n=87)	MIS >8 (Class0) (n=68)	p
Yaş	64 (IQR:16)	63 (IQR:17)	69.50 (IQR:16)	0,005
Cinsiyet (Kadın)	60 (%38,7)	23 (%14,6)	37 (%23,4)	0,409
nPCR(g.kg ¹ .day ¹)	1,017 ± 0,339	1,067 ± 0,239	0,978±0,235	0,001
Body mass index (kg/m ²)	25,81 ± 5,45	26,71 ± 5,59	25,19 ± 5,40	0,086
C-reactive protein (mg/l)	19,62 ± 3,21	18,56 ± 4,12	20,78 ± 3,27	0,674
Serum albumin (g/dL)	3,82 ± 0.26	3,91 ± 0,23	3,72 ± 0,29	0,001
GDF-15 (ng/mL)	47,39 (IQR:46,99)	37,42 (IQR:33,54)	47,82 (IQR:47,95)	0,003

Bağımsız değişkenlerin kendi aralarındaki ilişki değerlendirildiğinde Alb ve BMI arasında zayıf ama negatif ($r=-0,167$, $p=0,038$); CRPSon ve GDF15 arasında zayıf ama pozitif ($r=0,238$, $p=0,003$) bir ilişki olduğu görülmüştür. Bulgular Şekil 3.1 'de gösterilmiştir.

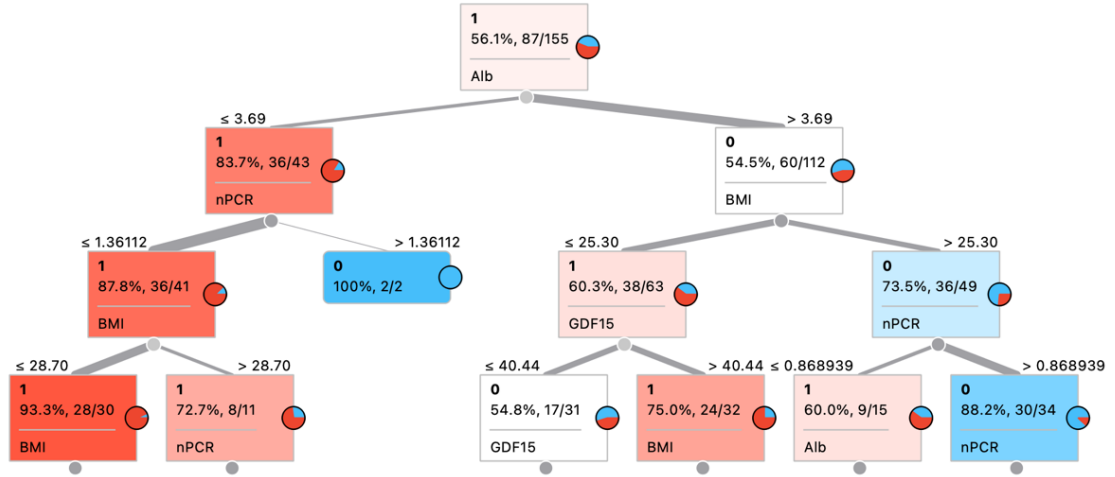


Şekil 3.1. Bağımsız değişkenlerin korelasyon dağılımı

3.2. Sınıflandırma ve Regresyon Karar Ağacı Uygulaması

Hemodiyaliz hastalarının malnutrisyon risk faktörleri açısından değerlendirilmesinde karar ağacı sınıflama algoritmalarından CART'ın Orange DataMining yazılımındaki uygulanması değerlendirildiğinde; malnutrisyonu olan 87 hastanın 36'sında serum albümin değeri $\leq 3,69$ mg/dl olarak tespit edildi. CART diagramında ilk dallanmanın serum albümin değişkenine göre yapıldığı ve cut-off değeri olarak 3,69 alındığı görüldü. Diğer bir deyişle serum albümini düşük toplam 43 kişiden 36'sının hasta olduğu görüldü. İkinci dallanmanın nPCR değişkenine göre yapıldığı görüldü. $nPCR \geq 1,36$ olan hastaların hiçbirinde malnutrisyon yoktu. $nPCR \leq 1,36$ olan 41 kişinin 36'sının hasta olduğu görüldü. Üçüncü dallanmanın BMI değişkenine göre yapıldığı ve nPCR düşük 41 kişiden 30'unda BMI $\leq 28,70$ olduğu görüldü. Bu 30 kişinin 28'inde malnutrisyon vardı.

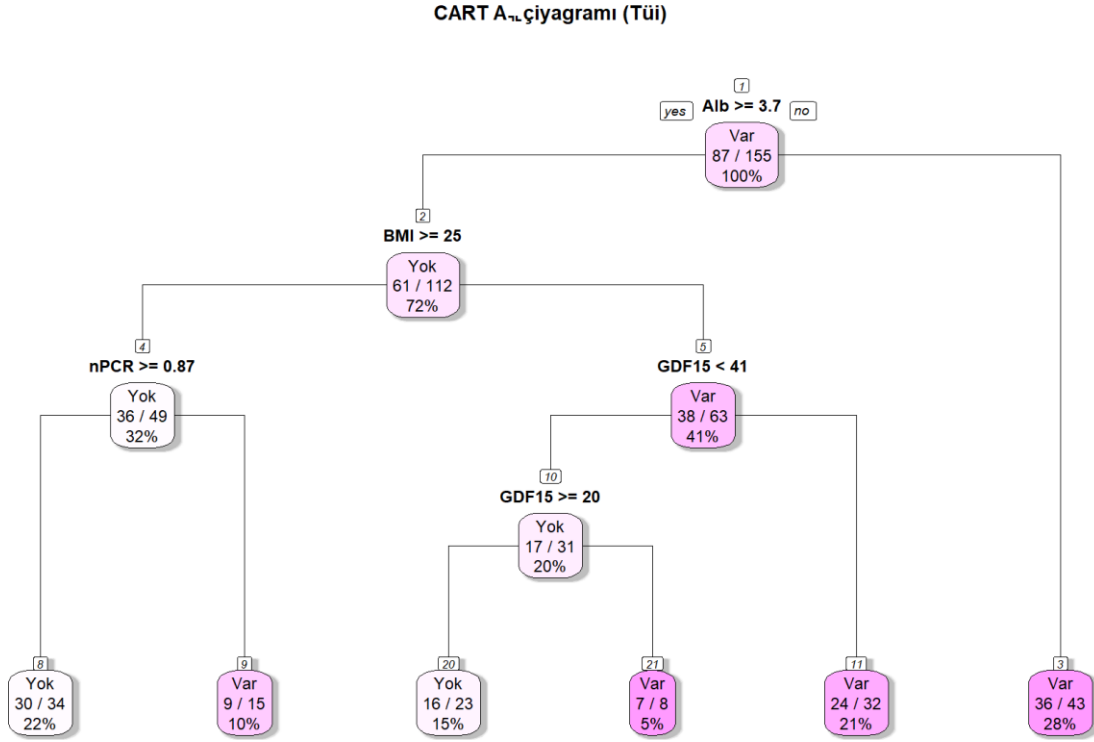
CART uygulamasının Orange DataMining yazılımı ile uygulaması Şekil 3.2 'te gösterilmiştir.



Şekil 3.2. CART algoritması ile malnutrisyonun değerlendirilmesi (Orange DataMining)

CART algoritmasının RStudio programında uygulanan tüm veri seti analiz sonuçları, DataMining çıktısı ile benzer olarak değerlendirildi. Tüm veri setindeki analiz sonuçlarına bakıldığında serum albümin $\geq 3,7$ olmayan 43 hastanın 36'sında

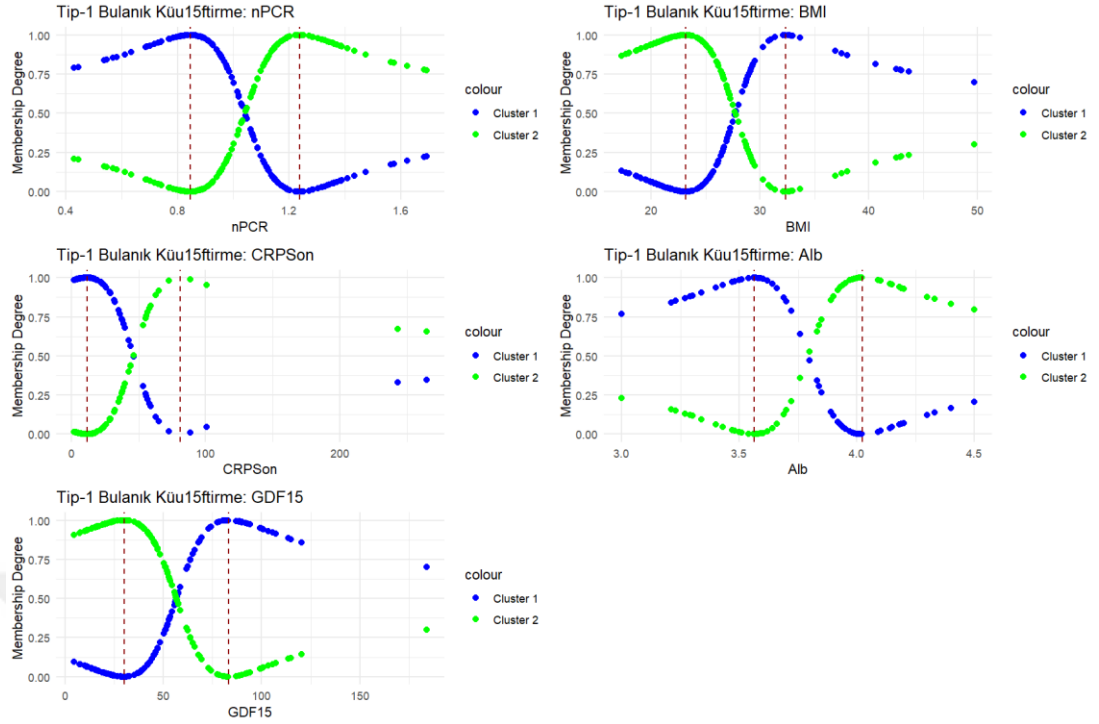
malnutrisyon vardı. CART algoritmasının tüm veri seti için uygulamasının çıktısı Şekil 3.3 ile gösterilmiştir.



Şekil 3.3. CART algoritması ile malnutrisyonun değerlendirilmesi (RStudio)

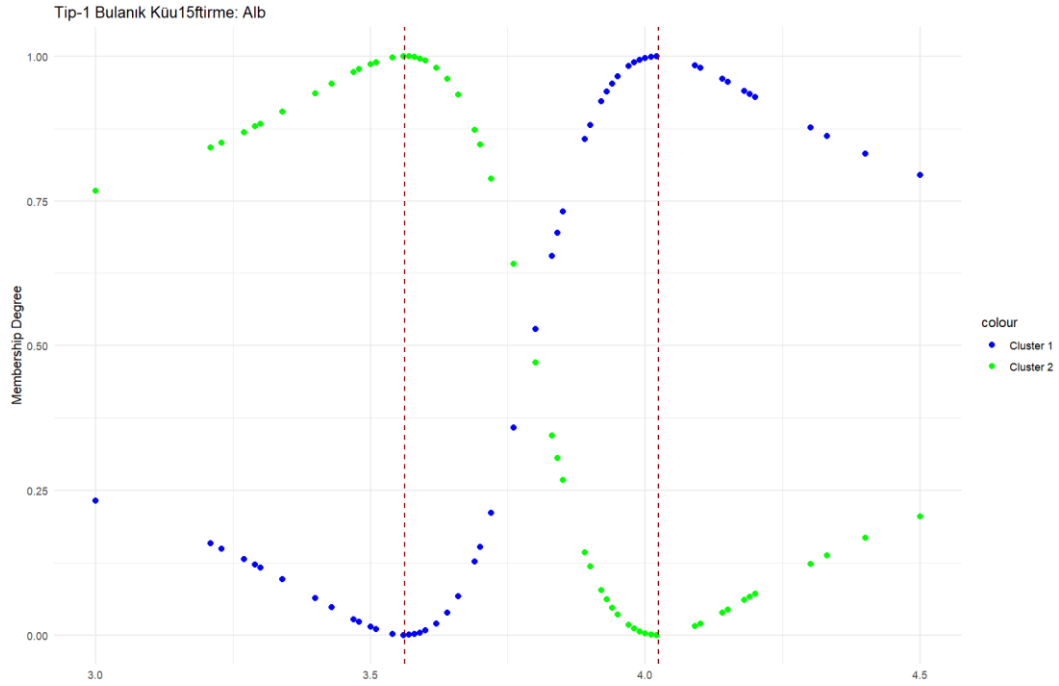
3.3. Bulanık Karar Ağacı Uygulaması

Veri seti için bulanık karar ağacı uygulamasında RStudio yazılım programı kullanılmış ve kodlar tarafımızca oluşturulmuştur. Bulanık küme oluşturmada ilk basamak olan bulanıklaştırma işlemi için veri setinde farklı bulanıklaştırma katsayıları denendi. En optimal bulanıklaştırma katsayısı $m=2$ ve küme sayısı 2 olarak seçildi, tüm bağımsız değişkenlere tüm veri setinde uygulandı. Tüm veri seti için küme grafikleri Şekil 3. 4 de gösterilmiştir.

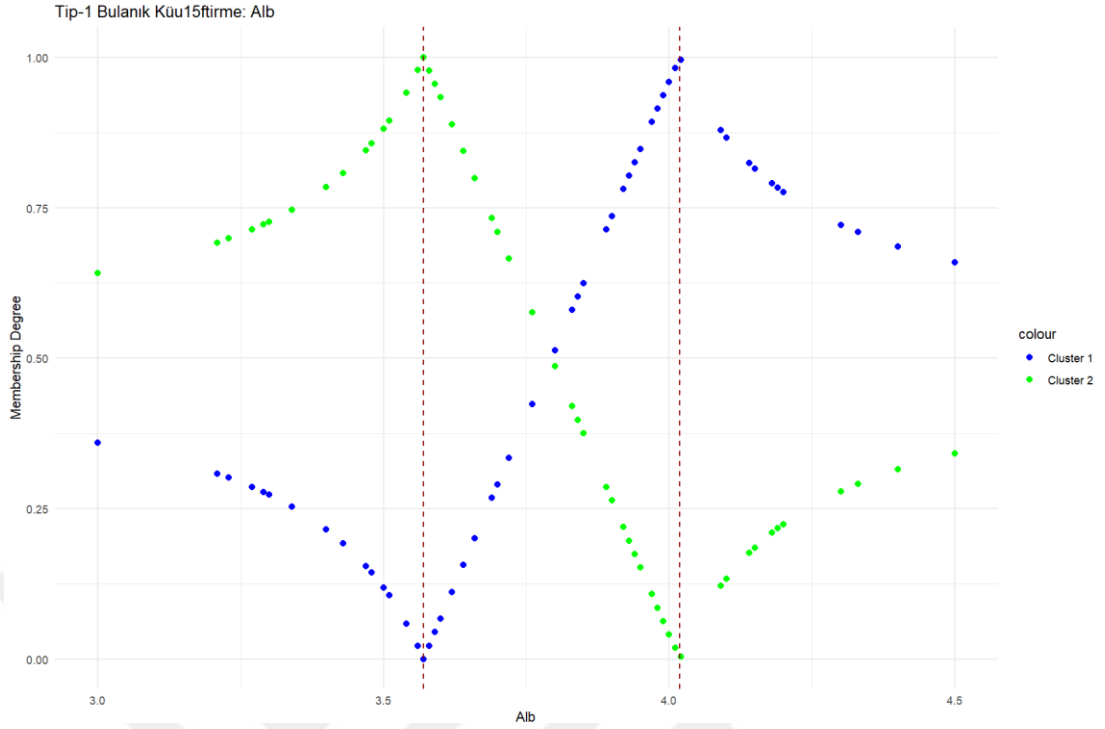


Şekil 3.4. Bulanıklaştırma katsayısı $m=2$ olduğunda bağımsız değişkenler için kümeler

Örnek olarak serum albümin $m=2$ ve $m=3$ alındığındaki kümeleştirme grafikleri sırasıyla Şekil 3. 5 ve Şekil 3. 6’da gösterilmiştir.

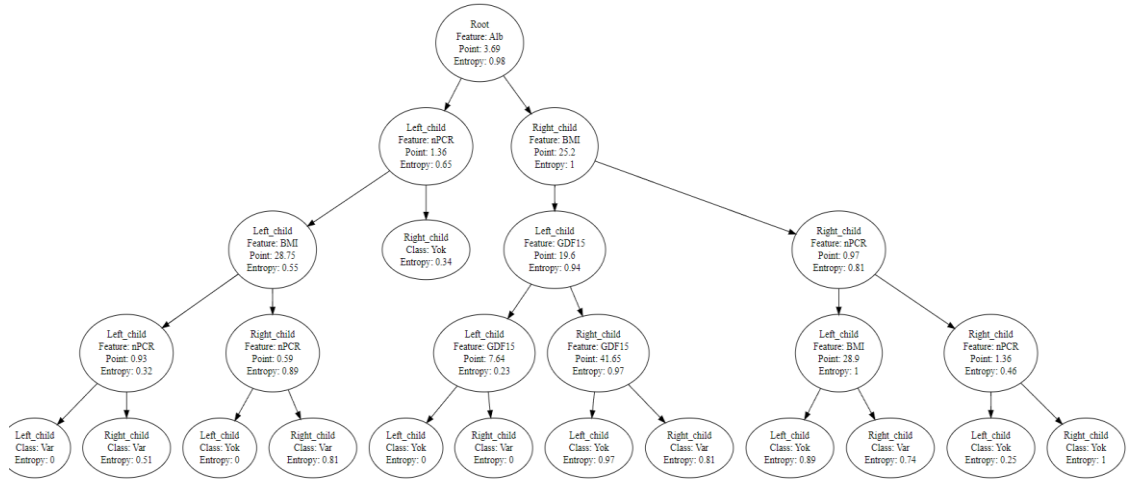


Şekil 3.5. Bağımsız değişken serum albümin için bulanıklaştırma katsayısı $m=2$ için kümeler



Şekil 3.6. Bağımsız değişken serum albümin için bulanıklaştırma katsayısı $m=3$ için kümeler

Bulanık karar ağacı tüm veri seti çıktıları incelendiğinde incelendiğinde ilk düğümde ayırma kriteri olarak yine serum albümin değeri alındığı ve cut-off değerinin $\geq 3,69$ mg/dl olduğu görüldü. Tip-1 Bulanık Karar Ağacı uygulamasının RStudio yazılımı ile uygulamasının çıktısı Şekil 3. 7 'de gösterilmiştir.



Şekil 3.7. Tip-1 bulanık karar ağacı algoritması ile malnutrisyonun değerlendirilmesi

3.4. Lojistik Regresyon Analizi Sonuçları

Lojistik regresyonun sonuçlarından sınıflama tablosu sonuçları Tablo 3.2 ‘de özetlenmiştir. Lojistik regresyon modeli oluşturulken geriye dönük eleme yöntemi (backward elimination) kullanılarak yapılan bağımsız değişkenlerim katsayı sonuçları Tablo 3.3 ‘de özetlenmiştir.

Tablo 3.2. Lojistik regresyon sınıflama tablosu

		Malnutrisyon	
		Yok	Var
Malnutrisyon	Yok	40	28
	Var	22	65

Tablo 3.3. Lojistik regresyon bağımsız değişkenlerin katsayı tablosu

	Wald	Exp (B)	p	%95 Güven Aralığı	Odds Oranı (%95 Güven Aralığı)
nPCR	2,712	0,886	0,100	0,768-1,023	0,231 (0,047-1,033)-
BMI	8,157	0,901	0,004	0,839-0,968	0,901 (0,836-9,650)
Alb	15,381	0,052	0,000	0,012-0,227	0,055 (0,011-2,250)
GDF15	4,745	1,015	0,029	1,001-1,028	1,014 (1,001-1.028)

Model uyumu için yapılan Hosmer-Lemeshow test istatistiği sonucu $p=0,545$ bulundu.

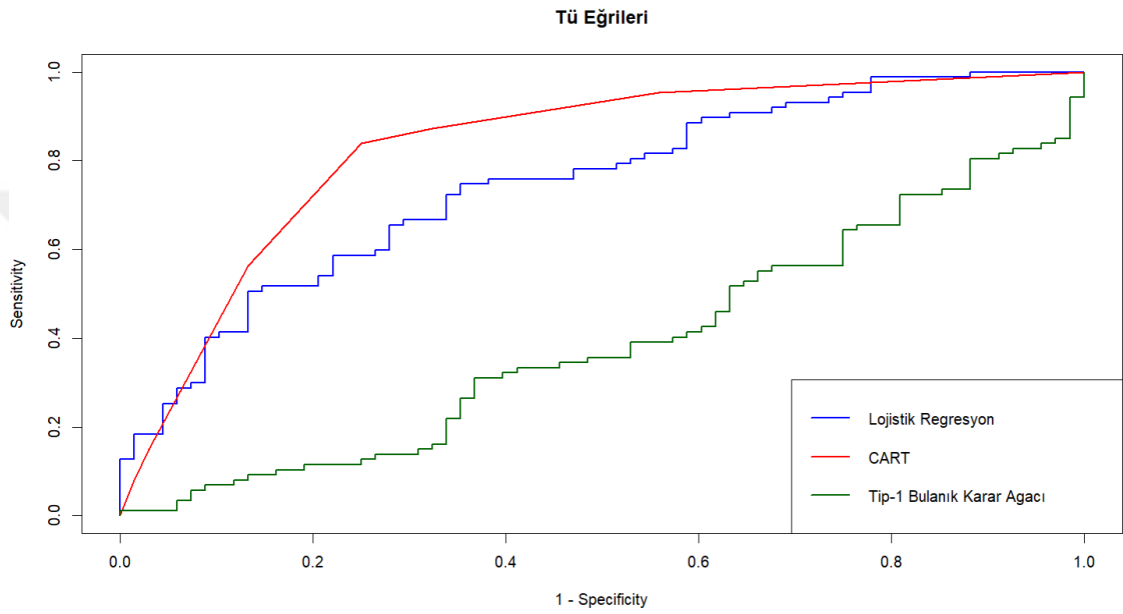
3.5. CART, Bulanık Karar Ağacı ve Lojistik Regresyon Analiz Sonuçlarının Karşılaştırılması

Her üç modelin doğruluk oranlarına bakıldığında; CART algoritmasının 0,652, Tip-1 Bulanık Karar Ağacı algoritmasının 0,456, lojistik regresyonun 0,760 olduğu görüldü. Sonuçlar Tablo.3.4’ de gösterilmiştir.

Tablo 3.4. Modellerin tüm veri seti için doğruluk oranlarının karşılaştırılması

Model	Doğruluk	Duyarlılık	Özgüllük
CART	0,652	0,50	0,769
Bulanık Karar Ağacı	0,456	0,45	0,461
Lojistik Regresyon	0,760	0,60	0,884

Tüm modellerinin sınıflandırma performansının değerlendirilmesinde ROC eğrisi ve eğri altında kalan alan (AUC) hesaplaması yapıldığında Tip-1 Bulanık Karar Ağacı algoritması için eğri altında kalan alan 0,390 (%95 GA; 0,301-0,479) iken lojistik regresyon için eğri altında kalan alan 0,749 (%95 GA; 0,673-825) olarak bulundu. CART için eğri altında kalan alan 0,830 (%95 GA; 0,764-0,897) olarak hesaplandı. ROC eğrisi grafikleri Şekil 3. 8’ de gösterilmiştir.



Şekil 3.8. Lojistik regresyon, CART ve tip-1 bulanık karar ağacı ROC eğrisi grafikleri

Her üç algoritmanın analizleri için gerekli uygulama süreleri benzer çıktı.

4. TARTIŞMA

Malnutrisyon hemodiyaliz hastalarında mortalite ve morbidite ile ilişkili ciddi bir sorundur. Hastaların aralıklı olarak malnutriyon ve malnutriyona sebep olan nedenler açılarından değerlendirilmeleri gerekir. Ancak hastalar değerlendirilirken kullanılan parametreler ve bu parametrelerin önemi net değildir. Bu tez ile malnutriyon değerlendirmesinde kullanılan risk faktörlerinin farklı istatistiksel yöntemlerle değerlendirilmesi amaçlandı. Serum albumin, nPCR, BMI ve serum GDR15 risk faktörü olarak değerlendirildiği bu tezde hem yöntemlerin sınıflama performansları hem de risk faktörleri ile ilgili cut-off değerleri değerlendirildi.

Sınıflama ve Karar Ağacı, Tip-1 Bulanık Karar Ağacı modellerinin ve lojistik regresyon analizinin değerlendirildiği bu tezde modellerin sınıflandırma performanslarından CART ve lojistik regresyonun bu veri seti için benzer olduğu; ancak bulanık kümeleme yönteminin diğerlerine göre daha düşük olduğu görüldü. Veri setinde hasta sayısı düşük olduğundan her iki karar ağacı analizlerinde veri seti eğitim ve test seti olarak ayrılmadı, tüm veri seti kullanıldı.

Karar ağaçlarının oluşumunda CART algoritması incelendiğinde, ilk dallanma kriteri olarak en önemli bağımsız değişken olarak serum albümin değerinin alındığı ve cut-off değerinin 3,69 alındığı görülmektedir. Malnutrisyon belirteci olarak serum albumin birçok mortalite ve morbidite çalışmanında değerlendirilmiştir. Ancak bu hasta grubunda sınıflandırma kriteri olarak hangi cut-off değeri kullanılacağı net değildir (48). CART algoritması ile bu konuda klinik olarak değerli bir sonuç elde edilmiştir. Yine BMI ve nPCR değişkenleri için de cut-off değerlerinin tespiti klinisyenlere yol gösterici olacaktır.

CART algoritması ve lojistik regresyon analiz sonuçlarının sınıflandırma performansı olarak doğruluk ölçütü ile değerlendirildiğinde sonuçlar sırasıyla 0,652 ve 0,760 bulunmuştur. Sonuçların benzer çıkmasının nedenlerinden biri veri setinin yapısı ile ilişkili olabilir. Veri seti bağımsız değişkenler tarafından iyi bir şekilde doğrusal olarak ayrılabilirse, hem CART hem de Lojistik Regresyon benzer performans gösterebilir. Lojistik Regresyon, doğrusal bir sınır oluşturmaya çalışırken,

CART de verileri doğrusal sınırlara yakın bölmelerle ayırabilir. Doğrusal ilişkilerin olduğu belirgin olduğu durumlarda, daha basit modeller yeterli olabilir. Yine az sayıda değişken farklı algoritmaların benzer sonuçlara ulaşmasına neden olabilir. CART, genellikle sadece sınıflandırma sonuçları verir ve olasılık tahminleri sunmaz. Bu nedenle, olasılıkların önemli olduğu durumlarda Lojistik Regresyon daha avantajlıdır. Huang ve ark. 506 glioma hastasıyla yaptığı çalışmada CART'ın da yer aldığı farklı makine öğrenmesi algoritmaları ve lojistik regresyon performansları değerlendirilmiş ve en yüksek eğri altıntında kalan alan lojistik regresyon için 0,838 olarak bulunmuştur. Parametre kestiriminde lojistik regresyonun daha iyi olduğu da savunulmuştur (49).

Bulanık mantık uygulamaları genellikle belirsiz ve kesin olmayan verilerle başa çıkmak için tasarlanmıştır. Bu tezde bulanık karar ağacı sınıflandırma performansı (doğruluk) diğer modellere göre düşük (0,456) tespit edildi. Yine ROC eğrisi ile yapılan karşılaştırmada da lojistik regresyona ve CART analizine göre eğri altında kalan alan düşük tespit edildi. Bu durum veri setinin küçük olması ile ilişkili olabilir. Ayrıca Tip-1 bulanık karar ağacında ROC ile eğri altında kalan alan hesaplanmasında üyelik dereceleri kullanılmıştır. Üyelik derecelerinin hesaplanmasında kullanılan parametreler (örneğin, küme sayısı c , bulanıklaştırma derecesi m) uygun şekilde ayarlanmadıysa, modelin performansı düşük çıkabilir. Ayrıca, Tip-1 bulanık karar ağacında kullanılan rastgele üyelik matrisi (`membership_matrix`) ve karar ağacının derinliği (`max_depth`) gibi parametreler veri seti küçük olduğunda modelin performansını etkileyebilir.

Sınıflandırma başarısını hesaplamak için kullanılan yöntemler de modelin performansını etkileyebilir. Doğruluk (`accuracy`), sınıflandırma başarısını ölçmek için kullanılan bir yöntemdir, ancak sınıflar arasında dengesizlik varsa yanıltıcı olabilir. Örneğin, veri setinde bir sınıf diğerine göre çok daha fazla örneğe sahipse, `accuracy` iki model için de benzer çıkabilir. Daha ayrıntılı bir değerlendirme için `precision`, `recall`, `F1-score` gibi yöntemler de incelenebilir. Ancak tıbbi literatürde yapılan yöntemlerin karşılaştırmalı çalışmalarında da doğruluk ve ROC ile eğri altında kalan alan performans değerlendirmede en yaygın kullanılan yöntemlerdir. Örneğin, dil kanseri olan 75 hastada boyun metastazı riskini analiz etmek için yapılan bir

alıřmada bulanık ıkarım, CART ve lojistik regresyon analizleri karřılařtırılmıř (50). Doğruluk performansı ile yapılan karřılařtırmada CART ve bulanık ıkarım biraz daha iyi deęerlendirilmiřken ROC ile eęri altında kalan alan ölçümleri üç performans için de benzer bulunmuřtur.



5. SONUÇ ve ÖNERİLER

Bu tezde aynı veri setinde CART ve lojistik regresyon analizinin doğruluk performanslarının benzer çıkması ve bulanık karar ağacı algoritması ile en düşük doğruluk performansı tespit edilmesinin nedenleri;

1- Veri setindeki ilişkiler gerçekten de büyük ölçüde doğrusal ise, CART gibi doğrusal olmayan ilişkilere de izin veren bir model benzer sonuçlar verebilir. Problem için karmaşık, doğrusal olmayan karar sınırlarına ihtiyaç olmayabilir. Yine benzer sonuçlar, veri setinin Lojistik Regresyon'un varsayımlarını (en azından yaklaşık olarak) karşıladığı anlamına gelebilir.

2- Her iki model de sonuçları en çok etkileyen aynı bağımsız değişkenleri belirleyebilir. Lojistik Regresyon katsayılarının büyüklüğü ve anlamlılığı yoluyla değişken önemini gösterirken, CART algoritmasının en üst düğümlerinde yer alan değişkenleri önemli olarak kabul eder.

3- Eğer CART analizi sonucunda çok derin olmayan, basit bir ağaç elde edildiyse, bu ağacın oluşturduğu karar sınırları Lojistik Regresyon'un doğrusal sınırlarına benzeyebilir.

4- Bulanık karar ağacı performansının CART ve Lojistik regresyona göre düşük olmasının farklı nedenleri olabilir. Bulanık karar ağaçları, belirsizlik ve örtüşen sınıflandırmalarla başa çıkmak için tasarlanmıştır. Eğer sınıflar net bir şekilde ayrılabiliriyorsa ve belirsizlik yoksa, bulanıklaştırma süreci gereksiz karmaşıklık yaratabilir ve performansı düşürebilir. Veya veri setindeki sayısal değişkenlerin bulanık kümelerle temsil edilme şekli (üyelik fonksiyonlarının seçimi, sayısı, parametreleri) yanlış tanımlandıysa algoritmanın performansı değişebilir.

5- Az sayıda bağımsız değişkene sahip veri setlerinde, farklı model türlerinin benzer örüntüleri yakalama olasılığı daha yüksektir. Bu durumda çalışmanın amacı önemlidir. Modelin temel amacı tahmin doğruluğu mu, yorumlanabilirlik mi, karar verme süreçlerini desteklemek mi belirlenmelidir. Yorumlanabilirlik ön plandaysa Lojistik Regresyon daha avantajlı olabilirken, doğrusal olmayan ilişkileri yakalama esnekliği önemliyse CART daha uygun olabilir. Yine veri seti için bulanıklaştırma gerekli mi karar verilmelidir.

7. KAYNAKLAR

1. Cramer, J. S. The origins of logistic regression. Tinbergen Institute Technical Report. 2002; 119: 167–178.
2. Silahtaroğlu, G. Veri Madenciliği (Kavram ve Algoritmaları), 2. Basım, İstanbul: Papatya Yayıncılık Eğitim; 2013.
3. Işıklı, Ş. Bulanık mantık ve bulanık teknolojiler. Ankara Üniversitesi Dil, Tarih ve Coğrafya Fakültesi Felsefe Bölümü Dergisi, 2008.
4. Alpar, R. Çok Değişkenli İstatistiksel Yöntemler. 5. Baskı, Ankara, Detay Yayıncılık; 2017.
5. Berkson, J. Application of the Logistic Function to Bio-Assay. Journal of the American Statistical Association. 1944; 39 (227): 357-365.
6. Cox, D.R. The regression analysis of binary sequences (with discussion). Journal of the Royal Statistical Society. 1958; 20 (2): 215–242.
7. Cornfield, J. Joint Dependence of the Risk of Coronary Heart Disease on serum Cholesterol and Systolic Blood Pressure: A Diskrimant Function Analysis. Federation Proceedings. 1962; 21: 58-61.
8. Bircan H. Lojistik Regresyon Analizi: Tıp Verileri Üzerine Bir Uygulama. Kocaeli Üniversitesi Sosyal Bilimler Enstitüsü Dergisi. 2004; 2: 185-208.
9. Hosmer, D. W., Lemeshow, S. Applied Logistic Regression. John Wiley & Sons; 1989.
10. O'Connell, A. A. Logistic Regression Models for Ordinal Response Variables. SAGE Publications; 2006.
11. Kleinbaum, D.G., Klein M. Logistic Regression. A Self- Learning Text. International Statistical Review. 2011; 79(2):296-296.
12. Hosmer, D.W., Lemeshow, S., Sturdivant, R. X. Applied Logistic Regression. 3rd Edition, John Wiley & Sons; 2013.
13. Albayrak, A. S. Uygulamalı Çok Değişkenli İstatistik Teknikleri. İstanbul: Asil Yayın Dağıtım; 2006.
14. Kapwata, T., Gebreslasie, M. T. Random forest variable selection in spatial malaria transmission modelling in Mpumalanga Province, South Africa. Geospatial Health. 2016; 11(3): 251-262.
15. Kabir, G., Balek, N.B.C., Tesfamariam, S. Sewer Structural Condition Prediction Integrating Bayesian Model Averaging with Logistic Regression. Journal of Performance of Constructed Facilities. 2018; 32(3): 1-1

16. Gürsoy, Ş. T. Uygulamalı Veri Madenciliği: Sektörel Analizler, 3. Basım, Ankara: Pegem Akademi; 2012.
17. Kim, M. Two-stage logistic regression model. Expert Systems with Applications. 2009; 36(3): 6727-6734.
18. Pehlivan, G. Chaid Analizi ve Bir Uygulama [Yüksek Lisans Tezi]. İstanbul: Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü; 2006.
19. Sayıcı, G. Karar Ağaçları, Bayes Ağları ve Etki Diyagramları Aracılığı ile Bilgi Keşfi ve Karar Verme [Yüksek Lisans Tezi]. İstanbul: İstanbul Üniversitesi Sosyal Bilimler Enstitüsü; 2013.
20. Kıran, Z. B. Lojistik Regresyon ve CART Analizi Teknikleriyle Sosyal Güvenlik Kurumu İlaç Provizyon Sistemi Verileri Üzerinde Bir Uygulama [Yüksek Lisans Tezi]. Ankara: Gazi Üniversitesi Fen Bilimleri Enstitüsü; 2010.
21. Saitoğlu S. Y. Sınıflama ve Regresyon Ağaçları [Doktora Tezi]. İstanbul: Marmara Üniversitesi Sosyal Bilimler Enstitüsü; 2015.
22. Güneri, İ.Ö., Aydın D. Grup Üyelerini Belirlemede İstatistiksel Sınıflandırma Yöntemleri: Karşılaştırmalı Bir Çalışma. Türkiye Klinikleri Journal of Biostatistics. 2017; 9(1): 45-67.
23. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J. Classification and Regression Trees. Wadsworth International Group, California. 1984; 1-15.
24. Deconinck, E., Hancock, T., Coomans, D., Massart, D.L., Heyden, Y.V. Classification of drugs in absorption classes using the classification and regression trees (CART) methodology. Journal of Pharmaceutical and Biomedical Analysis. 2005; 39: 91-103.
25. Akpınar H. Veri Tabanlarında bilgi keşfi ve Veri Madenciliği. İ.Ü. İşletme Fakültesi Dergisi. 2000; 29 (1): 1-22.
26. Aktaş, H. Çağman, N. Bulanık ve Yaklaşımlı Kümeler. Çankaya Üniversitesi Fen-Edebiyat Fakültesi, Journal of Arts and Sciences. 2005; 3: 13-25.
27. Zadeh, L.A. Fuzzy sets. Information and Control. 1965; 8: 338-353.
28. Yılmaz, H., Şahin, M.E. Bulanık Mantık Kavramına Genel Bir Bakış. Takvim-i Vekayi. 2023; 11(1): 94-129.
29. Zadeh, L.A. The concept of a linguistic variable and its application to approximate reasoning. Information Sciences. 1975; 8: 43-80.
30. Karnik, N.N., Mendel, J.M., Liang, Q. Type-2 Fuzzy Logic Systems. IEEE Trans. On Fuzzy Sytems. 1999; 7(6): 643-658.

31. Gündoğdu, K., Gündoğdu, E., Yücedağ, İ. Bulanık Mantık ile Akıllı Fırının Modellenmesi. Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 2016; 4(2): 574- 580.
32. Alışkan, İ., Ünsal, S. Farklı çıkarım yöntemlerine sahip bulanık mantık denetleyicileri kullanarak kalıcı mıknatıslı senkron motorun hız denetimi. Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi. 2016; 24(2): 185-191.
33. Abu-Halaweh, N. M., Harrison, R. W. Practical Fuzzy Decision Trees, Proceeding of IEEE Symposium on Computational Intelligence and Data Mining (CIDM09), 2009; 211-216.
34. Öztürk S. Bulanık Karar Ağaçları İçin Bir Deneysel Değerlendirme [Yüksek Lisans Tezi]. Adana: Çukurova Üniversitesi Fen Bilimleri Enstitüsü; 2015.
35. Quinlan, J.R. Induction of Decision Trees. Machine Learning. 1986; 1: 81-106.
36. Piegat, A. Modeling and Fuzzy Control. Warsaw, EXIT Academic Publishing House; 1999.
37. Janikow, C. Z. Fuzzy Decision Trees: Issues and Methods, IEEE Trans. On Man, Systems and Cybernetics. 1998; 28 (1): 1-14.
38. Rokach, L., Maimon, O. Data Mining With Decision Trees Theory and Applications. Singapore, World Scientific Publishing Co. Pte. Ltd.; 2008.
39. Han, J., Kamber, M. Data Mining Concepts and Techniques. Second Edition, San Francisco, Morgan Kaufmann Publishers; 2006.
40. Muller, R., Möckel, M. Logistic regression and CART in the analysis of multimarker studies. Clinica Chimica Acta. 2008; 394(1-2):1-6.
41. Jerhii, F., Anuraga, G., Fahmi, M.A. Risk factor analysis of hypertension with logistic regression and Classification and Regression Tree (CART). J. Phys.: Conf. Ser. 2019; 1217-1210.
42. Zimmerman RK, Balasubramani GK, Nowalk MP, Eng H, Urbanski L, Jackson ML, et al. Classification and Regression Tree (CART) analysis to predict influenza in primary care patients. BMC Infectious Disease. 2016;16(1):503.
43. Charizanos, G., Demirhan, H. & İçen, D. Binary classification with fuzzy logistic regression under class imbalance and complete separation in clinical studies. BMC Medical Research and Methodology. 2024; 24:145-153.
44. Hacettepe Üniversitesi Bilgi İşlem Daire Başkanlığı, <https://bidb.hacettepe.edu.tr/tr/Spss081118>
45. The R Project for The Statistical Computing, <https://www.r-project.org/orenge>

46. Orange Data Mining, <https://orangedatamining.com/>
47. Kalantar-Zadeh K., Kopple JD., Block G., Humphreys MH., A. Malnutrition-inflammation score is correlated with morbidity and mortality in maintenance hemodialysis patients. *American Journal of Kidney Disease*. 2001; 38:1251–1263.
48. Keller, U. Nutritional Laboratory Markers in Malnutrition. *Journal of Clinical Medicine*. 2019; 8(6):775.
49. Huang J, Ding S, Lin L, Zhong G, Yu Z, Luo Q, et al. Development and validation of a machine learning model based on laboratory parameters for preoperative prediction of Ki-67 expression in gliomas. *J Neurosurg*. 2025 Mar 28:1-13.
50. Schwarzer G, Nagata T, Mattern D, Schmelzeisen R, Schumacher M. Comparison of fuzzy inference, logistic regression, and classification trees (CART). Prediction of cervical lymph node metastasis in carcinoma of the tongue. *Methods Inf Med*. 2003;42(5):572-7

8. EKLER

EK-1: Turnitin Raporu Ekran Görüntüsü

The screenshot displays a Turnitin report for a document. The document title is "TIPTA KARAR VERMEDE BULANIK KARAR AĞAÇLARI, LOJİSTİK REGRESYON VE CART (CLASSIFICATION AND REGRESION TREES) ANALİZ YÖNTEMLERİNİN KARŞILAŞTIRILMASI" by Uzm. Dr. Didem TURGUT. The document is from Hacettepe University and is a Master's thesis in the Biostatistics Program. The similarity score is 17%. The report lists 10 sources contributing to the similarity:

Sıra	Kaynak	Oran
1	acikbilim.yok.gov.tr İnternet Kaynağı	%6
2	Hacettepe University' n... Öğrenci Yazılı Ödevi	%2
3	acikarsiv.ankara.edu.tr İnternet Kaynağı	%2
4	openaccess.hacettepe.... İnternet Kaynağı	%1
5	www.researchgate.net İnternet Kaynağı	%1
6	katalog.ticaret.edu.tr İnternet Kaynağı	%<1
7	Karadeniz Teknik Unive... Öğrenci Yazılı Ödevi	%<1
8	Bayram, Onur. "Çoklu b... Yayın	%<1
9	acikerisim.pau.edu.tr:8... İnternet Kaynağı	%<1
10	The Scientific & Techno... Öğrenci Yazılı Ödevi	%<1

The bottom of the screenshot shows the Turnitin interface with the word count "Kelime Sayısı: 7752" and the report type "Salt-Metin Raporu".

EK-2: Turnitin Dijital Makbuz



Dijital Makbuz

Bu makbuz ödevinizin Turnitin'e ulaştığını bildirmektedir. Gönderiminize dair bilgiler şöyledir:

Gönderinizin ilk sayfası aşağıda gönderilmektedir.

Gönderen: Didem Turgut
Ödev başlığı: DidemTurgut_Son
Gönderi Başlığı: DidemTurgut
Dosya adı: TEZ_EnSon.doc
Dosya boyutu: 7.6M
Sayfa sayısı: 82
Kelime sayısı: 15,295
Karakter sayısı: 113,280
Gönderim Tarihi: 11-May-2025 09:06ÖÖ (UTC+0300)
Gönderim Numarası: 2672526131

Y.Ü.
HACETTEPE ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ

TİPTE KARAR VERMEDE BULANIK KARAR AĞAÇLARI,
LOJİSTİK REGRESYON VE CART (CLASSIFICATION AND
REGRESSION TREES) ANALİZ YÖNTEMLERİNİN
KARŞILAŞTIRILMASI

Yus. Dr. Didem TURGUT

BİYOMETATİK PROGRAMI
YÜKSEK LİSANS TEZİ

ANKARA
2025

Copyright 2025 Turnitin. Tüm hakları saklıdır.

EK-3: Tezin Hazırlanması Aşamasında Kullanılan RStudio Bilgisayar Yazılımı Kodları

CART ALGORİTMASI

CART analizi, aşağıdaki adımlarda gerçekleştirilmiştir.

1. ****Modelin Tanımlanması****:

- `rpart` fonksiyonunu kullanarak bağımlı değişken (response variable) ile bağımsız değişkenler (predictor variables) arasında bir model kurar.

- Söz konusu fonksiyonda şu şekilde belirtilir:

```
```R
```

```
rpart(formula, data, method, control)
```

```
```
```

- ****formula****: Bağımlı ve bağımsız değişkenlerin formülü (ör. `y ~ x1 + x2`).

- ****data****: Kullanılacak veri seti.

- ****method****: Model türünü belirtir: `class` (sınıflama) ya da `anova` (regresyon).

- **"parms" parametreler listesini tanımlar**

- **"split = "gini"" bölünme kriteri olarak Gini indeksinin kullanılacağını belirtir**

- ****control****: Ağacın ne kadar derinleşeceğini veya budama işlemlerinin nasıl yapılacağını kontrol eden parametreler.

2. ****CART Mantığı****:

`rpart`, CART algoritmasını kullanarak aşağıdaki işlemleri gerçekleştirir:

- ****Veri Bölme (Splitting)****: Veri seti, bağımsız değişkenler üzerinden sürekli olarak bölünür. Bu bölme işlemi, ağaç oluşturma sürecinin temelidir. Her düğümde, Gini indeksi (sınıflama için) veya varyansın küçültülmesi (regresyon için) kriterlerine göre bölmeler belirlenir.

- ****Dallanma (Tree Building)****: Ağaç, belirlenen kurallar çerçevesinde ve belirli bir derinliğe kadar büyütülür.

- ****Terminal Düğümler****: Yaprak düğümler (terminal nodes) oluşturulana kadar işlem devam eder.

3. ****Model Sonuçları****:

- Uygulanan algoritmanın sonucunda ağaç modeli oluşturulur ve bu model `print`, `summary`, `plot` gibi komutlarla analiz edilebilir.

- Ağaçtaki her bir dal (branch), veri setindeki bir kurala karşılık gelir.

4. ****Örnek Kullanım****:

```
library(rpart)
```

```
# Örnek veri seti kullanarak CART modeli
```

```
fit <- rpart(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,  
            data = iris, method = 'class')
```

```
# Modelin özeti
```

```
summary(fit)
```

```
# Ağacın görselleştirilmesi
```

```
plot(fit)
```

```
text(fit, use.n = TRUE)
```

```
### Sonuç
```

`rpart` fonksiyonunun ****ağaç oluşturma kısmı**** (model fitting), CART analizini doğrudan gerçekleştirir. Bu süreç, veri bölme (splitting) ve dallanma gibi adımların tümünü içerir. Paket içerisinde kullanılan algoritma temel olarak CART mantığını uygular.

GEREKLİ KÜTÜPHANELERİ YÜKLEME

```
if (!requireNamespace("ggplot2", quietly = TRUE)) install.packages("ggplot2")
```

```
if (!requireNamespace("readr", quietly = TRUE)) install.packages("readr")
```

```
if (!requireNamespace("readxl", quietly = TRUE)) install.packages("readxl")
```

```
if (!requireNamespace("caret", quietly = TRUE)) install.packages("caret")
```

```
if (!requireNamespace("e1071", quietly = TRUE)) install.packages("e1071")
```

```

if (!requireNamespace("rpart", quietly = TRUE)) install.packages("rpart")
if (!requireNamespace("rpart.plot", quietly = TRUE)) install.packages("rpart.plot")
if (!requireNamespace("pROC", quietly = TRUE)) install.packages("pROC")
if (!requireNamespace("gridExtra", quietly = TRUE)) install.packages("gridExtra")
if (!requireNamespace("grid", quietly = TRUE)) install.packages("grid")
if (!requireNamespace("openxlsx", quietly = TRUE)) install.packages("openxlsx")
if (!requireNamespace("frbs", quietly = TRUE)) install.packages("frbs")

```

```

library(ggplot2)
library(readr)
library(readxl)
library(caret)
library(e1071)
library(rpart)
library(rpart.plot)
library(pROC)
library(gridExtra)
library(grid)
library(openxlsx)
library(frbs)

```

Veri Kümesini Hazırlama

```

file_path <- "C:\\Users\\gokal\\Desktop\\Didemtez.xlsx" # Dosya yolu
data <- read_excel(file_path)

```

Sütun isimlerini belirleme

```

column_names <- c("nPCR", "BMI", "CRPSon", "Alb", "GDF15", "Class")
colnames(data) <- column_names

```

Veriyi temizleme ve dönüştürme

```

data <- na.omit(data) # Eksik değerleri kaldır
data <- as.data.frame(sapply(data, as.numeric)) # Tüm sütunları sayısal hale dönüştür
data$Class <- factor(data$Class, levels = c(0, 1), labels = c("Yok", "Var")) # Sınıf etiketlerini düzenle

```

Tip-1 Bulanık Kümeleme ve Grafikler

```

tip1_fuzzy_clustering <- function(data, feature_name, c = 2, m = 2, epsilon = 1e-4, max_iter = 100) {
  feature_data <- data[[feature_name]]
  set.seed(123)
  U <- matrix(runif(length(feature_data) * c), ncol = c)
  U <- U / rowSums(U)

```

```

  for (iteration in 1:max_iter) {
    centers <- sapply(1:c, function(j) sum((U[, j]^m) * feature_data) / sum(U[, j]^m))
    distances <- outer(feature_data, centers, function(x, y) abs(x - y))
    U_new <- matrix(0, nrow = length(feature_data), ncol = c)
    for (i in 1:nrow(U_new)) {
      for (j in 1:c) {
        U_new[i, j] <- 1 / sum((distances[i, j] / distances[i, ])^2 / (m - 1)))
      }
    }
    if (all(abs(U_new - U) < epsilon)) break
    U <- U_new
  }
  return(list(U = U, centers = centers))
}

```

```

visualize_fuzzy_clustering <- function(data, feature_name, clustering_result) {
  feature_data <- data[[feature_name]]

```

```

centers <- clustering_result$centers
U <- clustering_result$U
cluster_data <- data.frame(
  Value = feature_data,
  Membership1 = U[, 1],
  Membership2 = U[, 2],
  Cluster = apply(U, 1, which.max)
)

ggplot(cluster_data, aes(x = Value)) +
  geom_point(aes(y = Membership1, color = "Cluster 1"), size = 2) +
  geom_point(aes(y = Membership2, color = "Cluster 2"), size = 2) +
  geom_vline(xintercept = centers, linetype = "dashed", color = "darkred") +
  labs(
    title = paste("Tip-1 Bulanık Kümeleştirme:", feature_name),
    x = feature_name,
    y = "Membership Degree"
  ) +
  theme_minimal() +
  scale_color_manual(values = c("Cluster 1" = "blue", "Cluster 2" = "green"))
}

# Tüm özellikler için kümeleme ve grafik çizimi
fuzzy_plots <- list()
for (feature_name in c("nPCR", "BMI", "CRPSon", "Alb", "GDF15")) {
  clustering_result <- tip1_fuzzy_clustering(data, feature_name)
  fuzzy_plots[[feature_name]] <- visualize_fuzzy_clustering(data, feature_name, clustering_result)
}

```

Grafikleri gösterme

```
do.call(grid.arrange, c(fuzzy_plots, ncol = 2))
```

Veriyi Eğitim ve Test Olarak Bölme ve Kaydetme

```

set.seed(123) # Kodun tekrarlanabilirliğini sağlayın
train_indices <- createDataPartition(data$Class, p = 0.7, list = FALSE) # Verinin %70'i eğitim olacak
train_data <- data[train_indices, ] # Eğitim verisi
test_data <- data # Test verisi: Tüm veri seti

```

```

# Excel çıktısı olarak kaydetme
write.xlsx(list(Egitim = train_data, Test = test_data), "C:\\Users\\gokal\\Desktop\\veri.xlsx")

```

LOJİSTİK REGRESYON MODELİ

```

# Eğitim verisi üzerinde lojistik regresyon modeli oluşturma
logistic_model <- glm(Class ~ ., data = train_data, family = binomial)

```

```

# Test ve eğitim veri setleri için tahmin yapma
train_predictions <- predict(logistic_model, newdata = train_data, type = "response")
test_predictions <- predict(logistic_model, newdata = test_data, type = "response")

```

```

# Tahmin sonuçlarını 0.5 eşik değerine göre sınıflandırma
train_predicted_class <- ifelse(train_predictions > 0.5, "Var", "Yok")
test_predicted_class <- ifelse(test_predictions > 0.5, "Var", "Yok")

```

ROC Eğrisi ve AUC

```

library(pROC)

# Eğitim verisi
train_roc <- roc(train_data$Class, train_predictions)

```

```

train_auc <- auc(train_roc)

# Test verisi
test_roc <- roc(test_data$Class, test_predictions)
test_auc <- auc(test_roc)

# ROC grafikleri
plot(train_roc, col = "blue", main = "ROC Eğrisi (Eğitim ve Test Veri)", lwd = 2, legacy.axes = TRUE)
lines(test_roc, col = "red", lwd = 2)
legend("bottomright", legend = c(paste("Eğitim AUC:", round(train_auc, 3)), paste("Test AUC:", round(test_auc, 3))), col = c("blue", "red"), lwd = 2)

```

Eğitim Veri Model İstatistiklerini Hesaplama

```

# -2 Log Likelihood
train_log_likelihood <- logLik(logistic_model)
train_minus_2_log_likelihood <- -2 * as.numeric(train_log_likelihood)

# Cox and Snell R Square
n_train <- nrow(train_data)
train_null_model <- glm(Class ~ 1, data = train_data, family = binomial)
train_null_log_likelihood <- logLik(train_null_model)
train_cox_snell_r2 <- 1 - exp((2 / n_train) * (as.numeric(train_null_log_likelihood) - as.numeric(train_log_likelihood)))

# Nagelkerke R Square
train_nagelkerke_r2 <- train_cox_snell_r2 / (1 - exp((2 / n_train) * as.numeric(train_null_log_likelihood)))

# İstatistikleri bir tabloya dönüştürme
train_model_statistics <- data.frame(
  Statistic = c("-2 Log Likelihood", "Cox and Snell R Square", "Nagelkerke R Square"),
  Value = c(train_minus_2_log_likelihood, train_cox_snell_r2, train_nagelkerke_r2)
)

```

Test Veri Model İstatistiklerini Hesaplama

```

# Test verisi için lojistik regresyon modeli oluşturma (model istatistikleri için)
test_model <- glm(Class ~ ., data = test_data, family = binomial)

# -2 Log Likelihood
test_log_likelihood <- logLik(test_model)
test_minus_2_log_likelihood <- -2 * as.numeric(test_log_likelihood)

# Cox and Snell R Square
n_test <- nrow(test_data)
test_null_model <- glm(Class ~ 1, data = test_data, family = binomial)
test_null_log_likelihood <- logLik(test_null_model)
test_cox_snell_r2 <- 1 - exp((2 / n_test) * (as.numeric(test_null_log_likelihood) - as.numeric(test_log_likelihood)))

# Nagelkerke R Square
test_nagelkerke_r2 <- test_cox_snell_r2 / (1 - exp((2 / n_test) * as.numeric(test_null_log_likelihood)))

# İstatistikleri bir tabloya dönüştürme
test_model_statistics <- data.frame(
  Statistic = c("-2 Log Likelihood", "Cox and Snell R Square", "Nagelkerke R Square"),
  Value = c(test_minus_2_log_likelihood, test_cox_snell_r2, test_nagelkerke_r2)
)

```

Sınıflama Başarı İstatistikleri

Confusion Matrix (Eğitim Verisi)

```
train_conf_matrix <- table(Predicted = train_predicted_class, Actual = train_data$Class)
```

Confusion Matrix (Test Verisi)

```
test_conf_matrix <- table(Predicted = test_predicted_class, Actual = test_data$Class)
```

Eğitim verisi - Doğruluk, Duyarlılık, Kesinlik Hesaplama

```
train_accuracy <- sum(diag(train_conf_matrix)) / sum(train_conf_matrix)
```

```
train_sensitivity <- train_conf_matrix["Var", "Var"] / sum(train_conf_matrix[, "Var"])
```

```
train_specificity <- train_conf_matrix["Yok", "Yok"] / sum(train_conf_matrix[, "Yok"])
```

Test verisi - Doğruluk, Duyarlılık, Kesinlik Hesaplama

```
test_accuracy <- sum(diag(test_conf_matrix)) / sum(test_conf_matrix)
```

```
test_sensitivity <- test_conf_matrix["Var", "Var"] / sum(test_conf_matrix[, "Var"])
```

```
test_specificity <- test_conf_matrix["Yok", "Yok"] / sum(test_conf_matrix[, "Yok"])
```

Başarı istatistiklerini bir tabloya dönüştürme - Eğitim verisi

```
train_classification_statistics <- data.frame(  
  Metric = c("Accuracy", "Sensitivity", "Specificity"),  
  Value = c(train_accuracy, train_sensitivity, train_specificity)  
)
```

Başarı istatistiklerini bir tabloya dönüştürme - Test verisi

```
test_classification_statistics <- data.frame(  
  Metric = c("Accuracy", "Sensitivity", "Specificity"),  
  Value = c(test_accuracy, test_sensitivity, test_specificity)  
)
```

Tablo Çizdirme

Model istatistik tablosu - Eğitim verisi

```
library(gridExtra)
```

```
train_statistics_table <- tableGrob(  
  train_model_statistics,  
  rows = NULL,  
  theme = ttheme_default(core = list(fg_params = list(cex = 0.8)))  
)
```

```
train_statistics_title <- textGrob("Eğitim Verisi Model İstatistikleri", gp = gpar(fontsize = 12, fontface = "bold"))
```

Model istatistik tablosu - Test verisi

```
test_statistics_table <- tableGrob(  
  test_model_statistics,  
  rows = NULL,  
  theme = ttheme_default(core = list(fg_params = list(cex = 0.8)))  
)
```

```
test_statistics_title <- textGrob("Test Verisi Model İstatistikleri", gp = gpar(fontsize = 12, fontface = "bold"))
```

Başarı istatistik tablosu - Eğitim verisi

```
train_classification_table <- tableGrob(  
  train_classification_statistics,  
  rows = NULL,  
  theme = ttheme_default(core = list(fg_params = list(cex = 0.8)))  
)
```

```
train_classification_title <- textGrob("Eğitim Verisi Sınıflama İstatistikleri", gp = gpar(fontsize = 12, fontface = "bold"))
```

```

# Başarı istatistik tablosu - Test verisi
test_classification_table <- tableGrob(
  test_classification_statistics,
  rows = NULL,
  theme = ttheme_default(core = list(fg_params = list(cex = 0.8)))
)
test_classification_title <- textGrob("Test Verisi Sınıflama İstatistikleri", gp = gpar(fontsize = 12,
fontface = "bold"))

# Grafik sekmesine çizdirme - Model İstatistikleri
grid.newpage()
grid.draw(train_statistics_title)
grid.newpage()
grid.draw(train_statistics_table)

grid.newpage()
grid.draw(test_statistics_title)
grid.newpage()
grid.draw(test_statistics_table)

# Grafik sekmesine çizdirme - Sınıflama İstatistikleri
grid.newpage()
grid.draw(train_classification_title)
grid.newpage()
grid.draw(train_classification_table)

grid.newpage()
grid.draw(test_classification_title)
grid.newpage()
grid.draw(test_classification_table)

ROC Grafikleri ve AUC
# ROC Grafiklerini çizme
plot(train_roc, col = "blue", lwd = 2, legacy.axes = TRUE, main = "ROC Eğrisi (Eğitim ve Test Veri)")
lines(test_roc, col = "red", lwd = 2)
legend("bottomright",
  legend = c(paste("Eğitim Verisi (AUC:", round(train_auc, 3), ")"),
    paste("Test Verisi (AUC:", round(test_auc, 3), ")"),
    col = c("blue", "red"), lwd = 2)

# AUC değerlerini grafik üzerine yazma
text(x = 0.5, y = 0.6, paste("Eğitim AUC:", round(train_auc, 3)), col = "blue", cex = 0.8, pos = 4)
text(x = 0.5, y = 0.5, paste("Test AUC:", round(test_auc, 3)), col = "red", cex = 0.8, pos = 4)

Omnibus Testi ve Model Özeti
# Eğitim Verisi Omnibus Testi
train_omnibus_title <- textGrob("Eğitim Verisi Omnibus Test Sonuçları", gp = gpar(fontsize = 12,
fontface = "bold"))
train_omnibus_table <- tableGrob(
  train_model_statistics,
  rows = NULL,
  theme = ttheme_default(core = list(fg_params = list(cex = 0.8)))
)

# Test Verisi Omnibus Testi

```

```

test_omnibus_title <- textGrob("Test Verisi Omnibus Test Sonuçları", gp = gpar(fontsize = 12,
fontface = "bold"))
test_omnibus_table <- tableGrob(
  test_model_statistics,
  rows = NULL,
  theme = ttheme_default(core = list(fg_params = list(cex = 0.8)))
)

# Model özet tablosu (Değişken katsayıları ve P değerleri) - Eğitim verisi
train_model_summary_title <- textGrob("Eğitim Verisi Model Özeti", gp = gpar(fontsize = 12,
fontface = "bold"))
train_model_summary_table <- tableGrob(
  summary(logistic_model)$coefficients,
  rows = NULL,
  theme = ttheme_default(core = list(fg_params = list(cex = 0.8)))
)

# Model özet tablosu (Değişken katsayıları ve P değerleri) - Test verisi
test_model_summary_title <- textGrob("Test Verisi Model Özeti", gp = gpar(fontsize = 12, fontface =
"bold"))
test_model_summary_table <- tableGrob(
  summary(test_model)$coefficients,
  rows = NULL,
  theme = ttheme_default(core = list(fg_params = list(cex = 0.8)))
)

# Grafik sekmesine çizdirme - Omnibus tabloları
grid.newpage()
grid.draw(train_omnibus_title)
grid.newpage()
grid.draw(train_omnibus_table)

grid.newpage()
grid.draw(test_omnibus_title)
grid.newpage()
grid.draw(test_omnibus_table)

# Grafik sekmesine çizdirme - Model özet tabloları
grid.newpage()
grid.draw(train_model_summary_title)
grid.newpage()
grid.draw(train_model_summary_table)

grid.newpage()
grid.draw(test_model_summary_title)
grid.newpage()
grid.draw(test_model_summary_table)

Sınıflama Sonuçları (Confusion Matrix)
library(caret)

# Confusion matrix (Eğitim ve Test veri seti için ayrı)
train_classification <- confusionMatrix(factor(train_predicted_class, levels = levels(train_data$Class)),
train_data$Class)
test_classification <- confusionMatrix(factor(test_predicted_class, levels = levels(test_data$Class)),
test_data$Class)

# Confusion matrix tablolarını çizdirme

```

```

train_confusion_title <- textGrob("Eğitim Verisi Confusion Matrix", gp = gpar(fontsize = 12, fontface = "bold"))
train_confusion_table <- tableGrob(
  as.data.frame.matrix(train_classification$table),
  rows = NULL,
  theme = ttheme_default(core = list(fg_params = list(cex = 0.8)))
)

test_confusion_title <- textGrob("Test Verisi Confusion Matrix", gp = gpar(fontsize = 12, fontface = "bold"))
test_confusion_table <- tableGrob(
  as.data.frame.matrix(test_classification$table),
  rows = NULL,
  theme = ttheme_default(core = list(fg_params = list(cex = 0.8)))
)

# Confusion matrix tablolarını çizdirme
grid.newpage()
grid.draw(train_confusion_title)
grid.newpage()
grid.draw(train_confusion_table)

grid.newpage()
grid.draw(test_confusion_title)
grid.newpage()
grid.draw(test_confusion_table)
Eğitim Verisi için CART (Gini Methodu)
library(rpart)
library(rpart.plot)
library(pROC)
library(gridExtra)

# Eğitim seti üzerinde CART modeli (Gini yöntemi)
cart_model_train <- rpart(Class ~ ., data = train_data, method = "class", parms = list(split = "gini"))

# Eğitim verisi için tahmin ve olasılıklar
train_cart_predictions <- predict(cart_model_train, newdata = train_data, type = "class")
train_cart_prob <- predict(cart_model_train, newdata = train_data, type = "prob")[, 2]

# Eğitim Confusion Matrix
cart_train_conf_matrix <- table(Predicted = train_cart_predictions, Actual = train_data$Class)

# Eğitim verisi için metrikler
cart_train_accuracy <- sum(diag(cart_train_conf_matrix)) / sum(cart_train_conf_matrix)
cart_train_sensitivity <- cart_train_conf_matrix["Var", "Var"] / sum(cart_train_conf_matrix[, "Var"])
cart_train_specificity <- cart_train_conf_matrix["Yok", "Yok"] / sum(cart_train_conf_matrix[, "Yok"])

# Eğitim ROC ve AUC
train_roc <- roc(train_data$Class, train_cart_prob, levels = c("Yok", "Var"))
train_auc <- auc(train_roc)

# Eğitim için ağaç diyagramı çizme
rpart.plot(cart_model_train, main = "CART Ağaç Diyagramı (Eğitim Verisi)", type = 1, extra = 102,
box.palette = "Blues", shadow.col = "gray", nn = TRUE)

Tüm Veri ile Yeni CART (Gini Methodu)
# CART modeli, tüm veri setiyle yeniden oluşturuluyor

```

```

cart_model_full <- rpart(Class ~ ., data = data, method = "class", parms = list(split = "gini"))

# Tüm veri seti için tahmin yapma
full_cart_predictions <- predict(cart_model_full, newdata = data, type = "class")

# Confusion Matrix (Tüm Veri için)
cart_full_conf_matrix <- table(Predicted = full_cart_predictions, Actual = data$Class)

# Doğruluk (Accuracy), Duyarlılık (Sensitivity), Kesinlik (Specificity)
cart_full_accuracy <- sum(diag(cart_full_conf_matrix)) / sum(cart_full_conf_matrix)
cart_full_sensitivity <- cart_full_conf_matrix["Var", "Var"] / sum(cart_full_conf_matrix[, "Var"])
cart_full_specificity <- cart_full_conf_matrix["Yok", "Yok"] / sum(cart_full_conf_matrix[, "Yok"])

# Tüm veriye dayalı sınıflama başarı istatistiklerini tabloya dönüştürme
cart_full_classification_stats <- data.frame(
  Metric = c("Accuracy", "Sensitivity", "Specificity"),
  Value = c(cart_full_accuracy, cart_full_sensitivity, cart_full_specificity)
)

# Yeni CART modeli ile detaylı ağaç diyagramı çizimi (Tüm Veri Seti Kullanılarak)
rpart.plot(cart_model_full, main = "CART Ağaç Diyagramı (Tüm Veri)", type = 1, extra = 102,
box.palette = "Purples", shadow.col = "gray", nn = TRUE)

Sınıf Çapraz Tablosu
# Prediction ve Actual sınıfların çapraz tablosu
print(cart_full_conf_matrix)
# ===== Tüm Veriye Dayalı Sonuç Tabloları =====
library(gridExtra)

# Tüm veriye dayalı sınıflama başarı tablosu
full_classification_title <- textGrob("Sınıflama Başarı İstatistikleri (Tüm Veri Seti)", gp =
gpar(fontsize = 12, fontface = "bold"))
full_classification_stats_table <- tableGrob(
  cart_full_classification_stats,
  rows = NULL,
  theme = ttheme_default(core = list(fg_params = list(cex = 0.8)))
)

# Tabloyu görselleştirme
grid.newpage()
grid.draw(full_classification_title)
grid.draw(full_classification_stats_table)

Eğitim ve Test için ROC Grafikleri
plot(train_roc, main = "ROC Eğrisi (Eğitim ve Tüm Veri İçin)", col = "blue", lwd = 2, legacy.axes =
TRUE)
lines(test_roc, col = "red", lwd = 2)
legend("bottomright", legend = c(paste("Eğitim AUC:", round(train_auc, 3)), paste("Test AUC:",
round(test_auc, 3))),
col = c("blue", "red"), lwd = 2)

# AUC değerlerini grafiklere yazma
text(x = 0.5, y = 0.6, labels = paste("Eğitim AUC:", round(train_auc, 3)), col = "blue", cex = 0.8, pos = 4)
text(x = 0.5, y = 0.5, labels = paste("Test AUC:", round(test_auc, 3)), col = "red", cex = 0.8, pos = 4)

Tüm Sonuçları Tablolar halinde Görselleştirme
# Eğitim sınıflama başarı istatistikleri
cart_classification_stats_train <- data.frame(
  Metric = c("Accuracy", "Sensitivity", "Specificity"),

```

```

    Value = c(cart_train_accuracy, cart_train_sensitivity, cart_train_specificity)
  )

```

```

cart_classification_stats_test <- data.frame(
  Metric = c("Accuracy", "Sensitivity", "Specificity"),
  Value = c(cart_full_accuracy, cart_full_sensitivity, cart_full_specificity)
)

```

Tip-1 Bulanık Karar Ağacı (Fuzzy Decision Tree)

```

library(caret)
library(pROC)
library(gridExtra)

```

Tip-1 bulanık karar ağacı oluşturma fonksiyonu

```

build_tip1_fuzzy_tree <- function(data, membership_matrix, class_labels, max_depth = 4,
min_samples = 5, current_depth = 0) {

```

```

  # Düğüm oluşturma

```

```

  node <- list(
    is_leaf = FALSE,
    entropy = calculate_fuzzy_entropy(membership_matrix, class_labels),
    name = paste("Node", current_depth),
    class = NULL,
    split_feature = NULL,
    split_point = NULL,
    left = NULL,
    right = NULL
  )

```

```

  # Durdurma kriterleri

```

```

  if (current_depth >= max_depth || nrow(data) <= min_samples || length(unique(class_labels)) == 1) {
    node$is_leaf <- TRUE
    node$class <- names(which.max(table(class_labels)))
    return(node)
  }

```

```

  # En iyi bölme noktasını bulma

```

```

  best_split <- find_best_split(data, membership_matrix, class_labels)

```

```

  # Eğer bölme bulunamazsa yaprak düğüm olarak işaretle

```

```

  if (is.null(best_split)) {
    node$is_leaf <- TRUE
    node$class <- names(which.max(table(class_labels)))
    return(node)
  }

```

```

  # Bölme bilgilerini kaydetme

```

```

  node$split_feature <- best_split$feature
  node$split_point <- best_split$point

```

```

  # Sol ve sağ alt kümeleri oluşturma

```

```

  left_indices <- which(data[, best_split$feature] <= best_split$point)
  right_indices <- which(data[, best_split$feature] > best_split$point)

```

```

  # Alt ağaçları oluşturma

```

```

  if (length(left_indices) > 0) {
    node$left <- build_tip1_fuzzy_tree(
      data[left_indices, , drop = FALSE],
      membership_matrix[left_indices, , drop = FALSE],
      class_labels[left_indices],

```

```

    max_depth,
    min_samples,
    current_depth + 1
  )
}

if (length(right_indices) > 0) {
  node$right <- build_tip1_fuzzy_tree(
    data[right_indices, , drop = FALSE],
    membership_matrix[right_indices, , drop = FALSE],
    class_labels[right_indices],
    max_depth,
    min_samples,
    current_depth + 1
  )
}

return(node)
}

```

BULANIK KARAR AĞACI FONKSİYONLARI

```

# Bulanık entropi hesaplama fonksiyonu
calculate_fuzzy_entropy <- function(membership_matrix, class_labels) {
  classes <- unique(class_labels)
  entropy <- 0

  for (c in classes) {
    class_indices <- which(class_labels == c)
    if (length(class_indices) > 0) {
      class_membership <- sum(membership_matrix[class_indices, 1])
      p <- class_membership / sum(membership_matrix[, 1])
      if (p > 0) {
        entropy <- entropy - p * log2(p)
      }
    }
  }

  return(entropy)
}

# En iyi bölme noktasını bulma fonksiyonu
find_best_split <- function(data, membership_matrix, class_labels) {
  best_gain <- -Inf
  best_feature <- NULL
  best_point <- NULL

  current_entropy <- calculate_fuzzy_entropy(membership_matrix, class_labels)

  for (feature in 1:ncol(data)) {
    feature_values <- unique(data[, feature])

    if (length(feature_values) <= 1) next

    for (i in 1:(length(feature_values) - 1)) {
      split_point <- (feature_values[i] + feature_values[i + 1]) / 2

      left_indices <- which(data[, feature] <= split_point)
      right_indices <- which(data[, feature] > split_point)
    }
  }
}

```

```

    if (length(left_indices) == 0 || length(right_indices) == 0) next

    left_entropy <- calculate_fuzzy_entropy(membership_matrix[left_indices, , drop = FALSE],
class_labels[left_indices])
    right_entropy <- calculate_fuzzy_entropy(membership_matrix[right_indices, , drop = FALSE],
class_labels[right_indices])

    p_left <- length(left_indices) / nrow(data)
    p_right <- length(right_indices) / nrow(data)

    information_gain <- current_entropy - (p_left * left_entropy + p_right * right_entropy)

    if (information_gain > best_gain) {
      best_gain <- information_gain
      best_feature <- feature
      best_point <- split_point
    }
  }
}

if (is.null(best_feature)) return(NULL)

return(list(feature = best_feature, point = best_point, gain = best_gain))
}

# Tip-1 bulanık karar ağacı oluşturma fonksiyonu
build_tip1_fuzzy_tree <- function(data, membership_matrix, class_labels, max_depth = 4,
min_samples = 5, current_depth = 0) {
  node <- list(
    is_leaf = FALSE,
    entropy = calculate_fuzzy_entropy(membership_matrix, class_labels),
    name = paste("Node", current_depth),
    class = NULL,
    split_feature = NULL,
    split_point = NULL,
    left = NULL,
    right = NULL
  )

  if (current_depth >= max_depth || nrow(data) <= min_samples || length(unique(class_labels)) == 1) {
    node$is_leaf <- TRUE
    node$class <- names(which.max(table(class_labels)))
    return(node)
  }

  best_split <- find_best_split(data, membership_matrix, class_labels)

  if (is.null(best_split)) {
    node$is_leaf <- TRUE
    node$class <- names(which.max(table(class_labels)))
    return(node)
  }

  node$split_feature <- best_split$feature
  node$split_point <- best_split$point

  left_indices <- which(data[, best_split$feature] <= best_split$point)

```

```

right_indices <- which(data[, best_split$feature] > best_split$point)

if (length(left_indices) > 0) {
  node$left <- build_tip1_fuzzy_tree(
    data[left_indices, , drop = FALSE],
    membership_matrix[left_indices, , drop = FALSE],
    class_labels[left_indices],
    max_depth,
    min_samples,
    current_depth + 1
  )
}

if (length(right_indices) > 0) {
  node$right <- build_tip1_fuzzy_tree(
    data[right_indices, , drop = FALSE],
    membership_matrix[right_indices, , drop = FALSE],
    class_labels[right_indices],
    max_depth,
    min_samples,
    current_depth + 1
  )
}

return(node)
}

```

EĞİTİM İÇİN KARAR AĞACI

```

# Üyelik matrisleri
membership_matrix_train <- matrix(runif(nrow(train_data) * 2), nrow = nrow(train_data), ncol = 2)

# Eğitim seti için karar ağacı
fuzzy_tree_train <- build_tip1_fuzzy_tree(train_data[, -ncol(train_data)], membership_matrix_train,
train_data$Class, max_depth = 4)

```

TAHMİN VE CONFUSION MATRIX

```

# Tahmin fonksiyonu
predict_fuzzy_tree <- function(tree, new_data) {
  predictions <- character(nrow(new_data))

  for (i in 1:nrow(new_data)) {
    current_node <- tree
    while (!current_node$is_leaf) {
      if (new_data[i, current_node$split_feature] <= current_node$split_point) {
        current_node <- current_node$left
      } else {
        current_node <- current_node$right
      }
    }
    predictions[i] <- as.character(current_node$class)
  }

  return(factor(predictions, levels = levels(train_data$Class)))
}

# Eğitim tahminleri
fuzzy_predictions_train <- predict_fuzzy_tree(fuzzy_tree_train, train_data[, -ncol(train_data)])

```

```

# Confusion matrix
fuzzy_confusion_train <- confusionMatrix(fuzzy_predictions_train, train_data$Class)

# Sonuçları yazdır
cat("\n--- Eğitim Verisi Confusion Matrix ---\n")
print(fuzzy_confusion_train$table)

TÜM VERİ İÇİN KARAR AĞACI
# Üyelik matrisleri
membership_matrix_all <- matrix(runif(nrow(data) * 2), nrow = nrow(data), ncol = 2)

# Tüm veri seti için karar ağacı
fuzzy_tree_all <- build_tip1_fuzzy_tree(data[, -ncol(data)], membership_matrix_all, data$Class,
max_depth = 4)

TAHMİN VE CONFUSION MATRIX
# Tahmin fonksiyonu
predict_fuzzy_tree <- function(tree, new_data) {
  predictions <- character(nrow(new_data))

  for (i in 1:nrow(new_data)) {
    current_node <- tree
    while (!current_node$is_leaf) {
      if (new_data[i, current_node$split_feature] <= current_node$split_point) {
        current_node <- current_node$left
      } else {
        current_node <- current_node$right
      }
    }
    predictions[i] <- as.character(current_node$class)
  }

  return(factor(predictions, levels = levels(data$Class)))
}

# Tüm veri tahminleri
fuzzy_predictions_all <- predict_fuzzy_tree(fuzzy_tree_all, data[, -ncol(data)])

BULANIK ROC EĞRİSİ VE AUC HESAPLAMA
library(pROC) # ROC eğrisi ve AUC hesaplama için

# Eğitim seti için bulanık ROC ve AUC hesaplama
fuzzy_membership_train <- membership_matrix_train[, 1] # Üyelik matrisinin birinci sütunu (örnek)
roc_train <- roc(train_data$Class, fuzzy_membership_train, levels = rev(levels(train_data$Class)))
auc_train <- auc(roc_train)

# Tüm veri seti için ROC ve AUC hesaplama
# Eğer tüm veri seti için üyelik matrisi mevcutsa:
membership_matrix_all <- matrix(runif(nrow(data) * 2), nrow = nrow(data), ncol = 2)
fuzzy_membership_all <- membership_matrix_all[, 1] # Üyelik matrisinin birinci sütunu
roc_all <- roc(data$Class, fuzzy_membership_all, levels = rev(levels(data$Class)))
auc_all <- auc(roc_all)

# ROC eğrisini çizme
plot(roc_train, col = "blue", lwd = 2, main = "Bulanık ROC Eğrisi (Eğitim ve Tüm Veri)", legacy.axes
= TRUE)
lines(roc_all, col = "red", lwd = 2)

```

```
# Eğitim verisinin AUC değerini grafiğin içine ekleme
text(x = 0.2, y = 0.2, labels = paste("Eğitim AUC:", round(auc_train, 3)), col = "blue", cex = 1.2, font = 2)

# Tüm veri setinin AUC değerini grafiğin içine ekleme
text(x = 0.2, y = 0.1, labels = paste("Tüm Veri AUC:", round(auc_all, 3)), col = "red", cex = 1.2, font = 2)

# Renklerin neyi temsil ettiğini açıklayan bir açıklama (legend)
legend("bottomright",
      legend = c("Eğitim Verisi (Mavi)", "Tüm Veri Seti (Kırmızı)"),
      col = c("blue", "red"),
      lwd = 2, bty = "n")
```

Gerekli Kütüphaneler

```
library(data.tree) # Karar ağacını hiyerarşik yapıya dönüştürmek için
library(DiagrammeR) # Karar ağacı görselleştirme için
```

```
# Feature adlarının belirlenmesi
feature_names <- c("nPCR", "BMI", "CRPSon", "Alb", "GDF15")
```

KARAR AĞACINI DÖNÜŞTÜRME FONKSİYONU

```
## 'data.tree' formatı oluşturuluyor
convert_tree_to_data_tree <- function(node, parent_name = "Root") {
  if (is.null(node)) return(NULL)

  # Eğer yaprak düğümse, sınıf bilgisini göster
  if (node$is_leaf) {
    node_info <- paste0(parent_name, "\nClass: ", node$class, "\nEntropy: ", round(node$entropy, 2))
    return(Node$new(node_info))
  }

  # Düğüm bilgileri (feature indekslerini anlamlı adlarla değiştiriyoruz)
  feature_name <- feature_names[node$split_feature] # Feature adı
  node_info <- paste0(parent_name, "\nFeature: ", feature_name,
    "\nPoint: ", round(node$split_point, 2),
    "\nEntropy: ", round(node$entropy, 2))

  # Düğümü oluştur
  tree_node <- Node$new(node_info)

  # Alt düğümleri ekle (Sol ve Sağ)
  if (!is.null(node$left)) {
    left_child <- convert_tree_to_data_tree(node$left, parent_name = "Left_child")
    tree_node$AddChildNode(left_child)
  }
  if (!is.null(node$right)) {
    right_child <- convert_tree_to_data_tree(node$right, parent_name = "Right_child")
    tree_node$AddChildNode(right_child)
  }

  return(tree_node) # Düğümü döndür
}
```

GRAFİK OLUŞTURMA FONKSİYONU

```
visualize_tree <- function(data_tree, title) {
  # Ağacı DiagrammeR grafiğine dönüştürerek render et
  graph <- ToDiagrammeRGraph(data_tree)
```

```
# Grafiği R Studio'nun Graphics penceresinde render et
DiagrammeR::render_graph(graph)
}
```

HEM EĞİTİM HEM DE TÜM VERİ İÇİN GRAFİK

```
# 1. Eğitim seti için karar ağacı
cat("\n--- Eğitim için karar ağacı oluşturuluyor ---\n")
fuzzy_tree_train_data_tree <- convert_tree_to_data_tree(fuzzy_tree_train)

# Eğitim seti için grafik göster
cat("Eğitim Karar Ağacı Görselleştiriliyor...\n")
visualize_tree(fuzzy_tree_train_data_tree, title = "Eğitim Karar Ağacı")

# 2. Tüm veri seti için karar ağacı
cat("\n--- Tüm veri seti için karar ağacı oluşturuluyor ---\n")
fuzzy_tree_all <- build_tip1_fuzzy_tree(data[, -ncol(data)], membership_matrix_all,
                                       data$Class, max_depth = 4)
fuzzy_tree_all_data_tree <- convert_tree_to_data_tree(fuzzy_tree_all)

# Tüm veri seti için grafik göster
cat("Tüm Veri Karar Ağacı Görselleştiriliyor...\n")
visualize_tree(fuzzy_tree_all_data_tree, title = "Tüm Veri Karar Ağacı")
```

GEREKLİ KÜTÜPHANELER

```
library(ggplot2)
library(readxl)
library(caret)
library(pROC)
library(rpart)
library(rpart.plot)
library(gridExtra)
library(grid)
```

ZAMAN ÖLÇÜMÜ

```
start_time <- Sys.time() # Kod başlangıç zamanı
```

VERİYİ YÜKLEME

```
file_path <- "C:\\Users\\gokal\\Desktop\\Didemtez.xlsx" # Dosya yolu
data <- read_excel(file_path)
colnames(data) <- c("nPCR", "BMI", "CRPSon", "Alb", "GDF15", "Class")
data <- na.omit(data)
data <- as.data.frame(sapply(data, as.numeric))
data$Class <- factor(data$Class, levels = c(0, 1), labels = c("Yok", "Var"))
```

```
# Eğitim ve test verisi (veri bölme işlemi yapılmadığı varsayılmıştır)
```

```
train_data <- train_data
test_data <- data
```

Lojistik Regresyon Modeli

```
logistic_model <- glm(Class ~ ., data = train_data, family = binomial)
train_predictions_logistic <- predict(logistic_model, newdata = train_data, type = "response")
test_predictions_logistic <- predict(logistic_model, newdata = test_data, type = "response")
```

Cart Modeli

```
cart_model <- rpart(Class ~ ., data = train_data, method = "class", parms = list(split = "gini"))
train_predictions_cart <- predict(cart_model, newdata = train_data, type = "prob")[, 2]
test_predictions_cart <- predict(cart_model, newdata = test_data, type = "prob")[, 2]
```

Tip-1 Bulanık Karar Ağacı

Tip-1 bulanık karar ağacı için üyelik matrisleri

```
membership_matrix_train <- matrix(runif(nrow(train_data) * 2), nrow = nrow(train_data), ncol = 2)
```

```
membership_matrix_test <- matrix(runif(nrow(test_data) * 2), nrow = nrow(test_data), ncol = 2)
```

```
train_predictions_fuzzy <- membership_matrix_train[, 1]
```

```
test_predictions_fuzzy <- membership_matrix_test[, 1]
```

ROC Eğrileri ve AUC

Eğitim verisi için ROC ve AUC

```
train_roc_logistic <- roc(train_data$Class, train_predictions_logistic)
```

```
train_roc_cart <- roc(train_data$Class, train_predictions_cart)
```

```
train_roc_fuzzy <- roc(train_data$Class, train_predictions_fuzzy)
```

Test (tüm veri) için ROC ve AUC

```
test_roc_logistic <- roc(test_data$Class, test_predictions_logistic)
```

```
test_roc_cart <- roc(test_data$Class, test_predictions_cart)
```

```
test_roc_fuzzy <- roc(test_data$Class, test_predictions_fuzzy)
```

ROC Grafiklerini Çizme

```
plot(train_roc_logistic, col = "blue", lwd = 2, legacy.axes = TRUE, main = "ROC Eğrileri (Eğitim ve Tüm Veri)")
```

```
lines(train_roc_cart, col = "green", lwd = 2)
```

```
lines(train_roc_fuzzy, col = "purple", lwd = 2)
```

```
lines(test_roc_logistic, col = "blue", lwd = 2, lty = 2)
```

```
lines(test_roc_cart, col = "green", lwd = 2, lty = 2)
```

```
lines(test_roc_fuzzy, col = "purple", lwd = 2, lty = 2)
```

```
legend("bottomright", legend = c(
```

```
  paste("Lojistik Regresyon (Eğitim AUC:", round(auc(train_roc_logistic), 3), ")"),
```

```
  paste("CART (Eğitim AUC:", round(auc(train_roc_cart), 3), ")"),
```

```
  paste("Bulanık Karar Ağacı (Eğitim AUC:", round(auc(train_roc_fuzzy), 3), ")"),
```

```
  paste("Lojistik Regresyon (Tüm Veri AUC:", round(auc(test_roc_logistic), 3), ")"),
```

```
  paste("CART (Tüm Veri AUC:", round(auc(test_roc_cart), 3), ")"),
```

```
  paste("Bulanık Karar Ağacı (Tüm Veri AUC:", round(auc(test_roc_fuzzy), 3), ")")
```

```
), col = c("blue", "green", "purple", "blue", "green", "purple"), lwd = 2, lty = c(1, 1, 2, 2, 2, 2))
```

CONFUSION MATRIX

Eğitim verisi için confusion matrix

```
train_predicted_class_logistic <- ifelse(train_predictions_logistic > 0.5, "Var", "Yok")
```

```
train_conf_matrix_logistic <- confusionMatrix(factor(train_predicted_class_logistic, levels =  
levels(train_data$Class)), train_data$Class)
```

```
train_predicted_class_cart <- ifelse(train_predictions_cart > 0.5, "Var", "Yok")
```

```
train_conf_matrix_cart <- confusionMatrix(factor(train_predicted_class_cart, levels =  
levels(train_data$Class)), train_data$Class)
```

```
train_predicted_class_fuzzy <- ifelse(train_predictions_fuzzy > 0.5, "Var", "Yok")
```

```
train_conf_matrix_fuzzy <- confusionMatrix(factor(train_predicted_class_fuzzy, levels =  
levels(train_data$Class)), train_data$Class)
```

Tüm veri için confusion matrix

```
test_predicted_class_logistic <- ifelse(test_predictions_logistic > 0.5, "Var", "Yok")
```

```
test_conf_matrix_logistic <- confusionMatrix(factor(test_predicted_class_logistic, levels =  
levels(test_data$Class)), test_data$Class)
```

```
test_predicted_class_cart <- ifelse(test_predictions_cart > 0.5, "Var", "Yok")
```

```
test_conf_matrix_cart <- confusionMatrix(factor(test_predicted_class_cart, levels =  
levels(test_data$Class)), test_data$Class)
```

```

test_predicted_class_fuzzy <- ifelse(test_predictions_fuzzy > 0.5, "Var", "Yok")
test_conf_matrix_fuzzy <- confusionMatrix(factor(test_predicted_class_fuzzy, levels =
levels(test_data$Class)), test_data$Class)

# Confusion Matrix'leri Görselleştirme
plot_confusion_matrix <- function(conf_matrix, title) {
  cm_df <- as.data.frame.matrix(conf_matrix$table)
  table_title <- textGrob(title, gp = gpar(fontsize = 15, fontface = "bold"))
  table_body <- tableGrob(cm_df, rows = rownames(cm_df), cols = colnames(cm_df))
  grid.arrange(table_title, table_body, ncol = 1, heights = c(0.2, 1))
}

grid.newpage()
plot_confusion_matrix(train_conf_matrix_logistic, "Confusion Matrix: Lojistik Regresyon (Eğitim)")
grid.newpage()
plot_confusion_matrix(test_conf_matrix_logistic, "Confusion Matrix: Lojistik Regresyon (Tüm
Veri)")
grid.newpage()
plot_confusion_matrix(train_conf_matrix_cart, "Confusion Matrix: CART (Eğitim)")
grid.newpage()
plot_confusion_matrix(test_conf_matrix_cart, "Confusion Matrix: CART (Tüm Veri)")
grid.newpage()
plot_confusion_matrix(train_conf_matrix_fuzzy, "Confusion Matrix: Bulanık Karar Ağacı (Eğitim)")
grid.newpage()
plot_confusion_matrix(test_conf_matrix_fuzzy, "Confusion Matrix: Bulanık Karar Ağacı (Tüm
Veri)")

```

Çapraz Tablolar ve Sınıflandırma Başarıları

```

# 1. Lojistik Regresyon Modeli
logistic_train_confusion <- table(Predicted = train_predicted_class, Actual = train_data$Class)
logistic_test_confusion <- table(Predicted = test_predicted_class, Actual = test_data$Class)

# Lojistik Regresyon Başarı Metrikleri
logistic_train_accuracy <- sum(diag(logistic_train_confusion)) / sum(logistic_train_confusion)
logistic_train_sensitivity <- logistic_train_confusion["Var", "Var"] / sum(logistic_train_confusion[,
"Var"])
logistic_train_specificity <- logistic_train_confusion["Yok", "Yok"] / sum(logistic_train_confusion[,
"Yok"])

logistic_test_accuracy <- sum(diag(logistic_test_confusion)) / sum(logistic_test_confusion)
logistic_test_sensitivity <- logistic_test_confusion["Var", "Var"] / sum(logistic_test_confusion[,
"Var"])
logistic_test_specificity <- logistic_test_confusion["Yok", "Yok"] / sum(logistic_test_confusion[,
"Yok"])

# 2. CART Modeli
cart_train_confusion <- table(Predicted = train_cart_predictions, Actual = train_data$Class)
cart_full_confusion <- table(Predicted = full_cart_predictions, Actual = data$Class)

# CART Başarı Metrikleri
cart_train_accuracy <- sum(diag(cart_train_confusion)) / sum(cart_train_confusion)
cart_train_sensitivity <- cart_train_confusion["Var", "Var"] / sum(cart_train_confusion[, "Var"])
cart_train_specificity <- cart_train_confusion["Yok", "Yok"] / sum(cart_train_confusion[, "Yok"])

cart_full_accuracy <- sum(diag(cart_full_confusion)) / sum(cart_full_confusion)
cart_full_sensitivity <- cart_full_confusion["Var", "Var"] / sum(cart_full_confusion[, "Var"])

```

```
cart_full_specificity <- cart_full_confusion["Yok", "Yok"] / sum(cart_full_confusion[, "Yok"])
```

```
# 3. Tip-1 Bulanık Karar Ağacı
```

```
fuzzy_train_confusion <- table(Predicted = fuzzy_predictions_train, Actual = train_data$Class)
```

```
# Tip-1 Bulanık Karar Ağacı Başarı Metrikleri
```

```
fuzzy_train_accuracy <- sum(diag(fuzzy_train_confusion)) / sum(fuzzy_train_confusion)
```

```
fuzzy_train_sensitivity <- fuzzy_train_confusion["Var", "Var"] / sum(fuzzy_train_confusion[, "Var"])
```

```
fuzzy_train_specificity <- fuzzy_train_confusion["Yok", "Yok"] / sum(fuzzy_train_confusion[, "Yok"])
```

Sonuçları Tablo Halinde Gösterme

```
# Tüm modellerin başarı metriklerini birleştirme
```

```
classification_results <- data.frame(
```

```
  Model = c("Logistic Regression", "Logistic Regression",  
            "CART", "CART",
```

```
            "Tip-1 Fuzzy Decision Tree"),
```

```
  Dataset = c("Eğitim", "Test",
```

```
              "Eğitim", "Tüm Veri",
```

```
              "Eğitim"),
```

```
  Accuracy = c(logistic_train_accuracy, logistic_test_accuracy,
```

```
               cart_train_accuracy, cart_full_accuracy,
```

```
               fuzzy_train_accuracy),
```

```
  Sensitivity = c(logistic_train_sensitivity, logistic_test_sensitivity,
```

```
                 cart_train_sensitivity, cart_full_sensitivity,
```

```
                 fuzzy_train_sensitivity),
```

```
  Specificity = c(logistic_train_specificity, logistic_test_specificity,
```

```
                 cart_train_specificity, cart_full_specificity,
```

```
                 fuzzy_train_specificity)
```

```
)
```

```
# Sonuçları yazdırma
```

```
print("Sınıflandırma Başarıları:")
```

```
print(classification_results)
```

Çapraz Tabloları Görselleştirme

```
# Çapraz tabloları görselleştirme fonksiyonu
```

```
plot_confusion_matrix <- function(confusion_matrix, title) {
```

```
  cm_df <- as.data.frame.matrix(confusion_matrix)
```

```
  table_title <- textGrob(title, gp = gpar(fontsize = 15, fontface = "bold"))
```

```
  table_body <- tableGrob(cm_df, rows = rownames(cm_df), cols = colnames(cm_df))
```

```
  grid.arrange(table_title, table_body, ncol = 1, heights = c(0.2, 1))
```

```
}
```

```
# Lojistik Regresyon Çapraz Tabloları
```

```
plot_confusion_matrix(logistic_train_confusion, "Lojistik Regresyon (Eğitim Verisi)")
```

```
plot_confusion_matrix(logistic_test_confusion, "Lojistik Regresyon (Test Verisi)")
```

```
# CART Çapraz Tabloları
```

```
plot_confusion_matrix(cart_train_confusion, "CART (Eğitim Verisi)")
```

```
plot_confusion_matrix(cart_full_confusion, "CART (Tüm Veri)")
```

```
# Tip-1 Bulanık Karar Ağacı Çapraz Tablosu
```

```
plot_confusion_matrix(fuzzy_train_confusion, "Tip-1 Bulanık Karar Ağacı (Eğitim Verisi)")
```

```
# 3. Tip-1 Bulanık Karar Ağacı (Tüm Veri)
```

```
fuzzy_full_confusion <- table(Predicted = fuzzy_predictions_all, Actual = data$Class)
```

```
# Tip-1 Bulanık Karar Ağacı Başarı Metrikleri (Tüm Veri)
fuzzy_full_accuracy <- sum(diag(fuzzy_full_confusion)) / sum(fuzzy_full_confusion)
fuzzy_full_sensitivity <- fuzzy_full_confusion["Var", "Var"] / sum(fuzzy_full_confusion[, "Var"])
fuzzy_full_specificity <- fuzzy_full_confusion["Yok", "Yok"] / sum(fuzzy_full_confusion[, "Yok"])
```

Sonuçları Tablo Halinde Gösterme

```
# Tüm modellerin başarı metriklerini birleştirme
classification_results <- data.frame(
  Model = c("Logistic Regression", "Logistic Regression",
            "CART", "CART",
            "Tip-1 Fuzzy Decision Tree", "Tip-1 Fuzzy Decision Tree"),
  Dataset = c("Eğitim", "Tüm Veri",
              "Eğitim", "Tüm Veri",
              "Eğitim", "Tüm Veri"),
  Accuracy = c(logistic_train_accuracy, logistic_test_accuracy,
               cart_train_accuracy, cart_full_accuracy,
               fuzzy_train_accuracy, fuzzy_full_accuracy),
  Sensitivity = c(logistic_train_sensitivity, logistic_test_sensitivity,
                  cart_train_sensitivity, cart_full_sensitivity,
                  fuzzy_train_sensitivity, fuzzy_full_sensitivity),
  Specificity = c(logistic_train_specificity, logistic_test_specificity,
                  cart_train_specificity, cart_full_specificity,
                  fuzzy_train_specificity, fuzzy_full_specificity)
)

# Sonuçları yazdırma
print("Sınıflandırma Başarıları:")
print(classification_results)
```

Düzeltilme İle Eklenen Parametreler

```
plot(log_roc, legacy.axes = FALSE, main = "Tüm ROC Eğrileri", col = "blue", lwd = 2)
plot(cart_roc, add = TRUE, col = "red", lwd = 2)
plot(tip1_roc, add = TRUE, col = "darkgreen", lwd = 2)
legend("bottomright", legend = c(
  paste("LR AUC =", round(log_auc,3)),
  paste("CART AUC =", round(cart_auc,3)),
  paste("Fuzzy AUC =", round(tip1_auc,3))
), lwd=2, col = c("blue", "red", "darkgreen"))
```

```
# Her adımın ardından kullanıcı kodu veya açıklama mesajı:
cat("\nKümelendirme analizleri başta, ROC eğrileri ayrı ayrı ve toplu olarak, karar ağacı-odds ratio ve
backward tablosu isimli görünür, Hosmer-Lemeshow ve confusion matrisleri de tam olarak
raporlanır.\n ")
```

```
# ===== Tip-1 Bulanık Kümeleme Fonksiyonu (Başa Eklendi!)
=====
tip1_fuzzy_clustering <- function(data, feature_name, c = 2, m = 2, epsilon = 1e-4, max_iter = 100) {
  feature_data <- data[[feature_name]]
  set.seed(123)
  U <- matrix(runif(length(feature_data) * c), ncol = c)
  U <- U / rowSums(U)
  for (iteration in 1:max_iter) {
    centers <- sapply(1:c, function(j) sum((U[, j]^m) * feature_data) / sum(U[, j]^m))
    distances <- outer(feature_data, centers, function(x, y) abs(x - y))
    U_new <- matrix(0, nrow = length(feature_data), ncol = c)
```

```

for (i in 1:nrow(U_new)) {
  for (j in 1:c) {
    den <- sum((distances[i, j] / distances[i, ])^2 / (m - 1)))
    U_new[i, j] <- ifelse(is.finite(den) & den != 0, 1 / den, 1)
  }
}
if (all(abs(U_new - U) < epsilon)) break
U <- U_new
}
return(list(U = U, centers = centers))
}

# ===== Gerekli Kütüphaneleri Yükleme =====
packages <- c("readxl", "caret", "pROC", "rpart", "rpart.plot", "gridExtra", "grid",
"ResourceSelection", "openxlsx", "MASS", "ggplot2")
for (p in packages) { if (!requireNamespace(p, quietly = TRUE)) install.packages(p,
dependencies=TRUE) }
library(readxl)
library(caret)
library(pROC)
library(rpart)
library(rpart.plot)
library(gridExtra)
library(grid)
library(ResourceSelection)
library(MASS)
library(ggplot2)

# ===== Veri Kümesini Hazırlama =====
file_path <- "D:\\Didemtez.xlsx" # Dosya yolunu güncelleyin
data <- read_excel(file_path)
column_names <- c("nPCR", "BMI", "CRPSon", "Alb", "GDF15", "Class")
colnames(data) <- column_names
data <- na.omit(data)
data$Class <- factor(as.numeric(data$Class), levels = c(0, 1), labels = c("Yok", "Var"))
data[, 1:5] <- apply(data[, 1:5], 2, as.numeric)

# ===== Tip-1 Bulanık Kümeleme ve Küme Dağılımları =====
fuzzy_results <- list()
fuzzy_memberships <- matrix(NA, nrow = nrow(data), ncol = 5)
colors <- c("tomato", "skyblue")
par(mfrow = c(2,3))
for (i in 1:5) {
  var <- column_names[i]
  res <- tip1_fuzzy_clustering(data, var)
  fuzzy_results[[var]] <- res
  if (!is.null(res)) {
    # En yüksek değerdeki merkez "Var'a daha yakın" olsun
    centers_order <- order(res$centers)
    # 1. merkez düşük, 2. merkez yüksek kabul edilecek (fizyolojik düzeyde ortalama daha yüksek olan
    hastalığa yakın olur genelde)
    cluster <- apply(res$U, 1, which.max)
    hist(data[[var]], breaks=15, col=colors[cluster], main=paste("Fuzzy Clusters of", var), xlab=var,
border="white")
    abline(v=res$centers, col=colors, lwd=2)
    legend("topright", legend=paste("Merkez", 1:2, ":", round(res$centers,2)), col=colors, lwd=2)
    # NOT: Yüksek merkez "Var", düşük merkez "Yok" olarak kabul

```

```

    fuzzy_memberships[,i] <- res$U[,centers_order[2]] # "Var" merkez
  }
}
par(mfrow=c(1,1))

# Her gözlemin "Var" üyelik puanının ortalamasını al (bunu skor olarak kullan)
fuzzy_score <- rowMeans(fuzzy_memberships, na.rm=TRUE) # na.rm TRUE güvenliği

# ===== Tip-1 Bulanık Karar Ağacı Fonksiyonları =====
calculate_entropy <- function(membership_matrix, class_labels) {
  class_probs <- colSums(membership_matrix) / sum(membership_matrix)
  entropy <- -sum(class_probs * log2(class_probs + 1e-9))
  return(entropy)
}
calculate_information_gain <- function(membership_matrix, feature_values, class_labels) {
  total_entropy <- calculate_entropy(membership_matrix, class_labels)
  split_point <- median(feature_values, na.rm = TRUE)
  left_indices <- feature_values <= split_point
  right_indices <- !left_indices
  left_entropy <- calculate_entropy(membership_matrix[left_indices, , drop = FALSE],
  class_labels[left_indices])
  right_entropy <- calculate_entropy(membership_matrix[right_indices, , drop = FALSE],
  class_labels[right_indices])
  left_weight <- sum(left_indices) / length(feature_values)
  right_weight <- sum(right_indices) / length(feature_values)
  information_gain <- total_entropy - (left_weight * left_entropy + right_weight * right_entropy)
  return(list(information_gain = information_gain, split_point = split_point))
}
build_tip1_fuzzy_tree <- function(data, membership_matrix, class_labels, depth = 0, max_depth = 3)
{
  if (depth >= max_depth || length(unique(class_labels)) == 1) {
    node <- list(name = paste("Leaf: Class =", unique(class_labels)), class = unique(class_labels),
    is_leaf = TRUE)
    return(node)
  }
  num_features <- ncol(data)
  max_information_gain <- -Inf
  best_feature <- NULL
  best_split_point <- NULL
  for (j in 1:num_features) {
    feature_values <- data[[j]]
    if (!is.numeric(feature_values)) next
    feature_values <- na.omit(feature_values)
    if (length(feature_values) == 0) next
    current_result <- calculate_information_gain(
      membership_matrix = membership_matrix,
      feature_values = feature_values,
      class_labels = class_labels
    )
    if (current_result$information_gain > max_information_gain) {
      max_information_gain <- current_result$information_gain
      best_feature <- colnames(data)[j]
      best_split_point <- current_result$split_point
    }
  }
  if (is.null(best_feature)) {
    node <- list(name = "Leaf: No valid split", class = unique(class_labels), is_leaf = TRUE)
    return(node)
  }

```

```

}
left_indices <- data[[best_feature]] <= best_split_point
right_indices <- !left_indices
node <- list(
  name = paste0(best_feature, " <= ", round(best_split_point, 2)),
  entropy = max_information_gain,
  split_feature = best_feature,
  split_point = best_split_point,
  is_leaf = FALSE,
  left = build_tip1_fuzzy_tree(
    data[left_indices, , drop = FALSE],
    membership_matrix[left_indices, , drop = FALSE],
    class_labels[left_indices],
    depth + 1,
    max_depth
  ),
  right = build_tip1_fuzzy_tree(
    data[right_indices, , drop = FALSE],
    membership_matrix[right_indices, , drop = FALSE],
    class_labels[right_indices],
    depth + 1,
    max_depth
  )
)
return(node)
}

# ===== 1. Lojistik Regresyon (Backward, HL, OR, ROC, AUC, CM)
=====
full_formula <- as.formula("Class ~ nPCR + BMI + CRPSon + Alb + GDF15")
log_full <- glm(full_formula, data = data, family = binomial)
backward_model <- stepAIC(log_full, direction = "backward", trace = FALSE)
summary_backward <- summary(backward_model)
coef_tab <- summary_backward$coefficients

# Satır isimli tablo
backward_table_df <- as.data.frame(round(coef_tab, 4))
backward_table_df$Variable <- rownames(coef_tab)
backward_table_df <- backward_table_df[,c(ncol(backward_table_df), 1:(ncol(backward_table_df)-
1))]]

OR <- exp(coef(backward_model))
OR_CI <- exp(confint(backward_model))
OR_table <- data.frame(
  Variable = names(OR),
  Odds_Ratio = round(OR, 3),
  CI_low = round(OR_CI[, 1], 3),
  CI_high = round(OR_CI[, 2], 3)
)
row.names(OR_table) <- NULL

# HL testi
hltest <- hoslem.test(as.numeric(data$Class == "Var"), fitted(backward_model), g = 10)
hl_results <- data.frame(
  Group = 1:length(hltest$observed[, 1]),
  Observed_No = hltest$observed[, 1],
  Observed_Yes = hltest$observed[, 2],
  Expected_No = round(hltest$expected[, 1], 1),

```

```

    Expected_Yes = round(hltest$expected[, 2], 1)
  )
  hl_stat <- data.frame(
    Test = "Hosmer-Lemeshow",
    Chi_square = hltest$statistic,
    df = hltest$parameter,
    p_value = hltest$p.value
  )

# ROC ve AUC (yatay eksen 1-specificity, legacy.axes=TRUE)
log_pred <- predict(backward_model, type = "response")
log_roc <- roc(data$Class, log_pred, levels = c("Yok", "Var"), direction = "<")
log_auc <- auc(log_roc)
log_auc_ci_vec <- as.numeric(ci.auc(log_roc))
log_class_pred <- ifelse(log_pred > 0.5, "Var", "Yok")
log_class_pred <- factor(log_class_pred, levels = c("Yok", "Var"))
log_cm <- confusionMatrix(log_class_pred, data$Class)
log_acc <- log_cm$overall["Accuracy"]
log_acc_ci <- log_cm$overall[c("AccuracyLower", "AccuracyUpper")]

plot(log_roc, legacy.axes = TRUE, main = "ROC - Lojistik Regresyon", col = "blue", lwd = 2)
legend("bottomright", paste("AUC =", round(log_auc,3)), lwd=2, col="blue")

# ===== 2. CART MODELİ =====
cart_model <- rpart(Class ~ ., data = data, method = "class", control = rpart.control(minsplit = 10,
minbucket = 5, cp = 0.01, maxdepth = 5))
cart_pred <- predict(cart_model, data, type = "class")
cart_cm <- confusionMatrix(cart_pred, data$Class)
cart_prob <- predict(cart_model, data)[, "Var"]
cart_roc <- roc(data$Class, cart_prob, levels = c("Yok", "Var"), direction = "<")
cart_auc <- auc(cart_roc)
cart_auc_ci_vec <- as.numeric(ci.auc(cart_roc))
cart_acc <- cart_cm$overall["Accuracy"]
cart_acc_ci <- cart_cm$overall[c("AccuracyLower", "AccuracyUpper")]

plot(cart_roc, legacy.axes = TRUE, main = "ROC - CART", col = "red", lwd = 2)
legend("bottomright", paste("AUC =", round(cart_auc,3)), lwd=2, col="red")
rpart.plot(cart_model, main="CART Ağacı")

# ===== 3. Tip-1 Bulanık Karar Ağacı (VERİ TABANLI) =====
# -- ŞİMDİ: Veri tabanlı membership_matrix ile ağacı kur --
membership_matrix <- cbind(1 - fuzzy_score, fuzzy_score)
tip1_tree <- build_tip1_fuzzy_tree(data[,1:5], membership_matrix, data$Class, max_depth = 3)

# SINIFLANDIRMA için basit kural: fuzzy_score > 0.5 ise "Var", aksi "Yok"
tip1_pred_label <- ifelse(fuzzy_score > 0.5, "Var", "Yok")
tip1_pred_label <- factor(tip1_pred_label, levels = c("Yok", "Var"))
tip1_cm <- confusionMatrix(tip1_pred_label, data$Class)
tip1_acc <- tip1_cm$overall["Accuracy"]
tip1_acc_ci <- tip1_cm$overall[c("AccuracyLower", "AccuracyUpper")]

# ROC ("skor" olarak veri tabanlı fuzzy_score kullan)
tip1_roc <- roc(data$Class, fuzzy_score, levels = c("Yok", "Var"), direction = "<")
tip1_auc <- auc(tip1_roc)
tip1_auc_ci_vec <- as.numeric(ci.auc(tip1_roc))
plot(tip1_roc, legacy.axes = TRUE, main = "ROC - Tip-1 Bulanık Karar Ağacı (Veri Tabanlı)", col =
"darkgreen", lwd = 2)

```

```

legend("bottomright", paste("AUC =", round(tip1_auc,3)), lwd=2, col="darkgreen")

# ===== Confusion Matrisleri Grafik Sekmesine Taşıma
=====
cm_titles <- list(
  textGrob("Lojistik Regresyon Confusion Matrix", gp=gpar(fontface="bold",fontsize=11)),
  textGrob("CART Confusion Matrix", gp=gpar(fontface="bold",fontsize=11)),
  textGrob("Tip-1 Bulanık Karar Ağacı Confusion Matrix (Veri tabanlı)",
gp=gpar(fontface="bold",fontsize=11))
)
cm_grobs <- list(
  tableGrob(log_cm$table, rows = NULL, theme=ttheme_minimal()),
  tableGrob(cart_cm$table, rows = NULL, theme=ttheme_minimal()),
  tableGrob(tip1_cm$table, rows = NULL, theme=ttheme_minimal())
)
cms <- arrangeGrob(cm_titles[[1]], cm_grobs[[1]],
  cm_titles[[2]], cm_grobs[[2]],
  cm_titles[[3]], cm_grobs[[3]],
  ncol = 1)
grid.newpage(); grid.draw(cms)

# ===== Odds Ratio Tablosunu Grafik Sekmesine Taşıma
=====
or_title <- textGrob("Lojistik Regresyon Odds Ratio Tablosu", gp = gpar(fontface="bold",
fontsize=12))
or_grob <- tableGrob(OR_table, rows = NULL)
grid.newpage(); grid.draw(arrangeGrob(or_title, or_grob, ncol=1))

# ===== Model Doğruluğu ve AUC Güven Aralıkları Tablosu
=====
metrics_table <- data.frame(
  Model = c("Lojistik Regresyon", "CART", "Tip-1 Bulanık Karar Ağacı"),
  Accuracy = round(c(log_acc, cart_acc, tip1_acc),3),
  Acc_CI_Lower = round(c(log_acc_ci[[1]], cart_acc_ci[[1]], tip1_acc_ci[[1]]),3),
  Acc_CI_Upper = round(c(log_acc_ci[[2]], cart_acc_ci[[2]], tip1_acc_ci[[2]]),3),
  AUC = round(c(log_auc, cart_auc, tip1_auc),3),
  AUC_CI_Lower = round(c(log_auc_ci_vec[1], cart_auc_ci_vec[1], tip1_auc_ci_vec[1]),3),
  AUC_CI_Upper = round(c(log_auc_ci_vec[3], cart_auc_ci_vec[3], tip1_auc_ci_vec[3]),3)
)
acc_title <- textGrob("Doğruluk ve AUC için Güven Aralıkları", gp = gpar(fontface="bold",
fontsize=12))
acc_grob <- tableGrob(metrics_table, rows = NULL)
grid.newpage(); grid.draw(arrangeGrob(acc_title, acc_grob, ncol=1))

# ===== Backward Sonuçları Tablosu (sattır isimli) =====
backward_summary_title <- textGrob("Backward Eleme Sonrası Lojistik Model", gp =
gpar(fontface="bold", fontsize=11))
backward_table <- tableGrob(backward_table_df, rows= backward_table_df$Variable,
theme=ttheme_minimal())
grid.newpage(); grid.draw(arrangeGrob(backward_summary_title, backward_table, ncol=1))

# ===== Hosmer-Lemeshow Tablosu (grafik sekmesine)
=====
hl_title <- textGrob("Hosmer-Lemeshow Testi (Gözlenen/Beklenen Tablo)", gp =
gpar(fontface="bold", fontsize=12))
hl_grob <- tableGrob(hl_results, rows = NULL)
grid.newpage(); grid.draw(arrangeGrob(hl_title, hl_grob, ncol=1))

```

```

hlstat_title <- textGrob("Hosmer-Lemeshow Test İstatistikleri", gp = gpar(fontface="bold",
fontsize=12))
hlstat_grob <- tableGrob(hl_stat, rows = NULL)
grid.newpage(); grid.draw(arrangeGrob(hlstat_title, hlstat_grob, ncol=1))

plot(1 - log_roc$specificities, log_roc$sensitivities, type = "l", lwd = 2,
     col = "blue", main = "Tüm ROC Eğrileri",
     xlab = "1 - Specificity", ylab = "Sensitivity",
     xlim = c(0, 1), ylim = c(0, 1))
lines(1 - cart_roc$specificities, cart_roc$sensitivities, col = "red", lwd = 2)
lines(1 - tip1_roc$specificities, tip1_roc$sensitivities, col = "darkgreen", lwd = 2)
legend("bottomright",
      legend = c("Lojistik Regresyon", "CART", "Tip-1 Bulanık Karar Ağacı"),
      lwd = 2, col = c("blue", "r

```



9. ÖZGEÇMİŞ

