T.C.
GAZİOSMANPAŞA ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ

LOJİSTİK REGRESYON ANALİZİ (LRA), YAPAY SİNİR AĞLARI (YSA) ve SINIFLANDIRMA ve REGRESYON AĞAÇLARI (C&RT) YÖNTEMLERİNİN KARŞILAŞTIRILMASI ve TIP ALANINDA BİR UYGULAMA

Hazırlayan
Yunus Emre Kuyucu

Biyoistatistik Anabilim Dalı
Yüksek Lisans Tezi

Danışman
Yrd. Doç. Dr. Ünal Erkorkmaz

Tokat-2012

T.C.
GAZİOSMANPAŞA ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ

LOJİSTİK REGRESYON ANALİZİ (LRA), YAPAY SİNİR AĞLARI (YSA) ve SINIFLANDIRMA ve REGRESYON AĞAÇLARI (C&RT) YÖNTEMLERİNİN KARŞILAŞTIRILMASI ve TIP ALANINDA BİR UYGULAMA

Hazırlayan
Yunus Emre Kuyucu

Biyoistatistik Anabilim Dalı
Yüksek Lisans Tezi

Danışman
Yrd. Doç. Dr. Ünal Erkorkmaz

Tokat-2012

LOJİSTİK REGRESYON ANALİZİ (LRA), YAPAY SİNİR AĞLARI (YSA) ve SINIFLANDIRMA ve REGRESYON AĞAÇLARI (C&RT) YÖNTEMLERİNİN KARŞILAŞTIRILMASI ve TIP ALANINDA BİR UYGULAMA

Tezin Kabul Ediliş Tarihi: 04/01/2012

Jüri Üyeleri (Unvanı, Adı Soyadı) İmzası

Prof. Dr. Hafize SEZER
(Başkan)

Yrd. Doç. Dr. Ünal ERKORKMAZ
(Üye, Danışman Öğretim Üyesi)

Yrd. Doç. Dr. İlker ETİKAN
(Üye)

Bu tez, Gaziosmanpaşa Üniversitesi Sağlık Bilimleri Enstitüsü Yönetim Kurulu'nun ……/……/……. tarih ve ……. sayılı oturumunda belirtilen jüri tarafından kabul edilmiştir.

Doç. Dr. Hüseyin ÖZYURT Mühür

Enstitü Müdürü İmza

T.C.
GAZİOSMANPAŞA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜ'NE

Bu belge ile bu tezdeki bütün bilgilerin akademik kurallara ve etik ilkelere uygun olarak toplanıp sunulduğunu, bu kural ve ilkelerin gereği olarak, çalışmada bana ait olmayan tüm veri, düşünce ve sonuçlara atıf yaptığımı ve kaynağını gösterdiğimi beyan ederim.

04/01/2012

Tezi Hazırlayan Öğrencinin
Adı ve Soyadı

Yunus Emre KUYUCU

İmzası

TEŞEKKÜR

Bu tezin hazırlanması aşamasında yardımlarını esirgemeyen, bana çalışmamın her aşamasında deneyim ve bilgileri ile yol gösteren danışman hocam Yrd. Doç. Dr. Ünal ERKORKMAZ' a, hocam Yrd. Doç. Dr. İlker ETİKAN' a, sevgisi ve desteği ile hep yanımda ve kalbimde olan eşim Nagehan KUYUCU' ya, biricik yavrumuz Elif Naz KUYUCU' ya ve desteklerinden dolayı tüm aileme teşekkür ediyorum.

ÖZET

Lojistik Regresyon Analizi (LRA), Yapay Sinir Ağları (YSA) ve Sınıflandırma ve Regresyon Ağaçları (C&RT) Yöntemlerinin Karşılaştırılması ve Tıp Alanında Bir Uygulama

Lojistik Regresyon, bağımlı değişkenin kategorik ve bağımsız değişkenlerin karışık ölçekli olması durumunda belirli bir dağılım varsayımına bağımlı kalmaksızın bağımlı değişken ile bağımsız değişkenler arasındaki neden-sonuç ilişkisinin belirlenmesinde kullanılan bir yöntemdir.

Yapay Sinir Ağları, insan beynindeki sinir ağları gibi çalışarak son derece karmaşık yapıya sahip problemlerin çözümünü sağlar. Kullandığı geriye yayılma algoritmasını ile ağ hatasını minimize ederek birimlerin en az hata ile sınıflarına atanması için ağın ağırlıklarını hesaplar.

Sınıflandırma ve Regresyon Ağaçları (C&RT); ikili karar ağacı oluşturan bir yöntemdir. Ağaçtaki her bir noda, her bir bağımsız değişken için gelişim skoruna dayalı olarak en iyi kesim noktası ya da en iyi kategori grupları oluşturulur.

Bu tez çalışmasında, Lojistik Regresyon Analizi (LRA), Yapay Sinir Ağları (YSA) ve Sınıflandırma ve Regresyon Ağaçları (C&RT) Yöntemlerinin sınıflandırma özelliklerinin karşılaştırılması yapılmıştır. Bu karşılaştırma için prostat kanseri yönünden değerlendirilmesi yapılan hastalara ait veriler kullanılmış, üç yönteme göre elde edilen sonuçlar başarı yönünden değerlendirilmiştir.

Anahtar kelimeler: Lojistik Regresyon Analizi (LRA), Yapay Sinir Ağları (YSA), Sınıflandırma Ve Regresyon Ağaçları (C&RT), Prostat Kanseri.

ABSTRACT

Comparison of Logistic Regression Analysis (LRA), Artificial Neural Networks (ANN) and Classfication and Regression Trees (C&RT) Methods And An Aplication In Medicine

Logistic regression, categorical dependent variable and independent variables in the case of mixed-scale without being dependent on the assumption that a certain distribution of cause-effect relationship between the dependent variable and independent variables used in the determination of a method.

Artificial Neural Networks, such as neural networks working in the human brain is extremely complex structure provides the solution of problems. Error back propagation algorithm uses the network by minimizing the weight of the units of the network accounts for the appointment of classes with fewer errors.

Classification and Regression Trees (C & RT) is a method of forming a binary decision tree. Each node in the tree, each independent variable on the basis of the best cut-off point score for the development or the best category groups are created.

In this thesis, Logistic Regression Analysis (LRA), Artificial Neural Networks (ANN) and Classification and Regression Trees (C & RT) characteristics of the classification methods were compared. This comparison of the data used in patients undergoing evaluation for prostate cancer in terms of the three methods to the results obtained were evaluated in terms of success.

Key words: Logistic Regression Analysis (LRA), Artificial Neural Networks (ANN), Classfication and Regression Trees (C&RT), Prostate Cancer.

İÇİNDEKİLER

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

KISATMALAR ve SİMGELER

C&RT	Sınıflandırma ve Regresyon Ağacı
YSA	Yapay Sinir Ağları
LRA	Lojistik Regresyon Analizi
ROC	Receiver Operating Characteristic Curve (Alıcı İşlem Karakteristikleri Eğrisi)
AUC	ROC eğrisi altında kalan alan (Area under the ROC Curve)
VM	Veri Madenciliği
CHAID	(Chi-Squared Automatic Interaction Detector: Otomatik Ki-Kare Etkileşim Belirleme),
MARS	(Multivariate Adaptive Regression Splines: Çok Değişkenli Uyumlu Regresyon Uzanımlan),
QUEST	(Quick, Unbiased, Efficient Statistical Tree: Hızlı, Yansız, Etkin İstatistiksel Ağaç),
SLIQ	(Supervised Learning in Quest),
SPRINT	(Scalable Parallelizable Induction of Decision Trees)
ID3	Iterative Dichotomiser 3

1. GİRİŞ ve AMAÇ

Günümüzde teknolojinin hızlı gelişimi ile tüm sektörlerden elde edilen veri miktarı devasa boyutlara ulaşmış ve verilerin toplanması, saklanması, işlenmesi gibi problemler ortaya çıkmıştır. Öyle ki bugün dünyadaki bilgi miktarının her yirmi ayda bir ikiye katlandığı kabul edilmektedir. Bu problemlere, gelişen bilgisayar teknolojileri, kapasitesi ve işlem gücü artan donanımlar ve yazılımlar ile birlikte çözüm sunmaya çalışmıştır. Bu çözümlerinde en önemlisi veri tabanları alanındaki ilerlemelerdir. Ancak veritabanı sistemlerinin artan kullanımı ve hacimlerindeki olağanüstü artış, işletmeleri toplanan verilerden nasıl faydalanılabileceği problemi ile karşı karşıya bırakmıştır. Veri miktarı arttıkça bu verilerin insanlarca anlaşılması zorlaşmış; veriler içerisinde saklı kalmış yararlı bilginin elde edilmesi ve Karar Destek Sistemi çerçevesinde kullanımı, herhangi bir araç kullanmaksızın olanaksız hale gelmiştir. Geleneksel sorgu veya raporlama araçlarının veri yığınları karşısında yetersiz kalması, veri madenciliği ve veri madenciliği altında yapılan sınıflandırmalar gibi yeni arayışlara neden olmaktadır [1, 2, 3].

Sınıflandırma uygulamalarında kullanılan pek çok model ve bu modellere ait farklı algoritmalar vardır. Bu algoritmalardan hangisinin daha efektif sonuçlar ürettiği, hangi algoritmanın hangi alanda daha başarılı olduğu sorusuna verilen cevaplar uygulamaların başarımını arttıracak ve yapılan işin verimini arttıracaktır. Bu sebeple algoritmaların karşılaştırılarak değerlendirilmesi büyük önem arz etmektedir. Çok sayıda algoritma olması, her algoritmanın kendi içinde farklı parametrelerle çalışması, her algoritmanın birden çok versiyonunun bulunması, farklı algoritmaların farklı amaca yönelik olması, kullanılan veri kaynağının farklı olması, algoritmaların farklı veri

tiplerini desteklemesi ve veri üzerinde yapılan önişlemlerin uygulayıcıya bağlı olması gibi sebeplerle farklı sonuçlar elde edilmiştir [1, 2, 4, 12, 13, 15].

Bağımlı değişkenlerin kategorik yapıda olduğu veri setlerinin analizinde bağımsız değişkenlerin bağımlı değişken üzerindeki etkilerini değerlendirmek ve birimlerin bağımlı değişkenin kategorilerine göre en az hata ile sınıflandırılması için kullanılan Sınıflandırma ve Regresyon Ağaçları (C&RT); Yapay Sinir Ağları (YSA) ve Lojistik Regresyon Analizi (LRA) yöntemlerinin karşılaştırılması yapılmış ve birbirlerine göre avantaj ve dezavantajları incelenmiştir. Yöntemlerin karşılaştırılması için gerçek bir veri seti ve karşılaştırma için açık kaynak koda sahip bir istatistik programı olan Weka (Waikato Environment for Knowledge Analysis Ver.3,6,5) programı kullanılmıştır.

Sınıflandırma ve Regresyon Ağaçları (C&RT); ikili karar ağacı oluşturan bir yöntemdir. Ağaçtaki her bir noda, her bir bağımsız değişken için gelişim skoruna dayalı olarak en iyi kesim noktası ya da en iyi kategori grupları oluşturulur. Sınıflandırma ve Regresyon Ağaçları'nda amaç bağımlı değişken ile ilgili verilerin mümkün olduğunca homojen alt setlerinin meydana getirilmesidir. C&RT, kategorik yapıdaki bağımlı değişkenler için genellikle Gini indeksi kullanarak en iyi kestirici değişken seçerek ağaç yapısını oluşturur [3, 4, 14].

Yapay Sinir Ağları insan beynindeki sinir ağları gibi çalışarak elle çözüm olanağı vermeyen son derece karmaşık yapıya sahip problemlerin çözümünü sağlayan ve değişken yapısı konusunda herhangi bir kısıtlama getirmeksizin değişkenler arası ilişkiyi ortaya koyan çok esnek bir yöntemdir. YSA, özellikle sınıflandırma problemleri için yaygın şekilde kullanılmaktadır. YSA, geriye yayılma algoritmasını kullanarak ağ

hatasını minimize ederek birimlerin en az hata ile sınıflarına atanması için ağın ağırlıklarını adım adım hesaplar [5 ,6, 7, 16, 18].

Lojistik Regresyon Analizi ise bağımlı değişkenin kategorik ve bağımsız değişkenlerin karışık ölçekli olması durumunda belirli bir dağılım varsayımına bağımlı kalmaksızın bağımlı değişken ile bağımsız değişkenler arasındaki neden-sonuç ilişkisinin belirlenmesinde kullanılan bir yöntemdir. LR, en çok olabilirlik tahmin yöntemini kullanarak veri setinden elde edilen olasılığı maksimum yapan bilinmeyen parametre değerleri tahmin eder. Böylece olabilirlik fonksiyonunu maksimum yapan parametre tahminleri seçilir ve gözlenen veri ile en iyi örtüşen parametre tahminleri elde edilir [3, 4, 17].

Literatür taraması yapıldığında bu çalışmada seçilen üç sınıflandırma yönteminin genellikle ikili varyasyonları ile karşılaşılmıştır. Tarama sonucunda güncel çalışmalar tercih edilmiş olup bunlardan bazıları aşağıda özetlenmiştir.

Güneri ve ark.[4] başarı sınıflandırmasında lojistik regresyon ve sinir ağları yaklaşımı sonucunda elde edilen doğru sınıflandırma oranlarını karşılaştırdıklarında verilerin doğru sınıflandırma olasılıkları lojistik regresyon uygulaması ve sinir ağları yaklaşımı için %95.17 olarak bulunmuştur. İki yöntemin aynı sonucu vermiş olması sinir ağlarının atama problemlerinde kullanılabilirliğini göstermiştir.

Kurt ve ark.[5] yaptığı çalışmada, öğrencilerin alkol kullanımını etkileyen faktörler lojistik regresyon analizi (LR) ve yapay sinir ağları (YSA) ile incelenmiştir. Bu yöntemlerin alkol kullanan ve kullanmayan öğrencileri ayırmadaki etkinlikleri ROC eğrisi yöntemiyle karşılaştırılmış. Sonuç olarak LR, modelin parametre tahminleri ve OR değerleri hakkında bilgi vermesi ve sonuçlarının kolay yorumlanabilir olması açısından YSA'dan daha avantajlı olduğu görülmüştür. Bu nedenle eğer uygulama

sonucunda YSA'nın sınıflandırma performansı LR'den kötü ise LR modeli tercih edilmeli; eğer YSA'nın performansı LR'den daha iyi ise önemsiz değişkenlerin modelden çıkarılmasında LR, YSA için bir ön eleme yöntemi olarak kullanılmalıdır sonucuna varılmıştır.

Ocakoğlu'nun[6] yaptığı çalışmada, lojistik regresyon analizi ve yapay sinir ağlarının sınıflama etkinliklerini karşılaştırmayı amaçlamaktadır. Lojistik regresyon analizi ve yapay sinir ağları yöntemleri, bireylerin sınıflandırma oranlarına göre karşılaştırılmıştır. Buna göre YSA modelleri ile sınıflandırmanın LRA kullanılarak yapılan sınıflandırmadan daha iyi sonuçlar verme eğiliminde olduğu ayrıca yine aşırı eğitme, mimarinin hatalı oluşturulması vb. problemleri olmayan YSA modellerinin daha iyi öngörü performansı sağlayabildiği görülmüştür.

Karakış'ın[7] yaptığı çalışmada, meme kanseri hastalarının koltuk altı lenf nod durumlarını belirleyen SLNB ve AD ameliyatları olmaksızın, her hastanede kolaylıkla elde edilebilir olan klinik ve patolojik verilerinin girildiği YSA' nın, hastaların koltuk altı lenf nod durumunu belirlemesi amaçlanmıştır. Çalışma için Ankara Numune Eğitim ve Araştırma Hastanesi ve Ankara Onkoloji Eğitim ve Araştırma Hastanesi'ne başvuran ve meme kanseri 270 kişinin verileri kullanılmıştır. Çok katmanlı ileri yayılımlı yapay sinir ağı modeli sonuçları regresyon ve korelasyon katsayılarına bakılarak değerlendirilmiştir. Aynı zamanda tıbbi çalışmalarda tıp testlerinin doğruluklarını tespit amaçlı kullanılan ROC analizi ile elde edilen eğitim ve test veri sonuçlarının belirlilik, duyarlılık ve doğruluk sonuçları değerlendirilmiştir. YSA'nın sonuçlarının kıyaslanması için lojistik regresyon yöntemi kullanılmıştır. Lojistik regresyon ile hangi verilerin anlamlı olduğuna bakılmış, anlamlı olan verilerle yapay sinir ağı tekrar eğitilmiş ve test

edilmiştir. Lojistik regresyon ve seçilen YSA modelleri kıyaslandığında YSA değerleri daha başarılı olduğu görülmüştür.

Kıran'ın[8] yaptığı çalışmada, veri madenciliği yöntemleri içerisinde sınıflama ve regresyon modellerinden en çok kullanılan karar ağaçları ile lojistik regresyonun sınıflama özellikleri karşılaştırılarak gerçek bir veri seti üzerinde uygulama yapılmış ve söz konusu iki yöntemin başarısını göstermek amaçlanmıştır. Sonuçta her iki modelin % 90'ın üzerinde sınıflandırma başarısı gösterdiği dikkat çekerken, C&RT yönteminin daha yüksek sınıflandırma başarısına sahip olduğu tespit edilmiştir. Bu noktadan hareketle C&RT ve lojistik regresyon analizi ile yapılan çalışmalarda hata riskini en aza indirmek amacıyla C&RT yönteminin kullanılması daha uygun bulunmuştur.

King ve ark.[39] sembolik öğrenme (C&RT, C4.5, NewID, ITrule,Cal5, CN2), istatistik (Naive Bayes, k-en yakın komşuluk, kernel yoğunluk, doğrusal ayırma, karesel ayırma, lojistik regresyon, Bayes Ağları) ve yapay sinir ağları (geri yayınım - merkez tabanlı uzaklık fonksiyonu) yöntemleri arasında karşılaştırma yapmışlardır. Bu yöntemleri beş adet resim, iki adet tıp, ikişer mühendislik ve finans veri kümeleri üzerinde uygulamışlardır. Daha iyi sonuçlar üreten algoritmanın, üzerinde araştırma yapılan veri kümesine bağlı olduğu sonucuna ulaşmışlardır ve örnek olarak iki kategorili değişkenlerde gözlenme oranının %38'in üzerinde olduğu veri kümelerinde sembolik öğrenme algoritmalarının daha verimli sonuçlar ürettiğini belirtmişlerdir.

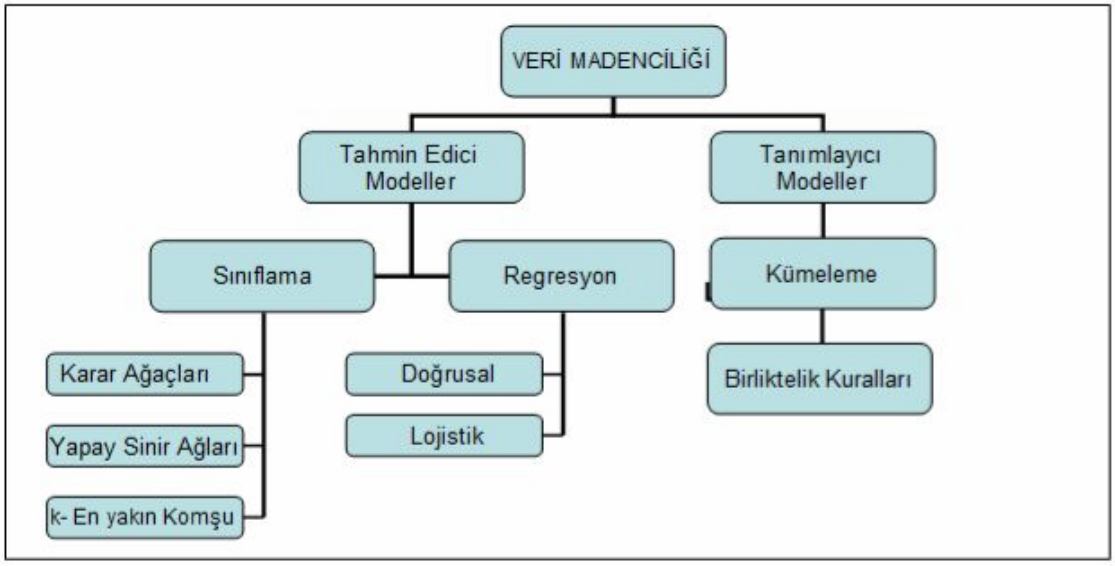
Sabzecari ve ark.[10] özel bir bankanın kredi derecelendirme amaçlı veri kümesi üzerinde uyguladıkları veri madenciliği yöntemlerini karşılaştırmışlardır. Bankalar, kredi verirken, veri madenciliği yöntemleri ile kredi başvurusunda bulunan müşterileri değerlendirerek müşteriye kredi verilmesinin uygun olup olmadığını belirlemektedirler. Sabzecari ve ark. Probit ve lojistik regresyon, C&RT, yapay sinir ağları, bagging ve

MARS algoritmalarını karşılaştırarak sonuçlarını değerlendirmişlerdir. Oldukça küçük bir veri kümesi üzerinde yaptıkları bu çalışma sonucunda istatistiksel modeller arasından lojistik regresyon ve makine öğrenme modelleri arasından da bagging modelinin daha başarılı sonuçlar ürettiğini ifade etmişlerdir.

Zurada ve ark.[11] sağlık endüstrisinde kötü kredilerin belirlenmesinde karşılaştırdıkları yapay sinir ağları, karar ağaçları, lojistik regresyon, hafıza-tabanlı sebepleme ve bütünleştirilmiş model arasında karşılaştırılma yapmışlardır. Sonuçta yapay sinir ağlarının, lojistik regresyon algoritmasının ve bütünleştirilmiş modelin daha iyi kesinlik oranına sahip sonuçlar ürettiğini, karar ağaçlarının ise iyi kredi sınıflandırmasını daha yüksek bir doğruluk derecesiyle tespit ettiğini belirtmişlerdir.

2. GENEL BİLGİLER

Veri Madenciliğinde kullanılan modeller, temel olarak şekil 2.1'de görüldüğü üzere tahmin edici (predictive) ve tanımlayıcı (descriptive) olmak üzere iki ana başlık altında incelenmektedir [26].



Şekil 2.1. Veri Madenciliği Modelleri

Tahmin edici modellerin amacı, verilerden hareket ederek bir model geliştirmek ve kurulan bu model yardımıyla sonuçları bilinmeyen veri kümelerinin sonuç değerlerini tahmin etmektir. Eğer tahmin edilecek değişken sürekli bir değişkense tahmin problemi regresyon, kategorik bir değişkense sınıflama problemi olarak nitelendirilmektedir [2]. Tanımlayıcı modellerde ise karar vermeye rehberlik etmede kullanılabilecek mevcut verilerdeki örüntülerin tanımlanması sağlanmaktadır.

Veri Madenciliği modelleri fonksiyonlarına göre ise;

- Sınıflama (Classification) ve Regresyon,
- Kümeleme (Clustering),

- Birliktelik kuralları (Association Rules) şeklinde sınıflandırılmaktadır.

Şekil 2.1'de gösterilen veri madenciliği modellerinden sınıflama ve regresyon modelleri tahmin edici, kümeleme, birliktelik kuralları ve ardışık zamanlı örüntüler ise tanımlayıcı modellerdir [1].

2.1. SINIFLAMA VE REGRESYON MODELLERİ

Veri madenciliği yöntemleri içerisinde en yaygın kullanıma sahip olan, büyük veri kümelerini sınıflandırarak önemli veri sınıflarını ortaya koyan veya gelecek veri eğilimlerini tahmin etmede faydalanılan yöntemlerden bir tanesi sınıflama ve regresyon modelleridir.

Veri madenciliğinin en önemli alanlarından biri olan kümeleme, nesneleri birbirlerine olan benzerliklerine göre gruplara ayırmaktadır. Yani kümeleme modellerinde amaç, küme üyelerinin birbirlerine çok benzediği ancak özellikleri birbirlerinden çok farklı olan kümelerin bulunması ve veri tabanındaki kayıtların bu farklı kümelere bölünmesidir. Böylece nesneler, örneklenen kitle özelliklerini iyi yansıtan etkili bir temsil gücüne sahip olacaktır [1,2,3].

Sınıflama en çok bilinen veri madenciliği yöntemlerinden birisidir; resim, örüntü tanıma, hastalık tanılan, dolandırıcılık tespiti, kalite kontrol çalışmaları ve pazarlama konulan sınıflama yöntemlerinin sıklıkla kullanıldığı alanlardır. Sınıflama tahmin edici bir model olup, havanın bir sonraki gün nasıl olacağı veya bir kutuda kaç tane mavi top olduğunun tahmin edilmesi bir sınıflama işlemidir [27]. Matematiksel olarak sınıflama;

$D = \{t_1,t_2,...,t_n\}$ bir veri tabanı ve her bir t_i bir kayıt (gözlem),

$C = \{C_l,C_2,...,C_m\}$ ise m adet sınıftan oluşan sınıflar kümesini temsil etmek üzere,

$f:$ D $\rightarrow$C ve her bir t_i bir sınıfa dahildir. Ayrıca her bir Cj ayrı bir sınıftır ve her bir sınıf kendisine ait kayıtları içerir. Yani,

$C_j = \{ t_i lf\{ t_i) = C_j, \backslash < i < n, ve\ t_t GD \}$ olarak tanımlanmaktadır.

Veri madenciliği yöntemleri içerisinde en yaygın kullanıma sahip olan sınıflama ve regresyon modelleri arasındaki temel farklılık bağımlı değişkenin kategorik veya sürekli olmasıdır. Daha önce de bahsedildiği gibi, eğer bağımlı değişken sürekli ise problem regresyon problemi, kategorik ise problem sınıflama problemi olarak adlandırılır. Ancak lojistik regresyon gibi kategorik değerlerin de tahmin edilmesine olanak veren yöntemlerle, her iki model giderek birbirine yaklaşmakta ve bunun bir sonucu olarak aynı yöntemlerden yararlanılması mümkün olmaktadır.

Sınıflama ve regresyon modellerinde kullanılan başlıca yöntemler;

- Yapay Sinir Ağları (Artificial Neural Networks),
- Karar Ağaçları (Decision Trees),
- Lojistik Regresyon (Logistic Regression),
- Genetik Algoritmalar (Genetic Algorithms),
- K-En Yakın Komşu (K-Nearest Neighbor),
- Bellek Temelli Nedenleme (Memory Based Reasoning),
- Naive-Bayes,
- Bulanık Küme Yaklaşımı (Fuzzy Set Approach) 'dır [8].

Çalışmanın kapsamında yukarıda sayılan söz konusu yöntemlerden Karar Ağaçları, Lojistik Regresyon ve Yapay Sinir Ağları'nın üzerinde durulacaktır.

2.1.1. Karar Ağaçları

Karar Ağaçları, verileri belli özellik değerlerine göre sınıflandırmaya yarar. Bunun için algoritmaya girdi olarak çeşitli veri özellikleri belirlenir. Çıktı olarak da

belli bir veri özelliği seçilerek algoritmanın bu çıktı özelliği değerlerine ulaşmak için hangi girdi özelliklerinin bir arada olması gerektiğini ağaç veri yapıları şeklinde keşfetmesi sağlanır [28].

Karar ağaçları, basit karar verme adımları uygulanarak, çok sayıda kayıt içeren bir veri kümesini çok küçük kayıt gruplarına bölmek için kullanılan bir yapıdır [29]. Her başarılı bölme işlemiyle, sonuç gruplarının üyeleri bir diğeriyle çok daha benzer hale gelmektedir.

Bu teknikte sınıflandırma için bir ağaç oluşturulur, daha sonra veri tabanındaki her bir kayıt bu ağaca uygulanır ve çıkan sonuca göre de bu kayıt sınıflandırılır. Karar ağaçları veri setinin çok karmaşık olduğu durumlarda bile, bağımlı değişkeni etkileyen değişkenleri ve bu değişkenlerin modeldeki önemini basit bir ağaç yapısı ile görsel olarak sunabilmektedir [8,28].

Karar ağacı yöntemini kullanarak verinin sınıflanması temel olarak iki adımdan oluşmaktadır. Birinci adım; önceden bilinen bir eğitim verisinin model oluşturmak amacı ile sınıflama algoritması tarafından çözümlendiği öğrenme basamağıdır. Öğrenilen model, sınıflama kuralları veya karar ağacı olarak gösterilir. İkinci adım ise eğitim verisinin sınıflama kurallarının veya karar ağacının doğruluğunu belirlemek amacıyla test edilerek kullanıldığı sınıflamadır. Eğer doğruluk kabul edilebilir oranda ise, kurallar yeni verilerin sınıflanması amacıyla kullanılır [8].

Karar ağacı algoritmalarını bir probleme uygulayabilmek için aşağıdaki koşulların sağlanması gerekir:

- ✓ Olayların özelliklerle ifade edilebilmesi gerekir. Nesnelerin belli sayıda özellik değerleriyle ifade edilebilmesi gerekir. Örneğin; soğuk, sıcak vb. özelliklerin sürekli ve ya kesikli olması fark etmez.

- ✓ Sınıfları belirleyebilmek için gereken ayırıcı özelliklerin olması gerekir. Karar ağacı, adında belirtildiği şekilde ağaç görünümünde bir tekniktir. Karar düğümleri, dallar ve yapraklardan oluşur [19].

Karar ağaçlarının kök, dallar ve yapraklardan oluşan ağaca benzeyen bir yapısı olup, örnekteki tüm gözlemleri kapsayan bir kök ile başlayıp aşağıya doğru inildikçe veriyi alt gruplara ayıran dallara ayrılırlar. Bu kökten dallara doğru büyüyen ağaç yapısında her boğum "düğüm" dür, oluşan ağaçlarda homojen olmayan düğümlere "yavru düğümü (child node)", homojen düğümlere ise "terminal düğüm (parent node)" adı verilir [30]. Düğümler üzerinde niteliklerin test işlemi yapılmakta ve test işleminin sonucu ağacın veri kaybetmeden dallara ayrılmasına neden olmaktadır. Her düğümde test ve dallara ayrılma işlemleri ardışık olarak gerçekleşmekte ve sonuç olarak ağaç sınıflar ile son bulmaktadır.

Düğüm: Veriye uygulanacak test tanımlanır. Her düğüm bir özellikteki testi gösterir. Test sonucunda ağacın dalları oluşur. Dalları oluştururken veri kaybı yaşanmaması için verilerin tümünü kapsayacak sayıda farklı dal oluşturulmalıdır.

Dal: Testin sonucunu gösterir. Elde edilen her dal ile tanımlanacak sınıfın belirlenmesi amaçlanır. Ancak dalın sonucunda sınıflandırma tamamlanamıyorsa tekrar bir karar düğümü oluşur. Karar düğümünden elde edilen dalların sonucunda sınıflandırmanın tamamlanıp tamamlanmadığı tekrar kontrol edilerek devam edilir.

Yaprak: Dalın sonucunda bir sınıflandırma elde edilebiliyorsa yaprak elde edilmiş olur. Yaprak, verileri kullanarak elde edilmek istenen sınıflandırmanın sınıflarından birini tanımlar [30].

Karar ağacında, tanımlanmış olan soruya ilişkin cevap gruplara ayrılmaktadır. Cevaplar soruya verilecek bir ölçüt belirlendikten sonra setler arasındaki riski

maksimize edecek şekilde bölünmekte ve en iyi bölünmeyi bulmak için her soruda aynı işlem tekrar edilmektedir. Bir soru için grup oluşturulduktan ve gruplar arasındaki risk maksimize edildikten sonra oluşan iki grup için bu işlemler devam ettirilmektedir. Bu işlemlere istatistiksel olarak anlamlı bir fark bulunana kadar devam edilmekte, istatistiksel olarak anlamlı bir fark bulunmadığında ise son verilmektedir. Ayrıştırma işlemi tamamlandıktan sonra ise o grup içerisinde yer alan gözlemlerin oranına göre grup değerlendirilmektedir [31].

Karar ağaçlarının oluşturulması ağaç oluşturma ve ağaç budama basamaklarından oluşur.

Ağaç oluşturma: Veri kaynağındaki bütün nesneleri içeren kök düğümden başlar, yinelemeli olarak her düğümde var olan nesneleri seçilecek olan bir niteliğe göre farklı dallara ayırarak bütün nesneleri sınıflandıracak şekilde yaprak düğümlere bölene kadar, ya da ayırım yapıcı bir nitelik kalmayana kadar devam eder. Sınama düğümlerinde eldeki nesnelerin hangi niteliğe göre alt düğümlere bölündüğünde en çok verimin alınacağını bulmak ve dallanmayı bu niteliğe göre yapmak algoritmanın gücünü arttırır. Nesneler alt düğümlere bölündüğünde alt düğümlerdeki nesnelerin türdeşliği ne kadar yüksek olursa o düğümdeki dallanma o kadar verimli olur. Bu sebeple, her düğümde, sınaması yapılacak olan nitelik (o düğümdeki nesneleri alt düğümlere böldüğünde) homojenite bakımından en yüksek kazancı sağlayacak nitelik olarak seçilir [8,31].

Ağaç Budama: Ağaç oluşturma basamağı, verileri tamamen aynı sınıf üyelerinden oluşan yapraklara bölünceye ya da karşılaştıracak nitelik kalmayıncaya kadar bölmeler. Bu algoritmanın sonucu olarak, çok derin ya da çok az deneme kümesi örneği içeren yaprak düğümlere sahip ağaçlar oluşabilir. Böyle bir ağacı öğrenme

kümesi üzerinde test edince elbette ki doğruluğu çok yüksek sonuçlar verir. Ancak böyle bir model henüz görülmemiş örneklerle karşılaşırsa çok kötü doğruluklu sonuçlar üretebilir. Böyle bir model verimli değildir ve veriyi genellemekten uzaktır. Böyle bir modelin sahip olduğu bu özelliğe ***aşırı uyum (overfitting)*** denir. Aşırı uyum bir modelde istenmeyen bir sonuçtur [27,31].

Aşın uyum genelde verideki gürültüden (hatalı sınıf değeri, yanlış değerli değişkenler) kaynaklandığı gibi problem alanının karmaşıklığından ya da rastgelelikten kaynaklanabilir.

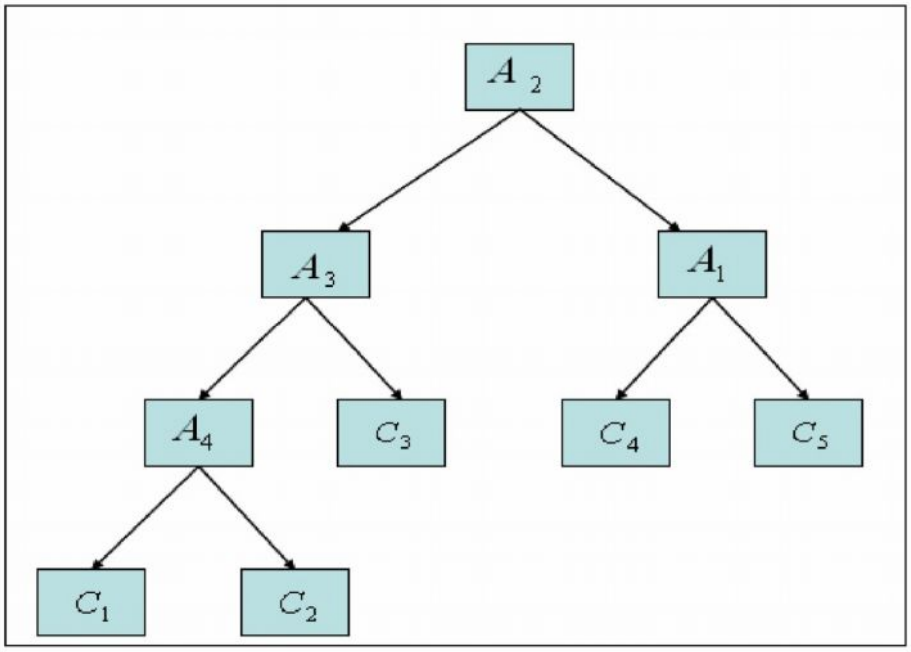
Aşırı uyumu azaltmak için ağaçlarda budama işlemi uygulanır. Budama işlemi, bazı dalların ya da alt dalların kaldırılarak o dala ait nesnelerin yoğun olduğu sınıfla etiketlenmiş yaprak düğümlerle yerleştirilmesiyle gerçekleştirilir. Ağaç oluşturulurken erken-dur yöntemiyle erken-budama yapılabileceği gibi ağaç oluşturulduktan sonra budama geç-budama yapılabilir. Geç-budama yönteminin daha başarılı olduğu bilinmektedir. Zira erken-budama yöntemi hatalı sonuçlara yol açabilir, çünkü henüz dallanma yapılmamış bir dal budandığında, ağacın o noktadan sonra ne şekil almış olacağı o aşamada bilinmemektedir. Ancak geç-budama yapılırken ağaç zaten oluşmuş bulunmaktadır ve hangi dalların aslında gereksiz olduğu, aşın uyum yarattığı bilinmektedir. Geç-budama yapılırken düğümlerdeki beklenen hata değerine bakılır. Eğer bir düğümdeki beklenen hata miktarı, o düğüme ait alt dallardaki beklenen hata miktarından küçük olursa alt dallar budanır [31,32].

$D = \{t_1,...,t_n\}$ bir veri tabanı olmak üzere, t_t, $t_i=\{t_1 \ldots t_{in}\}$ den ve bu veri tabanı $\{A_1, A_2, \ldots, A_n\}$ alanlarından oluşmaktadır.

Bunun dışında $C = \{C_1,...,C_{,,}\}$ kadar da sınıf verilmiş olduğunda,

- Her bir düğümü At alanıyla tanımlanmış,

- Her düğümden ayrılan kollar bu alanla ilgili bir soruya yanıt veren,
- Her yaprağın bir sınıf olduğu karar ağacı şekil 2.2'da gösterilmiştir [27].



Şekil 2.2. Karar Ağacı Yapısı

Şekil 2.2'de görülen karar ağacındaki A_1, A_2 ,..., A_n 'den her biri bir düğümü oluşturmakta ve her düğüm kendinden sonra iki dala ayrılmaktadır. Bu ayrılma işlemi sürecinde, A_i düğümü hakkında cevabı veri tabanında bulunacak bir soru sorulmakta ve verilen yanıta göre de bir dal izlenmektedir. Ağaçtaki C_1 ,C_2 ,...,C_n 'lerin her biri birer yaprağı aynı zamanda sınıfı temsil etmektedirler.

Karar ağaçları oluşturulurken kullanılan algoritmanın ne olduğu önemli bir husustur. Kullanılan algoritmaya göre ağacın şekli değişebilir. Bu durumda değişik ağaç yapıları da farklı sınıflandırma sonuçları verecektir. Kök denilen ilk düğümü oluşturan A_i 'nin farklı olması, en uçtaki yaprağa ulaşırken izlenecek yolu ve dolayısıyla sınıflandırmayı da değiştirecektir [27].

Değişkenlerin seçiminde yinelemeli olan algoritmanın döngüden çıkması için o düğümdeki tüm öğelerin aynı sınıfa dâhil olması şartı vardır. Eğer kalan değerler sadece bir sınıfa aitse veya sınıflandırılabilecek değer kalmadıysa döngüsel algoritma sonlanır ve karar ağacı oluşturulmuş olur. Sonuçta oluşan sınıflardaki her bir eleman aynı sınıfın diğer elemanları ile benzer özellikler gösterir. Ağaç yapısı heterojen yapıdaki veri kümesinin daha küçük ve homojen bir yapıya dönüşmesi için kurallar tanımlar. Ağaç inşası sonunda elde edilen ağaç maksimum ağaç olarak adlandırılır ve öğrenme kümesindeki deney ünitelerine en uygun ağaçtır. Ancak maksimum ağaç pratikte iki dezavantaja sahiptir [32].

- Maksimum ağaç başlangıç veri setini (öğrenme kümesini) kusursuz biçimde tanımlar çünkü eklenen her bağımsız değişken hatalı sınıflama oranını düşürür. Bu durumda, maksimum ağaç veri için olması gerekenden daha iyi bir tahmin modeli sunar. Ancak, başlangıç veri setine aşırı uyumlu maksimum ağaçlar farklı bir veri seti söz konusu olduğunda iyi bir tahmin sağlayamazlar.
- Bir sınıflama ağacının karmaşıklık ölçüsü o ağacın terminal düğüm sayısına eşittir. Terminal düğüm sayıları ve dolayısıyla karmaşıklığı yüksek olan maksimum ağacın anlaşılması ve yorumlanması güçtür.

Maksimum ağacın pratikte ortaya çıkardığı bu sorunların çözümü için maksimum ağacın budanması gereklidir. Maksimum ağacın budanması daha küçük ağaçlar dizisi oluşturur ve oluşturulan bu dizi içerisinden optimum ağaç seçilir. Optimum ağaç maksimum ağaçtan daha az karmaşıklığa sahiptir ancak öğrenme kümesine maksimum ağaçtan daha az uyumludur ve hatalı sınıflama oranı da daha yüksektir [27,31,32].

Karar ağacının kurulması için kullanılacak girdi olarak bir dizi kayıt verilirse bu kayıtlardan her biri aynı yapıda olan birtakım özellik/değer çiftlerinden oluşur. Bu özelliklerden biri kaydın hedefini belirtir. Amaç, hedef-olmayan özellikler kullanılarak edef özellik değerini doğru kestiren bir karar ağacı belirlemektir. Hedef özellik çoğunlukla sadece {evet, hayır}, veya {başarılı, başarısız} gibi ikili değerler alır [19].

Bir karar ağacı ne bildiğimizi (örn., öğrenme verisi) özetlediği için değil, yeni durumların sınıflamasını doğru yaptığı için önemlidir.

"Karar ağacı tekniğini kullanarak verinin sınıflanması üç aşamadan oluşur.

- ***Öğrenme:*** Önceden sonuçları bilinen verilerden (eğitim verisi) model oluşturulur.
- ***Sınıflama:*** Yeni bir test verisi kümesi modele uygulanır, bu şekilde karar ağacının doğruluğu belirlenir. Test verisine uygulanan bir modelin doğruluğu, yaptığı doğru sınıflamanın test verisindeki tüm sınıflara oranıdır. Her test örneğinde bilinen sınıf, model tarafından tahmin edilen sınıf ile karşılaştırılır.
- ***Uygulama:*** Eğer doğruluk kabul edilebilir oranda ise, karar ağacı yeni verilerin sınıflanması amacıyla kullanılır [33].

Test verisini en iyi şekilde dallara ayıran özellik tespit edilerek, bu özellik karar ağacının oluşturulmasında daha önce seçilir. Böylece daha iyi bir karar ağacı oluşturulur. En iyi dallara ayıran özelliğin tespitinde çeşitli ölçütler geliştirilmiştir.

Bu ölçütlerin bazı örnekleri şunlardır:

$p(i \mid t)$, i' sınıfına ait verilerin, verilen bir t düğümündeki bölünmesini göstersin.

c sınıf sayısıdır ve entropi hesaplamasında $0 \log_2 0 = 0$ şeklinde düşünülmüştür.

$$\text{Entropi(t)} = -\sum_{c-1} p(i \mid t) \log_2 p(i \mid t), \qquad (2.1)$$

$$i=0$$

Entropi bir düğümün ne kadar bilgi verici olduğunu ölçmede kullanılır. Bu "*İyi*" ile ne kastedildiğini belirtir. Bu kavram Claude Shannon tarafından ilk kez Bilgi Teorisi içinde tanımlanmıştır.

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i \mid t)]^2, \tag{2.2}$$

$$\text{Sınıflandırma hatası}(t) = 1 - \max_i [p(i \mid t)],$$

Karar ağacı oluşturulduktan sonra, kökten yaprağa doğru inilerek kurallar belirlenir. Ancak çok sayıda dal ve düğümden oluşan karar ağaçlarında kuralların belirlenmesi zorlaşır. Karar ağacı modelini daha okunabilir hale getirmek için kuralların yazımında IF-THEN (Eğer-ise-O Zaman) biçiminde şartlı ifadeler kullanılır. IF (Eğer-ise) kısmı dalın sonuna giden yol üzerindeki tüm testleri içerirken THEN (O Zaman) kısmı son sınıflamayı gösterir. IF THEN yapısındaki kurallara Karar Kuralları (*Decision Rules*) denir [18,22].

Karar ağaçları, model kurulumu ve sonuçlarının yorumlanmasının kolay olması, veritabanı sistemleri ile kolayca entegre edilebilmeleri ve güvenilirliklerinin iyi olması nedenleri ile sınıflama modelleri içerisinde en yaygın kullanıma sahip tekniktir [34].

Karar ağaçlarının güçlü yönleri aşağıdaki gibi özetlenebilir:

- ✓ Karar ağaçları anlaşılabilir kurallar üretirler.
- ✓ Karar ağaçları aşırı hesaplamaya gerek kalmadan sınıflandırma yaparlar.
- ✓ Karar ağaçları hem sürekli ve hem de kesikli değişkenler için uygundur.
- ✓ Karar ağaçları sınıflandırma ve kestirim için hangi alanların en önemli olduğunu açık biçimde gösterir [35].

Karar ağaçlarına dayalı olarak geliştirilen birçok algoritma vardır. Bu algoritmalar kök, düğüm ve dallanma kriteri seçimlerinde izledikleri yol açısından birbirlerinden ayrılırlar. Karar ağacı oluşturmak için geliştirilen bu algoritmalar arasında;

- CHAID (Chi-Squared Automatic Interaction Detector: Otomatik Ki-Kare Etkileşim Belirleme),
- C&RT (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 yer almaktadır.

2.1.2. Yapay Sinir Ağları

İnsanların beynin çalışmasını sayısal bilgisayarlar üzerinde taklit etmek istemesi sonucunda yapay sinir ağları ortaya çıkmıştır. Temelde beynin çalışma yapısı, insanlar gibi düşünen ve öğrenen sistemler elde etme fikrinin olması, çalışmaları insan beynini oluşturan hücrelerin incelenmesi üzerine yoğunlaştırmıştır. Bu çalışmalar esnasında her bir nöronun diğer nöronlar ile ilişkili olduğu ve aldığı bazı girdileri çıktıya dönüştürdüğü gözlemlenmiştir. Günümüzde yapay sinir ağlan olarak adlandırılan bu alan, birçok nöronun belirli kurallar çerçevesinde bir araya gelerek bir işlevin

gerçekleştirmesi esnasındaki yapısal, matematiksel ve felsefi sorunlara cevap bulmaya çalışan bir bilim dalı olmuştur [45].

YSA kavramı beynin çalışma ilkelerinin sayısal bilgisayarlar üzerinde taklit edilmesi fikri olarak ortaya çıkmış ve ilk çalışmalar beyni oluşturan biyolojik hücrelerin, ya da literatürdeki ismi ile nöronların matematiksel olarak modellenmesi üzerinde yoğunlaşmıştır [46].

Yapay sinir ağı, beynin belirli bir işi veya fonksiyonu yerine getirme yöntemini modelleyen bir makinedir. Ağın yapısı, elektronik sistemler veya bilgisayar yazılımları ile tanımlanmaktadır. Tanımlanan bu model, sinir hücresi veya işlem birimi adı verilen hücreler arasındaki bağlantıyı kullanmakta ve bu işlemler esnasında öğrenme adı verilen bir süreç ile performansını artırabilmektedir. Bu noktada yapay sinir ağı kavramını deneysel bilgi saklama ve kullanıma hazır hale getirme yeteneğine sahip basit işleme birimlerinden oluşan, çok yoğun, paralel ve dağıtılmış düzende çalışan bir işlemci olarak tanımlamak mümkündür. İnsan beyni ile benzerliği ise bilgiyi öğrenme yoluyla elde etmesi ve bilginin depolanması için sinir hücreleri arası bağı kullanmasıdır [47].

2.1.2.1. Yapay Sinir Ağı Nedir?

Bir yapay sinir ağı, biyolojik sinir ağlarının karakteristiklerine benzer karakteristiklere sahip bir bilgi işleme sistemidir. YSA, insanın idrak etmesi ve biyolojik nöron yapısının matematiksel modelinin aşağıdaki kurallar varsayılarak genelleştirilmesi sonucunda oluşturulmuştur [48]:

- Bilgi işleme, nöron adı verilen birçok basit elemanlarda gerçekleşir;
- Sinyaller, nöronlar arasındaki ilişkiyi sağlayan bağlantılarla iletilir;

- Her bir bağlantının bir ağırlık değeri vardır ve bu değer, gerçek nöronlarda olduğu gibi sinyal geçişini üretmektedir;
- Sinir ağı içindeki her bir nörona aynı bir aktivasyon fonksiyonu uygulanır (genelde bu doğrusal olmayan bir fonksiyondur) ve bu fonksiyonun çıkış değeri sayesinde nöronun çıkış sinyali hesaplanır;

Herhangi bir yapay sinir ağı;

- Nöronlar arasındaki bağıntının bir modeli yani mimarisi ile,
- Bağlantılardaki ağırlıkların hesaplanması (bu hesaplama, eğitim kuralı ya da öğrenme algoritması olarak da adlandırılır) ile,
- Aktivasyon fonksiyonu ile tanımlanabilir.

Yapay sinir ağlarının karakteristik özellikleri uygulanan ağ modeline göre değişmektedir. Ancak tüm modeller için genel özellikler şunlardır: [49]

- Yapay sinir ağları makine öğrenmesi gerçekleştirirler.
- Programları çalışma stili bilinen programlama yöntemlerine benzememektedirler.
- Bilgiyi saklarlar.
- Örnekleri kullanarak öğrenirler.
- Güvenle çalıştırılabilmeleri için önce eğitilmeleri ve performanslarının test edilmesi gerekmektedir.
- Görülmemiş örnekler hakkında bilgi üretirler. Bunu genelleme özelliği sayesinde yaparlar.
- Algılamaya yönelik olaylarda kullanılabilirler.
- Şekil ilişkilendirme ve sınıflandırma yapabilirler.

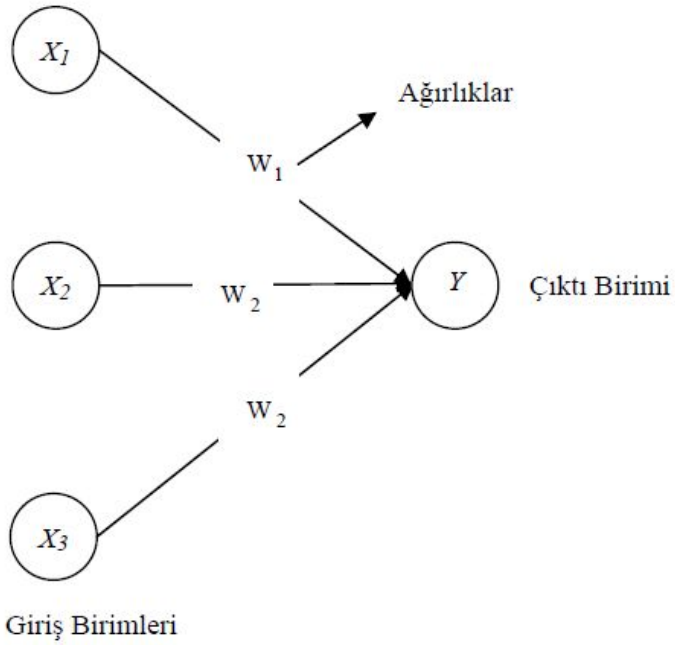
- Örüntü tamamlama gerçekleştirebilirler.
- Kendi kendine organize etme ve öğrenebilme yetenekleri vardır.
- Eksik bilgi ile çalışabilmektedirler.
- Hataya ve gürültüye karşı duyarlılığa ve toleransa sahiptirler.
- Belirsiz, tam olmayan bilgileri işleyebilmektedirler.
- Dereceli bozulma (Graceful degradation) gösterirler.
- Dağıtık belleğe sahiptirler. Veri dağıtılmış birleşik hafıza yapısı kullanılır ve bilgi farklı formlara dönüştürülerek işlenebilir.
- Sadece nümerik bilgiler ile çalışabilmektedirler [49,50].

Bir yapay sinir ağı, nöron, birim, hücre ya da düğüm olarak adlandırılan çok sayıdaki basit işlem birimlerinden oluşur. Her bir nöron, diğer bir nörona belli bir ağırlık değerine sahip olan haberleşme bağlantılarıyla bağlanır. Ağırlıklar, yapay sinir ağının bir problemi çözmesi için gerekli olan bilgiyi hazırlamaktadır. YSA çok çeşitli problemlerin çözümünde kullanılabilirler. Örnek olarak, bilgileri ve numuneleri saklamada ve onları daha sonra tekrar tanımada, numuneleri sınıflandırmada, giriş numunelerinin çıkış numunelerine dönüştürülmesinde, benzer örneklerin gruplandırılmasında ya da doğal olmayan optimizasyon problemlerinin çözümlerinin bulunmasında ve daha pek çok alanda YSA çok geniş bir biçimde kullanılabilir [50].

Örnek olarak Şekil 2.3'de gösterilen bir Y nöronunu düşünelim. Bu nöron X_1,X_2 ve X_3 nöronlarından giriş sinyallerini alır. Bu nöronların aktivasyonları yani çıkış sinyalleri, sırasıyla x_1,x_2 ve x_3 'tür. Bağlantılar üzerindeki ağırlıklar X_1,X_2 ve X_3 nöronlarından Y nöronuna doğru sırasıyla w_1,w_2 ve w_3 'tür. Ağ girişi olan y _ in değeri

X_1,X_2 ve X_3 'den Y'ye giden ağırlıklı sinyallerin toplamıdır. y _ in değeri aşağıdaki eşitlikteki gibi hesaplanır [50].

$$y\_in = w_1x_1 + w_2x_2 + w_3x_3 \quad (2.3)$$



Şekil 2.3. Basit Bir Yapay Nöron

Y nöronunun aktivasyonu y, ağa giriş değerlerinin bir fonksiyonu olarak tanımlanır.

$$y = f(y\_in)$$

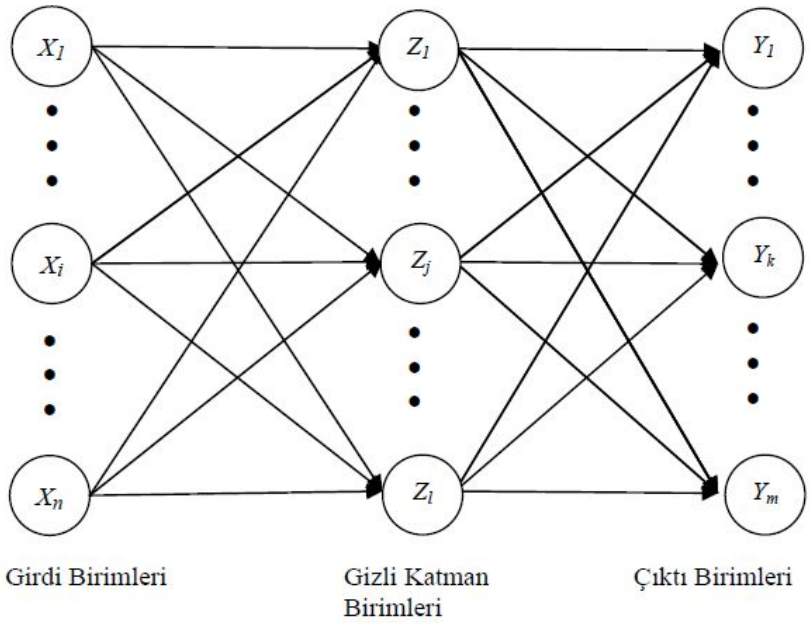
Bu fonksiyon, S-biçimli sigmoid fonksiyon olabilir. Sigmoid fonksiyonun formu aşağıda verildiği gibidir.

$$f(x) = \frac{1}{1+\exp(-x)} \quad (2.4)$$

Bu fonksiyonu, diğer aktivasyon fonksiyonlarından biri de olabilir.

Y nöronunun Z1 ve Z2 nöronlarına v1 ve v2 ağırlıklarıyla bağlandığı varsayılsın. Bu durum, Şekil 2.4'de gösterilmektedir. Y nöronu y sinyalini diğer birimlere gönderir. Bununla birlikte, genel olarak Z1 ve Z2 nöronları tarafından alınan sinyaller farklı olmaktadır; çünkü her bir sinyal aktarıldığı bağlantıda bulunan v1 ve v2 ağırlıkları ile orantılıdır. Z1 ve Z2 'nin aktivasyonları olan z1 ve z2 değerleri, sadece tek bir nörona bağlı değildir. Onlar birbirinden farklı birden fazla nörondan gelen sinyallere bağlıdır.

Şekil 2.4' deki yapay sinir ağı basit olmasına rağmen, gizli birimin görünümü ve doğrusal olmayan aktivasyon fonksiyonu sayesinde birçok problemi çözebilir. Başka bir yönden gizli bir birime sahip yapay sinir ağının öğretilmesi yani ağırlıkların optimal değerlerinin bulunması oldukça zordur [6].

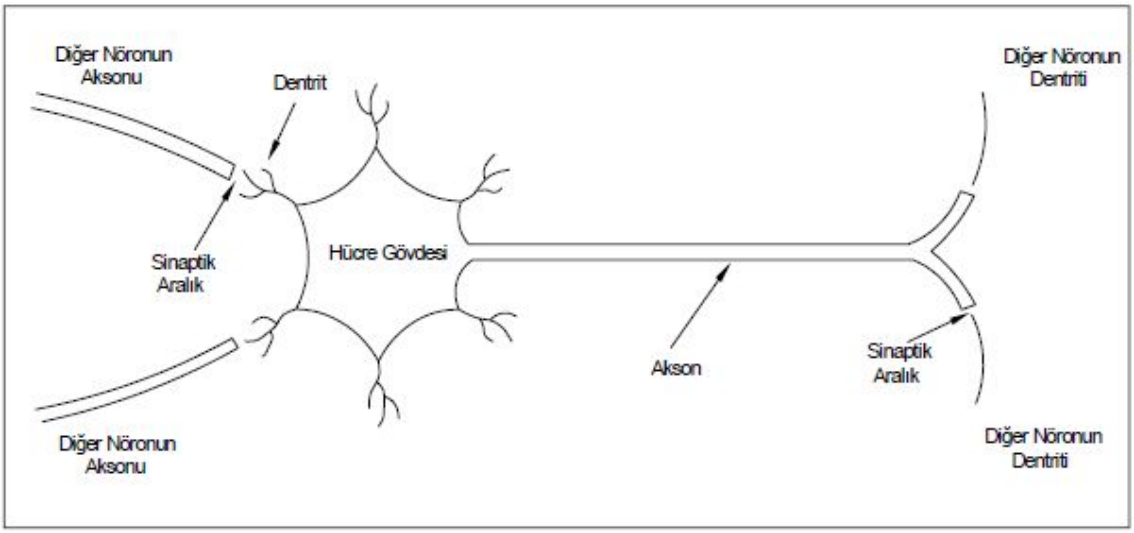


Şekil 2.4. Basit Bir Yapay Sinir Ağı [6]

2.1.2.2. Biyolojik Sinir Ağları

Şekil 2.5. Biyolojik Sinir Sisteminin Blok Gösterimi

Biyolojik sinir sistemi verinin alınması, yorumlanması ve karar üretilmesi gibi işlevlerin yürütüldüğü beyni merkezde bulunduran 3 katmanlı bir yapıdır. Uyarılar alıcı sinirler ile birlikte elektriksel sinyallere dönüştürülerek beyne iletilir. Beynin oluşturduğu çıktılar ise, tepki sinirleri tarafından belirli tepkilere dönüştürülür.



Şekil 2.6. Biyolojik Sinir Hücresi [5,47]

Sinir ağı yapısında bilgilerin alıcı ve tepki sinirleri arasında ileri ve geri beslemeli olarak değerlendirilmesi ve sonucunda tepkilerin oluşması, kapalı bir çevrim sürecine benzemektedir. Temel işlem elemanı sinir hücreleridir. İnsan beyninde

yaklaşık 10 milyar sinir hücresi olduğu tahmin edilmektedir. Şekil 2.6 da gösterildiği gibi sinir hücreleri; hücre gövdesi, gövdeye giren alıcı lifler (dentrit) ve gövdeden çıkan sinyal iletici lifler (akson) olmak üzere 3 temel bileşenden meydana gelir.

Dentritler aracılığı ile bilgiler diğer hücrelerden hücre gövdesine iletilir. Hücrelerde oluşan çıktılar ise akson yardımı ile bir diğer hücreye aktarılır. Aktarımın gerçekleştiği bu noktada aksonlarda ince yollara ayrılabilmekte ve diğer hücrenin dentritlerini oluşturmaktadırlar. Akson-dentrit bağıntısını oluştuğu bu noktalara sinaps adı verilir [5,47].

Sinapsa ulaşan ve dentritler tarafından alınan bilgiler genellikle elektriksel darbelerdir, fakat bu bilgiler sinapstaki kimyasal ileticilerden etkilenirler. Hücrenin tepki oluşturması için bu tepkilerin belirli bir sürede belirli seviyeye ulaşması gerekmektedir. Bu değer eşik değeri olarak adlandırılır [23,47].

YSA'lar, insan beyninin çalışma prensibi örnek alınarak geliştirilmeye çalışılmıştır ve aralarında yapısal olarak bazı benzerlikler vardır [52]. Bu benzerlikler Tablo 2.1'de ve istatistiksel terimler yapay sinir ağları terimleri arasındaki terminolojik ilişkiler de Tablo 2.2 de verilmiştir [50].

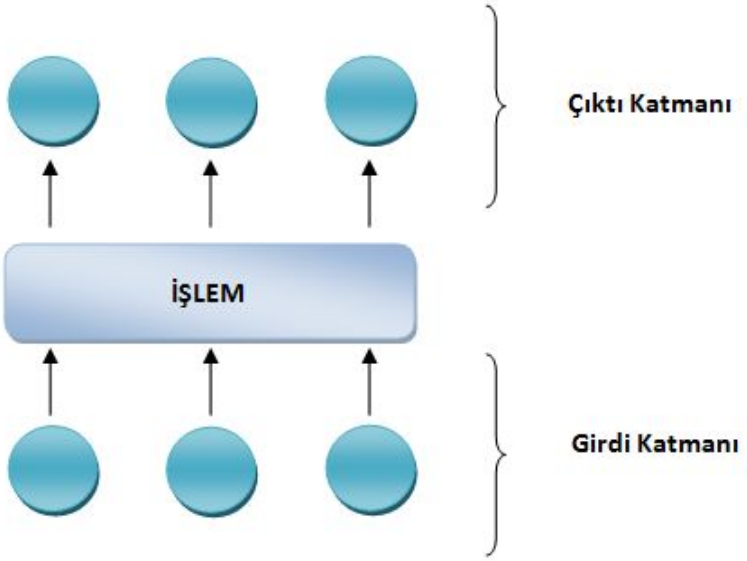
Tablo 2.1. Biyolojik Sinir Hücresi

Sinir Sistemi	**Yapay Sinir Ağı**
Nöron	İşlemci Eleman
Dentrit	Girdiler
Hücre Gövdesi	Transfer Fonksiyonu
Akson	Yapay Nöron Çıkışı
Sinaps	Ağırlıklar

Tablo 2.2. İstatistiksel Yöntemler İle Yapay Sinir Ağlarının Benzeşimi

İstatistik	**Yapay Sinir Ağı**
Model	Ağ
Tahmin	Öğrenme
Regresyon	Danışmalı Öğrenme
İnterpolasyon	Genelleştirme
Gözlem	Öğrenme Algoritması
Parametre	Ağ Parametreleri
Bağımsız Değişken	Giriş Verileri
Bağımlı Değişken	Çıkış Verileri
Sınır Regresyonu	Ağırlık Budama İşlemi

2.1.2.3. Yapay Sinir Ağlarında Genel Yapı



Şekil 2.7. Yapay Sinir Ağlarının Genel Yapısı

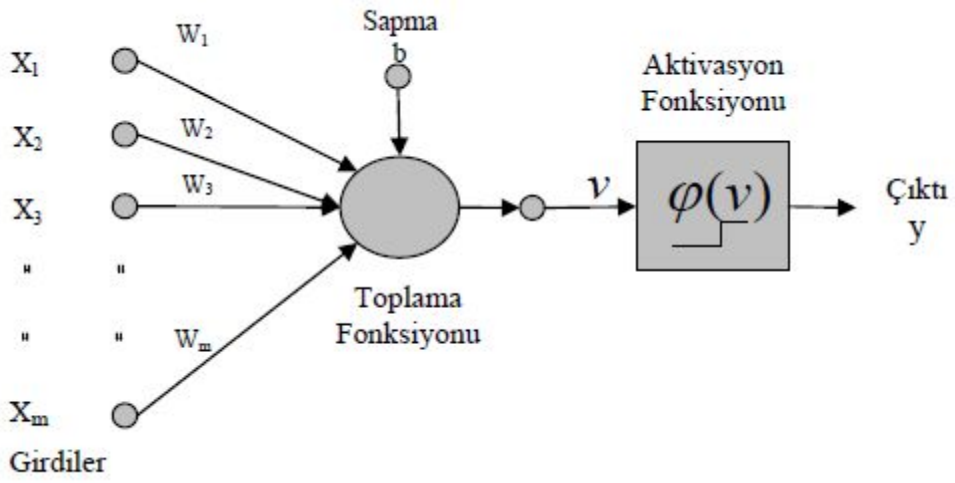
***Girdi Katmanı*:** Girdi katmanı en az bir girdi elemanının bulunduğu bölümdür. Girdi katmanında veriler herhangi bir işleme tabi tutulmadan girdileri ile aynı değerde çıktı üretirler.

***Çıktı Katmanı*:** Çıktı katmanı en az bir çıktıdan oluşur ve çıktı ağ yapısında bulunan fonksiyona bağlıdır. Bu birimlerde girdi katmanında olanın aksine işlem gerçekleştirilir ve birimler kendi çıktılarını üretirler.

***İşlem Katmanı*:** Genellikle "*Kara Kutu*" olarak adlandırılır. Girdi birimlerinin belirli işlemlere tabi tutulduğu bölgedir. Seçilen ağ yapısına göre işlem katmanının yapısı ve fonksiyonu da değişebilir. Tek bir katmandan oluşabileceği gibi birden fazla katmandan da oluşabilir [47].

2.1.2.4. Yapay Sinir Hücresi

Biyolojik sinir ağlarının yapı bileşenleri sinir hücreleridir benzer şekilde yapay sinir ağlarının da yapay sinir hücreleri bulunmaktadır (Şekil 2.8). YSA, insan sinir ağındaki gibi nöronlardan ve onlar arasındaki bağlantılardan oluşur. Bilgi, ağ tarafından bir öğrenme süreciyle çevreden elde edilir. Elde edilen bilgileri biriktirmek için sinaptik ağırlıklar olarak da bilinen hücreler arası bağlantı güçleri kullanılır [53].



Şekil 2.8 Basit Algılayıcı Modeli [53]

YSA'ya bilgi sayısal olarak dış dünyadan, diğer hücrelerden ya da kendi kendisinden gelebilir. Sinir hücresine bilgiler ağırlıklar yoluyla taşınırlar. Ağırlık değerleri bilginin önemini ifade eder, değişken ya da sabit değerler olabilirler, pozitif ya da negatif değerler alabilirler. Bir sinir hücresine gelen net bilgi yaygın olarak toplama fonksiyonu aracılığıyla hesaplanır. Her girdi değeri kendi ağırlığı ile çarpılır. Toplama fonksiyonu tüm girdiler için gelen bu değerleri toplayarak net hücre çıktısını hesaplar. Her hücre diğer hücrelerden bağımsız olarak bu net değerini hesaplar. Sapma (bias-b_k) değerinin aktivasyon fonksiyonuna giren değeri yükseltme ya da düşürme etkisi vardır. Aşağıdaki eşitlikte kullanılan x_j; gelen bilgileri, w_{kj} her girdi değerine ait ağırlıkları, b_k sapma değerini, v_k nöronun çıktı değerini ifade etmektedir [7].

$$V_k = \sum_{j=1}^{m} w_{kj} x_j + b_k \qquad (2.5)$$

Aşağıdaki eşitlikte görüldüğü gibi her sinir hücresinin net bilgisi eşik değerine sahip bir aktivasyon fonksiyondan geçirilerek gerçek bir çıktı oluşturulur. Genellikle kullanılan aktivasyon fonksiyonları eşik, sigmoid, hiperbolik tanjant vb.

fonksiyonlardır. Aktivasyon fonksiyonu (φ(.)) genellikle doğrusal olmayan bir fonksiyondur.

$$y_k = \varphi(v_k) \tag{2.6}$$

YSA herhangi bir konu ile ilgili veri setleriyle eğitilirken eğitim algoritmaları kullanırlar. Öğrenilmesi istenen olay için oluşturulan eğitim seti ağa sunulurken hedef çıktı değerleri de ağa sunulabilir. Sadece girdi seti ağa sunulabilir, sistemin kendi kendine öğrenmesi istenilebilir ya da her girdi seti için sistemin kendisinin bir çıktı üretmesi sağlanabilir. Üretilen çıktının doğru ya da yanlış olduğunu gösteren sinyal üretilerek, bu sinyale göre sistem eğitime devam edilebilir [54].

2.1.2.5. Yapay Sinir Ağının Temel Elemanları

Yapay sinir ağları aşağıdaki varsayımlar üzerine kurulmuştur: [55]

- Bilgi işleme nöron olarak isimlendiren basit elemanlarda gerçekleştirilir
- İşaretler nöronlar arasındaki bağlantılardan geçer
- Her bağlantı birçok işareti taşıyan bir ağırlığa sahiptir
- Her nöron kendi giriş değerine çıkış işaretini belirlemek için aktivasyon fonksiyonu uygular.

Yapay sinir ağları, birbirine bağlı çok sayıda işlemci elemanlardan oluşmuş, genellikle paralel işleyen yapılar olarak adlandırılabilir. Yapay sinir ağlarındaki işlem elemanları (düğümler) basit sinirler olarak adlandırılırlar. Bir yapay sinir ağı birbirine bağlantılı, çok sayıda düğümlerden oluşur.

Yapay sinir ağları insan beyni gibi öğrenme, hatırlama ve genelleme yeteneğine sahiptirler.

İnsan beyninde öğrenme 3 şekilde olur;

- Yeni aksonlar üreterek,
- Aksonların uyarılmasıyla,
- Mevcut aksonların güçlerini değiştirerek.

Her aksonun üzerinden geçen işaretleri değerlendirebilecek yetenekte olduğu savunulmaktadır. Aksonun bu özelliği, bir işaretin belli bir sinir için ne kadar önemli olduğunu göstermektedir [48].

2.1.2.5.1. Girişler

Girişler tarafından bir yapay sinir hücresine bir başka yapay sinir hücresinden veya dış dünyadan bilgi alışı yapılır. Bunlar ağın öğrenmesi istenen örnekler tarafından belirlenir.

2.1.2.5.2. Ağırlıklar

Ağırlıklar bir yapay sinir hücresinin girişleri tarafından alınan bilgilerin önemini ve hücre üzerinde etkisi gösteren uygun katsayılardır. Her bir giriş için bir ağırlık vardır. Bu ağırlığın büyük olması bu girişin önemli olduğu ya da ağırlığın küçük olması girişin önemsiz olduğunu göstermez. Bir ağırlığın değerinin sıfır olması o ağ için en önemli olay olabilir. Eksi değerler de yine girişin önemsiz olduğunu göstermez. Ağırlığın artı ve eksi olması girişin etkisinin pozitif ya da negatif olduğunu gösterir. Ağırlıklar değişken ya da sabit olabilirler [56].

2.1.2.5.3. Toplama İşlevi

Toplama işlevi bir yapay sinirdeki her bir giriş ile o girişe ait olan ağırlığın çarpılarak bu çarpımların toplanmasıdır.

$$Net\,Toplam = \sum_{i}^{n} x_i w_i \tag{2.7}$$

Ancak birçok uygulama aşağıdaki gibi eşik değeri olan Θ'da bu toplamaya katılmıştır.

$$Net\,Toplam = \sum_{i}^{n} x_i w_i + \theta \quad \text{ya da} \quad Net\,Toplam = \sum_{i}^{n} x_i w_i - \theta \tag{2.8}$$

Θ eşik değerinin girişlerden bağımsız olduğu için bütün girişlerin sıfır olması durumunda çıkış değerinin sıfır değil de eşik değerine eşit olduğu görülür ki bu da, belirtilen şartlar altında nöron çıkışının sıfır olması zorunluluğunu ortadan kaldırır. Eşik değerinin kullanımı, toplama fonksiyonuna +1 ya da -1 değerine sahip sabit bir girişin Θ ağırlığına sahip bir bağlantı ile eklendiği şeklinde yorumlanır [48].

Ayrıca her model ve her uygulama için bu toplama fonksiyonun kullanılması şart değildir. Bazı modeller, kullanılacak toplama fonksiyonunu kendileri belirler. Çoğu zaman daha karmaşık olan değişik toplama fonksiyonları kullanılır. Bunlar Tablo 2.3'de gösterilmiştir. Bazı durumlarda girişlerin değeri önemli olurken, bazılarında sayısı önemli olabilir. Bir problem için en uygun toplama fonksiyonunu belirlemek için bir formül geliştirilememiştir. Bu yüzden en uygun toplama fonksiyonunun bulunması deneme yanılma yoluyla belirlenir. Ayrıca aynı problem için kullanılan yapay sinir hücrelerinden hepsi aynı toplama fonksiyonunu kullanabileceği gibi her biri için farklı toplama fonksiyonu kullanılabilir [48].

Tablo 2.3 Toplama Fonksiyonu Örnekleri *Toplama İşlevi*

Toplama İşlevi	**Açıklama**
Çarpım $Net\ Girdi = \prod_i x_i w_i$	Girişler ve ağırlıklar çarpılır sonra bu değerler de birbiriyle çarpılır.
Maksimum (En çok) $Net\ Girdi = Max(x_i w_i), i = 1, 2, ..., N$	N adet giriş ve ağırlık birbiriyle çarpıldıktan sonra en büyüğü net girdi olarak kabul edilir.
Maksimum (En az) $Net\ Girdi = Min(x_i w_i), i = 1, 2, ..., N$	N adet giriş ve ağırlık birbiriyle çarpıldıktan sonra en küçüğü net girdi olarak kabul edilir.
Çoğunluk $Net\ Girdi = \sum_i \text{sgn}(x_i w_i)$	N adet giriş ve ağırlık birbiriyle çarpıldıktan sonra pozitif ve negatif olanların sayısı bulunur. Bunlardan büyük olan net girdi olarak alınır.
Kümülâtif Toplam $Net\ Girdi = Net(eski) + \sum_i (x_i w_i)$	Hücreye gelen bilgiler ağırlıklı olarak toplanır ve daha önce gelen bilgilere eklenerek hücrenin net girdisi bulunur.

2.1.2.5.4. Etkinlik İşlevi (Aktivasyon Fonksiyonu)

Yapay nöronun davranışını belirleyen önemli bir etken aktivasyon fonksiyonudur. Buna aynı zamanda "öğrenme eğrileri" de denir. Aktivasyon fonksiyonu hücreye gelen net girdiyi, diğer bir deyişle toplama fonksiyonunu işleyerek bu hücreye gelen girişlere karşılık olan çıkışı belirler.

Aktivasyon fonksiyonu da yapay sinir ağlarının farklı modelleri için farklı olabilir. En uygun aktivasyon fonksiyonunu belirlemek için geliştirilmiş bir fonksiyon yoktur. Toplama fonksiyonuna benzer şekilde hücrelerin hepsi için aynı aktivasyon

fonksiyonu kullanma zorunluluğu yoktur. Bazıları aynı aktivasyon fonksiyonunu kullanırken bazıları kullanmayabilir [48].

Bazı modeller için özellikle de Çok Katmanlı Algılayıcı model için bu fonksiyon türevi alınabilir ve sürekli olmalıdır. Yapay sinir ağlarının kullanım amacına göre tek veya çift yönlü aktivasyon fonksiyonu kullanılabilir.

Doğrusal olmayan fonksiyonların kullanılması yapay sinir ağlarının çok karmaşık ve farklı problemlere uygulanmasını sağlamıştır. En çok kullanılan aktivasyon fonksiyonları şunlardır [23,26,29,48,53]:

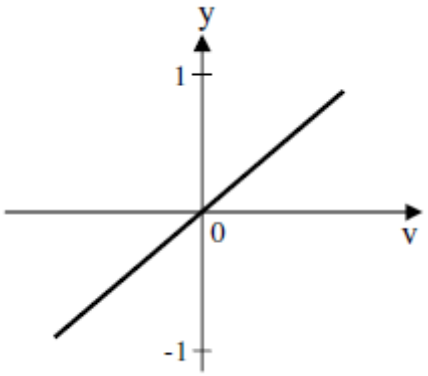
2.1.2.5.4.1. Doğrusal Fonksiyon

Doğrusal fonksiyon, hücreye gelen girişleri olduğu gibi çıkışa verir. Fonksiyonun şekli Şekil 2.9'da verilmiştir. Çoğunlukla ADALINE olarak bilinen doğrusal modelde, klasik işaret işleme ve regresyon analizinde kullanılır. Denklemi;

$$v = \sum_{i}^{n} x_i w_i \quad \text{veya} \quad v = \sum_{i}^{n} x_i w_i + \theta \quad \text{olmak üzere;}$$

$$y = F(v) = Av \tag{2.9}$$

Formüldeki A sabit katsayısıdır.



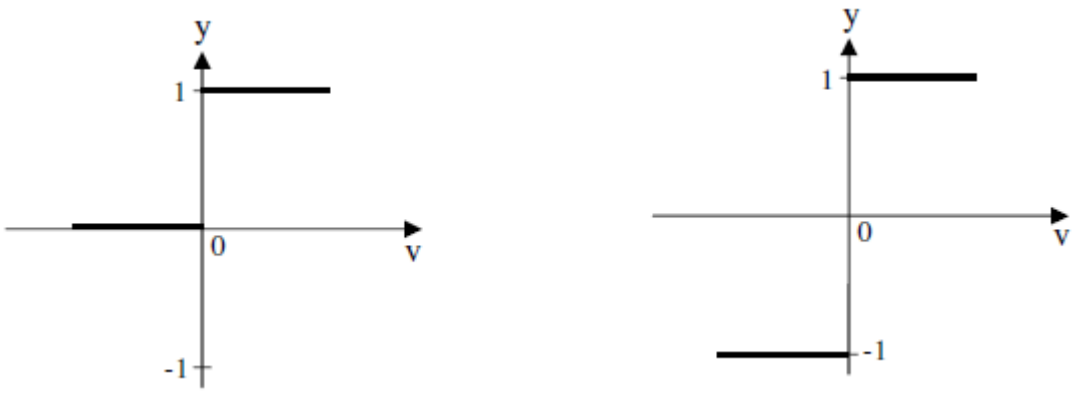
Şekil 2.9. Doğrusal veya Lineer Fonksiyon

2.1.2.5.4.2. Basamak Fonksiyonu

Basamak fonksiyonu tek veya çift kutuplu olabilir. Bu fonksiyonların şekli Şekil 2.10'de, matematiksel ifadeleri de aşağıda verilmiştir. Perceptron (Basit Algılayıcı Model) olarak bilinen yapay sinir hücresi aktivasyon fonksiyonu olarak bu fonksiyonu kullanır.

$$y = F(v) = \begin{Bmatrix} 1 & v \geq 0 \\ 0 & v < 0 \end{Bmatrix} \tag{2.10}$$

$$y = F(v) = \begin{Bmatrix} +1 & v \geq 0 \\ -0 & v < 0 \end{Bmatrix}$$



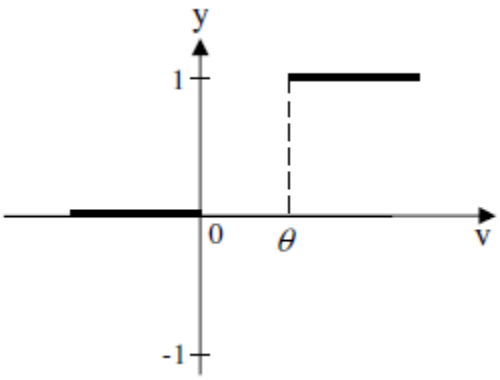
(a) Tek kutuplu (b) Çift Kutuplu

Şekil 2.10. Basamak Fonksiyonları

2.1.2.5.4.3. Kutuplamalı Basamak Fonksiyonu

Kutuplama değeri tek kutuplu ve çift kutuplu basamak fonksiyonunun her ikisine de eklenebilir. Aktivasyon fonksiyonu eşik değeri olan Θ'yı aştığı zaman nöron aktif olur. Tek kutuplu basamak fonksiyonu için denklem aşağıdaki eşitlik ve Şekil 2.11 - Şekil 2.12'da verilmiştir.

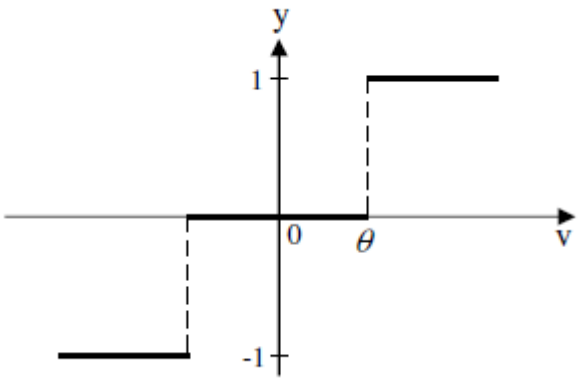
$$y = F(v) = \begin{Bmatrix} 1 & w.x \geq 0 \\ 0 & w.x < 0 \end{Bmatrix} \tag{2.11}$$



Şekil 2.11. Tek kutuplamalı Basamak Fonksiyonu

Çift kutuplu basamak fonksiyonu ise;

$$y = F(v) = \begin{Bmatrix} +1 & w.x \geq 0 \\ -1 & w.x < 0 \end{Bmatrix} \qquad (2.12)$$



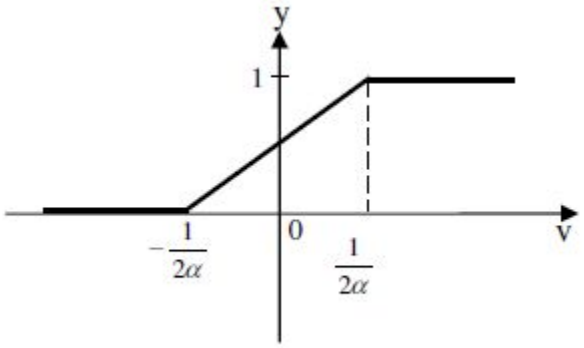
Şekil 2.12. Çift Kutuplamalı Basamak Fonksiyonu

2.1.2.5.4.4. Parçalı Doğrusal Fonksiyon

Bu fonksiyon, küçük aktivasyon potansiyeli için, α kazancı olan bir doğrusal toplayıcı (Adaline) olarak çalışır. Büyük aktivasyon potansiyeli için, nöron doyuma ulaşır ve çıkış işareti 1 olur. Büyük kazançlar için, $\alpha \longrightarrow \infty$ iken, parçalı doğrusal

fonksiyon basamak fonksiyonu gibi davranır. Aşağıdaki denklem fonksiyonu, Şekil 2.13'de grafiği gösterilmiştir.

$$y = F(v) \quad \begin{cases} 0 & v \le 1/2\alpha \\ \alpha v + 1/2 & |v| < 1/2\alpha \\ 1 & v \ge 1/2\alpha \end{cases} \tag{2.13}$$

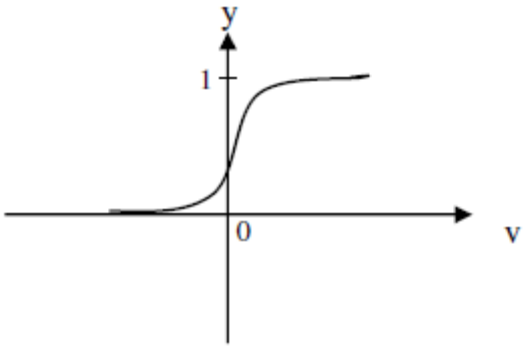


Şekil 2.13. Parçalı Doğrusal Fonksiyon

2.1.2.5.4.5. Sigmoid Tipi Fonksiyon

Uygulamalarda en çok kullanılan aktivasyon fonksiyonlANNdan biridir. Fonksiyonun formülü Denklem 1.9'da, şekli ise Şekil 2.14'de gösterilmiştir. Fonksiyonun en aktif bölgesi 0,2 ile 0,8 arasındadır. Tek kutuplu fonksiyon olarak da adlandırılır.

$$y = F(v) \quad \frac{1}{1+e^{-v}} = \frac{1}{2}\left[\tanh(v/2) + 1\right] \tag{2.14}$$

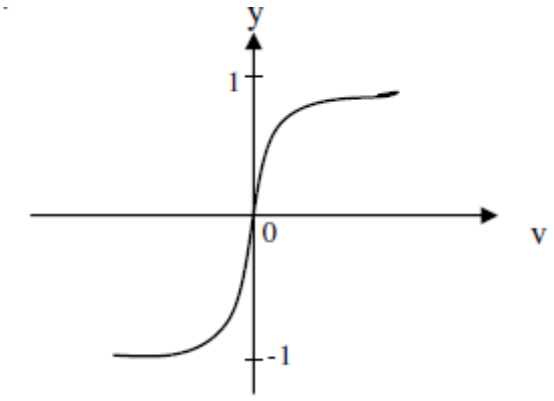


Şekil 2.14. Sigmoid Tipli Fonksiyon

2.1.2.5.4.6. Tanjant Hiperbolik Tipli Fonksiyon

Uygulamalarda çok kullanılan aktivasyon fonksiyonlarından biri de Tanjant Hiperbolik fonksiyondur. Bu fonksiyon çift kutuplu fonksiyon olarak da bilinir. Giriş uzayının genişletilmesinde etkin bir şekilde kullanılan bir fonksiyondur. Fonksiyonun şekli Şekil 2.15'de formülü ise aşağıdaki gösterildiği gibidir.

$$y = \frac{1 - e^{-2v}}{1 + e^{2v}} = \tanh(\beta v) \tag{2.15}$$

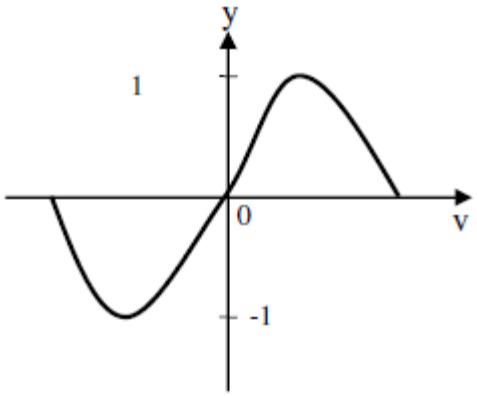


Şekil 2.15. Tanjant Hiperbolik Tipli Fonksiyon

2.1.2.5.4.7. Sinüs Tipli Fonksiyon

Öğrenilmesi düşünülen olaylar sinüs fonksiyonuna uygun dağılım gösteriyorsa bu gibi durumlarda aktivasyon fonksiyonu olarak sinüs fonksiyonu kullanılır. Fonksiyonun şekli Şekil 2.16'de ve formülü aşağıda verilmiştir.

$$y = f(v) = Sin(v) \tag{2.16}$$



Şekil 2.16. Sinüs Tipli Fonksiyon

Bir yapay sinir ağının bu 5 temel elemanı dışında zaman zaman ihtiyaç duyulduğunda kullanılan bir elemanı daha vardır. Bu eleman ölçekleme ve sınırlama olarak adlandırılır.

2.1.2.5.5. Çıkış İşlevi

Çıkıs y = f(v), aktivasyon fonksiyonunun sonucunun dış dünyaya veya diğer sinirlere gönderilmesidir. Bu sinirin çıkısı kendine ve kendinden sonra gelen bir ya da daha fazla sayıda sinire giriş olabilir.

2.1.2.6. Ağ Girişlerinin Hesaplanması İçin Matris Çarpma Metodu

Sinir ağlarının bağlantılı ağırlıklarını W = (w_{ij}) matrisinde kaydedelim. Bu halde, Y_j' nin ağ girdisi x = (x_1,..., x_i, ...,x_n) vektörünün $w_{.j}$ ağırlık matrisinin $_j$. kolonu ile (eğer $_j$. elemanda sapma yoksa) basit bir çarpılması işlemi ile gerçekleştirilir :

$$y\_in = x.w_{.j} \quad \sum_{i=1}^{1} x_i.w_{ij} \tag{2.17}$$

Sapma

Sapma değeri, x vektörüne $x_0 = 1$ değeri eklenerek oluşturulabilir.

$$x = (1, x_1, ..., x_i, ..., x_n) \tag{2.18}$$

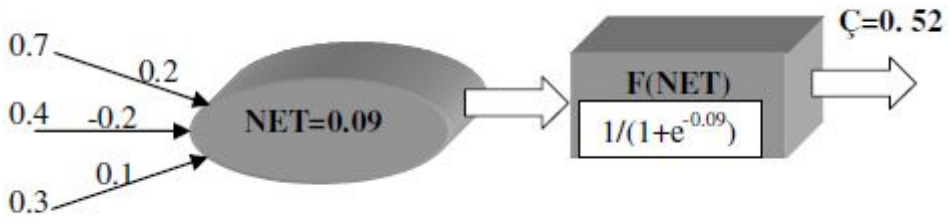
Sapma aynı diğer ağırlıklar gibi davramr, yani $w_{0j} = b_j$ dir. Y_j birimi için ağ girdisi şu şekilde hesaplanır:

$$y\_in_j = x.w_{.j} \quad \sum_{i=1}^{n} x_i.w_{ij} \quad w_{0j} + \sum_{1}^{n} x_i.w_{ij} \quad b_j + \sum_{i=1}^{n} x_i.w_{ij} \tag{2.19}$$

Bir YSA düğümünün görevi, girişindeki sayıları kendi ağırlık değerleri ile çarpıp, sonra da bu çarpımları toplayıp, toplamı bir yumuşatma fonksiyonundan (genelde sigmoid $f(x) = \frac{1}{1+e^{-x}}$ veya *tanh*) geçirdikten sonra çıkışa vermektır. Ancak giriş ve çıkış katmanındaki nöronlar bu kuralın dışındadır. Giriş katmanındaki nöronlar ise sadece kendi girişlerindeki verilerin uygun ağırlıklarla çarpılmış durumlarını toplayıp saklarlar. Bu işleme *ilerleme* denir [58,60].

2.1.2.7. Yapay Sinir Hücresinin Çalışma Prensibi

Şekil 2.17'da girişleri ve ağırlıkları verilmiş olan bir yapay sinir hücresinin çalışması şöyledir:



Şekil 2.17. Yapay Sinir Ağının Çalışma Örneği

Hücreye gelen net girdi, ağırlıklarla girişler çarpılarak aşağıdaki gibi hesaplanır.

NET Girdi = 0.7 * 0.2 + 0.4 *(-0.2) + 0.3*0.1

NET Girdi = 0.14 - 0.08 + 0.03

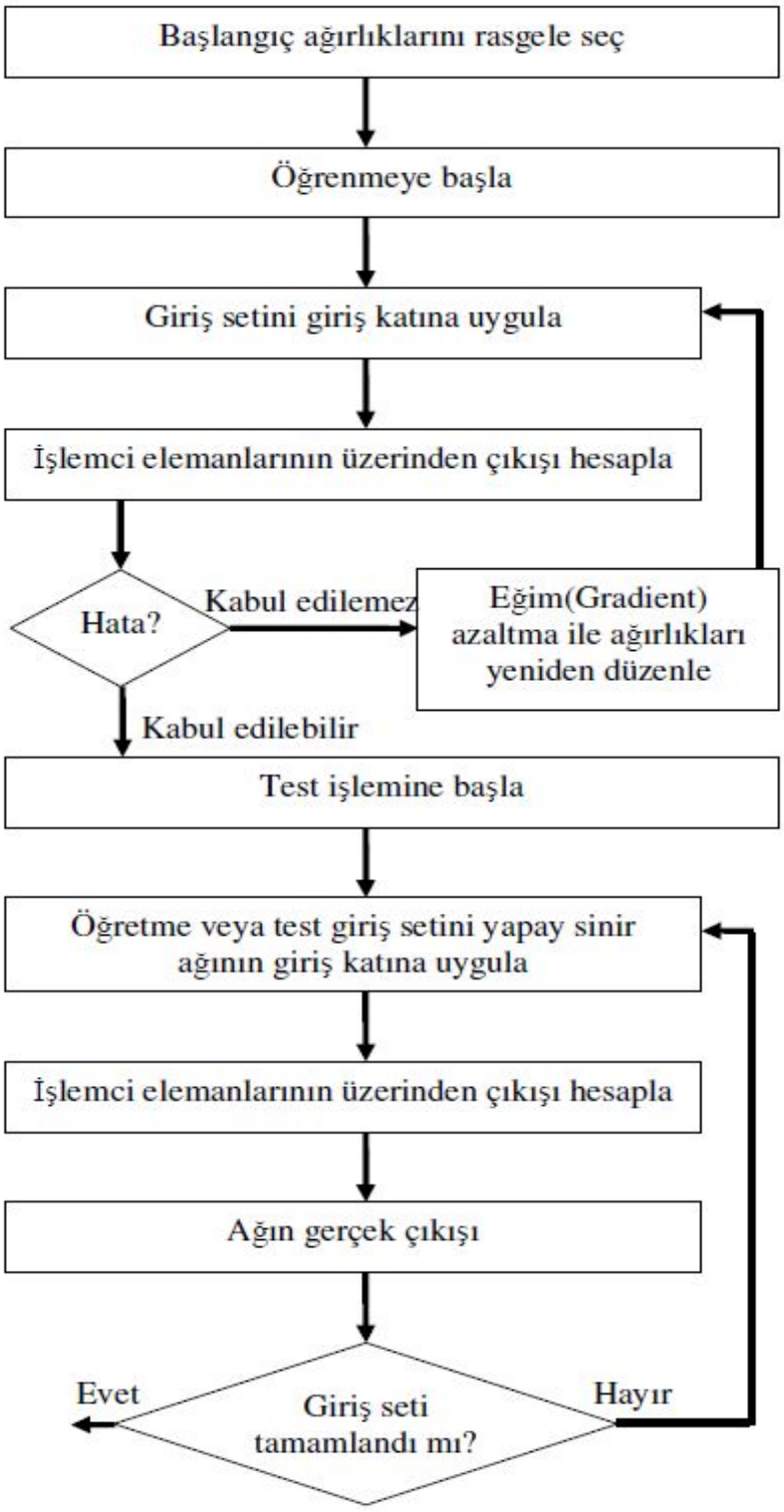
NET Girdi = 0.09

Hücrenin sigmoid tipli aktivasyon fonksiyonuna göre çıkışı y=f(v) şöyledir;

$y = f(v) = 1 / (1+e^{-0.09})$

y = 0.52

Sonuçta verilen girdilere karşılık yukardaki işlemler sonucunda y çıkış değeri bulunur.



Şekil 2.18 YSA' de Kullanılan Geri Yayılımlı Öğrenme Algoritması

2.1.3. Lojistik Regresyon Analizi

Lojistik regresyon; bağımlı değişkeninin kategorik ve ikili, üçlü ve çoklu kategorilerde gözlendiği durumlarda bağımsız değişkenlerle neden sonuç ilişkisini belirlemede yararlanılan bir yöntemdir. Bağımsız değişkenlere göre bağımlı değişkeninin beklenen değerlerinin olasılık olarak elde edildiği bir regresyon yöntemidir. Basit ve çoklu regresyon analizleri bağımlı değişken ile bağımsız değişken ya da değişkenler arasındaki matematiksel bağıntıyı analiz etmekte kullanılmaktadır. Bu yöntemlerin uygulanabileceği veri setlerinde bağımlı değişkenin normal dağılım göstermesi, bağımsız değişkenlerinde normal dağılım gösteren toplum ya da toplumlardan çekilmiş olması ve hata varyansının $\varepsilon \cong N(0,\sigma^2)$ parametreli normal dağılım göstermesi gerekmektedir. Bu ve benzeri koşulların yerine getirilmediği veri setlerinde basit ya da çoklu regresyon analizleri uygulanamaz. Lojistik regresyon analizi, sınıflama ve atama işlemi yapmaya yardımcı olan bir regresyon yöntemidir. Normal dağılım varsayımı, süreklilik varsayımı ön koşulu yoktur. Bağımlı değişken üzerinde bağımsız değişkenlerin etkileri olasılık olarak elde edilerek risk faktörlerinin olasılık olarak belirlenmesi sağlanır [11,18,19,25].

Araştırıcılar üzerinde çalıştıkları konuda çok etken olması durumunda etkenlerin tek tek bağımlı değişken üzerine etkisi yanında, bunların birlikte etkisini de öğrenmek ya da incelemek istemektedirler. Birlikte etkinin incelenmesinde kullanılan değişik istatistik yöntemler bulunmaktadır. Örneğin, bağımlı değişkenin sürekli, bağımsız değişkenlerin kesikli olması durumunda varyans analizi, hepsinin kesikli olması durumunda "log-linear model"ler, hepsinin sürekli olması durumunda regresyon analizi gibi. Tıp alanındaki araştırmalarda çoğu zaman bağımlı ve bağımsız değişkenlerin tür ve yapıları yukarıda belirtilenlere benzemez, sürekli ve kesikli karışımı bağımsız

değişkenlerle karşılaşılır. Üzerinde en çok durulan ve araştırıcı için önemli olan diğer bir konu da etken veya etkenlerle hastalık arasındaki ilişkinin risk yönünden incelenmesidir. Bu tip incelemelerde ağırlıklı olarak LRA kullanılmaktadır [20,41].

Lojistik regresyon modelinin kullanımına ilişkin ilk çalışmalar Berkson (1944) tarafından yapılmış ve model Finney (1972) tarafından biyolojik deneylerde probit analize bir alternatif olarak önerilmiştir . Son yıllarda yoğun bir şekilde kullanılan LRA, gözlemlerin gruplara atanmasında sık kullanılan üç yöntemden (diğerleri kümeleme analizi ve ayırma analizi) birisidir. LRA da grup sayısı bilinmekte, mevcut veriler kullanılarak bir ayrımsama modeli elde edilmekte ve kurulan bu model yardımıyla veri kümesine eklenen yeni gözlemlerin gruplara atanması mümkün olabilmektedir [42].

Doğrusal regresyon analizinde bağımlı değişkenin değeri tahmin edilirken, LRA da bağımlı değişkenin alacağı değerlerden birinin gerçekleşme olasılığı tahmin edilir. Bu olasılık değerinin tahmininde aşağıdaki model kullanılmaktadır.

$$\pi(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}} \tag{2.20}$$

Lojistik regresyon modeliyle tahmin yapılacağı genel olarak kullanılan yöntem en çok olabilirlik metodudur. Genel anlamda en çok olabilirlik metodu, gözlenen veri kümesini elde etmenin olasılığını maksimum yapan bilinmeyen parametrelerin değerlerini tahmin etmede kullanılır. Bu metodu uygulayabilmek için en çok olabilirlik fonksiyonunun oluşturulması gerekmekte ve ilgili parametrelerin en çok olabilirlik tahmincileri, fonksiyonu maksimum yapacak değerleri bulacak şekilde seçilmelidir [6].

LRA' da gözlenen değerlerle tahmin edilen değerler aşağıdaki ifadeyle karşılaştırılır.

$$D = -2\ln\left[\frac{Su\ andaki\ Modelin\ olabilirligi}{Doymus\ Modelin\ olabilirligi}\right] \tag{2.21}$$

Modelde bulunan herhangi bir bağımsız değişkenin önemliliğine karar vermek için denklemde o bağımsız değişkenin bulunduğu ve bulunmadığı durumlardaki *D* değerleri, *G* istatistiği kullanılarak karşılaştırılırlar. *G* istatistiği *p* serbestlik derecesiyle ki-kare dağılımı gösterecektir.

G = D(Değişkensiz model için) - D(Değişkenli model için)

$$D = -2\ln\left[\frac{Degiskensiz\ Modelin\ olabilirligi}{Degiskenli\ Modelin\ olabilirligi}\right] \qquad (2.22)$$

Katsayıların önemlilikleri test edildikten sonra katsayıların yorumlanması odds oranları kullanılarak yapılmaktadır. LRA' nın kullanım amaçlarından en önemlisi tıp biliminde sıklıkla karşılaşılan bağımlı değişkenin iki ya da daha çok düzey içerdiği, bağımsız değişkenlerin ise hem kesikli hem de sürekli olabildiği durumlarda verilerin ait oldukları gruplara en doğru şekilde atayacak ve hastalıklara ilişkin risk faktörlerini belirleyebilecek modeli kurmaktır. Bunun yanında lojistik regresyon, bağımlı değişkenin tahminini olasılık olarak hesaplayarak olasılık kurallarına uygun sınıflama işlemi yapma olanağı vermektedir [20,23,43].

2.1.3.1. Lojistik Sınıflandırma ve Lojistik Regresyon Modeli

Bağımlı değişkenin 0.1 değerlerine karşılık gelen G1 ve G2 grupları $x_1,x_2,...,x_p$ bağımsız değişkenlerine dayanılarak sınıflandırılmak istensin. Gruplardaki birey sayısı sıra ile n1 ve n2 olduğunda, N=n1+n2 gözleme dayalı sınıflandırma kuralının oluşumu $f_s(x_1,x_2,...,x_p)$ şeklindeki olasılık fonksiyonunun fonksiyonel yapısına ilişkin varsayımlara dayanır. Fonksiyon yapısı için üç tür varsayım söz konusudur [13,14,44].

i. Çok değişkenli normal dağılım fonksiyonu

ii. Lojistik sınıflandırma fonksiyonu

iii. Dağılımdan bağımsız kernel sınıflandırma fonksiyonu

Lojistik sınıflama fonksiyonu söz konusu olduğunda X_0=l iken $f_s(x_1,x_2,...,x_p)$, G_s(s=l,2) grubunun olasılık yoğunluk fonksiyonu olarak tanımlanır. Lojistik varsayım, β'=(βo, βı...., β_p) için,

$$\frac{f_1(x)}{f_2(x)} = \exp(\beta' X') \; ya \; da \ln\left(\frac{f_1(x)}{f_2(x)}\right) = \beta' X' \tag{2.23}$$

Şeklinde tanımlanmaktadır. Bu son eşitlik log-olabilirlik oram olup x ‘ ler doğrusaldır. Lojistik varsayım bilinmeyen P parametrelerini içermektedir. Her bir gözlem için X koşulu altında gruplardan birine atanma olasılığı olarak tanımlanan sonsal olasılıkları hesaplamak için P tahminleri gerekmektedir. Bunun için lojistik varsayım altında örneklemin olabilirlik fonksiyonu belirlenmelidir. Karışık örneklemede gözlemler (X,G) bileşik dağılımından örneklenmekte yani gözlemler hangi gruptan olduğu bilinmeksizin rasgele seçilmektedir. Buradan G grup üyeliğini gösteren değişken olup iki grup olduğunda G_1 ve G_2 şeklinde gösterilmektedir [44].

Koşullu örneklemede G’ nin x koşulu altında dağılımı incelenmektedir. Biyolojik deneylerin analizinde çok sık kullanılan bu örnekleme türüne ilişkin olabilirlik fonksiyonu diğer örnekleme türlerinin olabilirlik fonksiyonuna temel teşkil etmektedir. Ayrı örnekleme de ise x’ in G koşulu altında dağılımından örnekleme yapılmaktadır. Anderson (1972, 1982) tarafından detaylı olarak incelenen bu örnekleme türünün uygulaması zor olup geçmişe yönelik (retrospective) çalışmalarda uygulanmaktadır [41].

Regresyon problemlerinde anahtar değer, verilen bir bağımsız değişken değerine bağlı olarak bağımlı değişkenin ortalama değerini bulmaktır. Bu değer koşullu ortalama olarak adlandırılır ve E(Y \ x) ile gösterilir. Burada y’nin bağımlı değişkeni, x ’in ise

bağımsız değişkeni gösterdiği varsayılsın. E(Y\ x) ifadesi " x değeri verildiğinde, y' nin beklenen değeri" ni göstermektedir. Doğrusal regresyon analizinde, koşullu ortalamanın, x 'in doğrusal bir denklemi olduğu varsayılır.

$$E(Y \setminus x) = \beta_0 + \beta_1 x \tag{2.24}$$

Yukarıdaki bu ifadeden, x'in aralığının -∞ ve +∞ arasında değişmesinden dolayı, E(Y\ x)'in mümkün olan her değeri alabileceği görülmektedir. Bağımlı değişken ikili olduğu zaman koşullu ortalama, sıfırla bir arasında değişmek zorundadır [44].

[0 < E(Y\ x) ≤ 1]. x 'deki her birim değişme sonucunda E(Y\ x)'de oluşan değişiklik, koşullu ortalama 0'a ya da 1'e yaklaştıkça ilerleyerek az olur.

İki düzey içeren bir bağımlı değişkenin analizinde kullanılmak üzere önerilen birçok dağılım fonksiyonu bulunmuştur [6]. Lojistik dağılım kullanıldığında gösterimi kolaylaştırmak için, x bilindiğinde Y' nin koşullu ortalamasını göstermek için π(x) = E(Y\ x) ifadesi kullanılmaktadır. Kullanılacak lojistik regresyon modelinin açık şekli aşağıdaki gibidir [6,16,17,44].

$$\pi(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}} \tag{2.25}$$

Lojistik regresyon çalışmasına merkez olacak π(x)'in bir transformasyonu yukarıda bahsedildiği gibi lojit transformasyondur. Bu transformasyon π (x)cinsinden tanımlanırsa:

$$g(x) = \ln\left[\frac{\pi(x)}{1-\pi(x)}\right] = \beta_{0+}\beta_1 x \tag{2.26}$$

Lojit g(x) parametreleri bakımından doğrusal ve x'in aldığı değerlere bağlı olarak -∞ ve +∞ arasında değişmektedir [25].

2.2. PROSTAT KANSERİ

2.2.1. Prostat Kanseri Hakkında Genel Bilgi

Prostat kanseri erkeklerde en yaygın görülen kanser türüdür ve İngiltere ve Galler'de yeni tanı konulan kanser olgularının %25'ini teşkil eder. Prostat bezesinin kötü huylu tümörü demek olan prostat kanseri, Robert Koch Enstitüsünün en yeni tahminlerine göre Almanya'da erkeklerde en sık rastlanan kanser hastalığıdır. 2000 yılında bu hastalığa ilk defa yakalananların sayısı yaklaşık 40.000 kişiydi, yaklaşık 11.000 kişi de bu hastalık yüzünden hayatını kaybetti.

Prostat kanserinin nasıl ve neden ortaya çıktığı sorusu bugüne kadar tam olarak cevaplandırabilmiş değildir. İlke olarak yaş ilerledikçe prostat kanseri riski de artmaktadır. 50 yaşın altındaki erkekler seyrek olarak hastalanır, en yüksek hastalanma oranları 70 yaşın üstündeki erkeklerde görülmektedir.

Kişinin kendi babası, erkek kardeşleri veya büyük babaları prostat kanserine yakalanmış ise daha yüksek bir risk var demektir. Bu nedenle kalıtım yoluyla geçme veya en azından genlerden kaynaklanan bir hastalanma eğiliminin mevcut olduğu tahmin edilmektedir.

Öte yandan prostat kanserine yakalanmış olan erkeklerin % 80'inde herhangi bir akrabada prostat kanseri görülmemiştir. Beslenme biçiminin de bir rol oynadığı yönünde işaretler vardır.

2.2.2. Prostat Nedir?

Prostat erkeğin en önemli cinsel salgı bezesidir. Prostat bezesi, mesanenin (idrar torbasının) idrar yoluna bağlandığı yerde, idrar borusunu çepeçevre saran bir organdır. Birçok erkekte yaşın ilerlemesiyle prostatın idrar yolunun hemen yanındaki iç

bölümünde iyi huylu bir tümör oluşur. Prostat 60 yaşındaki bir erkekte ortalama 30 ila 40 ml büyüklüktedir. Bu, yaklaşık iki ceviz büyüklüğü demektir. Bu iyi huylu ve prostat hiper plazisi (aşırı büyüme) adı verilen büyüme idrar yolunu daraltarak idrar yapmada zorluklara sebep olabilir.

Buna karşılık prostat kanseri çoğunlukla prostatın makattan (göden bağırsağı veya rektum) parmakla erişilebilen dış bölümünde oluşur. Prostat karsinomlarının yaklaşık % 80'i prostat bezesinin bu dış bölümünde ortaya çıkar. Prostat kanserinin prostat bezesinin iç bölümünde veya pubis kemiğinin arkasında yer alan ön kısmında oluşması enderdir.

2.2.3. Teşhis

Şüpheli bir durum ortaya çıktıktan sonra prostat kanseri teşhisi koyulabilmesi için bir dizi muayene daha yapılması gerekir. Bunlar:

2.2.3.1. Prostatın Parmakla Muayenesi

Prostat bezesi makattan (göden bağırsağı veya rektum) parmakla yoklanarak muayene edilebilir. Normal olarak prostat (el gergin tutulduğunda insanın elinin ayası gibi) yumuşak ve esnektir. Prostat kanseri ise çoğunlukla kemik gibi serttir. Bu sayede tecrübeli bir doktor prostat kanserini parmakla yoklayarak bulabilir ve düğümün büyüklüğünü tahmin edebilir. Eğer parmakla yoklandığında bir düğüm hissediliyorsa tümör çoğunlukla prostat kapsülüne de yerleşmiş veya prostat kapsülünü delip geçmiş demektir.

2.2.3.2. Prostata Özgü Antijenin Belirlenmesi (PSA)

Prostat meni sıvısının büyük bölümünü üreten organdır. Bu meni sıvısı içinde meni sıvısını akışkan hale getirmeye yarayan bir protein vardır. Söz konusu proteine prostata özgü antijen denir, bunun kısaltması da PSA'dır. PSA normal şartlar altında da

kana karışabilir ve kandaki miktarı PSA testi ile ölçülebilir. Yaş ilerledikçe prostat büyüdüğünden kandaki PSA miktarı da yükselir. Her gram iyi huylu prostat dokusu başına aşağı yukarı 0,3 ng/ml antijen bulunur. Buna karşılık prostat kanseri her gram doku başına yaklaşık 10 kere daha fazla PSA üretir. Bu yüzden PSA prostat kanserinin teşhis edilebilmesi için, yani tümör belirteci (tümör markeri) olarak kullanılmaya, uygundur.

Tablo 2.4 Yaşa Özgü Normal Serum PSA Değerleri

YAŞ	**PSA**
40-49	0-2,5 ng/ml
50-59	0-3,5 ng/ml
60-69	0-4,5 ng/ml
70-79	0-6,5 ng/ml

2.2.3.3. Makattan (rektum) Ultrason Muayenesi (TRUS)

Ultrason muayenesi ile vücudun iç organlarını, şua yükü yüklemeden görüntülemek mümkün olmaktadır. Prostatın değerlendirilmesi için makata (rektum veya göden bağırsağı da denir) ufak bir çubuk biçiminde bir sonda sokulur. Böylece prostatın yalnızca büyüklüğü ve biçimi değil, prostatın doku yapısı ve eğer mevcutsa, kötü huylu doku değişikliklerinin olup olmadığı da görülebilir. Ancak prostat kanseri için tipik olan doku değişiklikleri çoğu kereler ancak büyükçe tümör düğümlerinde görülebilmektedir. Bunun ötesinde, ultrason ile görüntüleme yöntemi prostattan hedefli olarak doku alabilmek için de gayet yararlıdır [62, 63, 64].

3. GEREÇ ve YÖNTEM

Uygulama verisi olarak, Tokat Gaziosmanpaşa Üniversitesi Tıp Fakültesi Hastanesinde 01.01.2005 tarihi ve 31.05.2011 tarihleri arasında Üroloji Polikliniği'ne başvuran hastaların sonuçları alınmıştır. Veritabanı yönetim sistemi Oracle 10 GR2 olan hastane veritabanından Structured Query Language (SQL) veritabanı sorgu dili kullanılarak gerekli bilgiler çekilmiştir. Belirtilen tarihler arasında Üroloji Polikliniği'ne başvurup muayenesi ve tetkikleri yapılıp kesin tanı olarak "C61", 'Prostat malign neoplazmı (prostat kanseri)' tanısı konulan hastaların muayene bilgileri incelenmiştir. İncelenen muayene bilgileri içerisinde prostat kanserini teşhis için kullanılan dört faktör incelenmiştir. Bu faktörler;

i. Yaş,
ii. Prostatın parmakla muayenesi (rektal tuşe),
iii. Genetik yatkınlık bilgisi ve
iv. PSA düzeyidir.

İncelenen hastalardan kesin tanısı prostat kanseri olup muayene notları içerisinde yukarda sıralanan faktörlerin hepsinin bulunduğu hastalar seçilmiştir. Bu şekilde yapılan tarama sonucunda 118 hasta verisi tespit edilmiştir. Aynı şekilde muayene notları içerisinde yaş, rektal tuşe, genetik yatkınlık ve PSA tetkik sonucu bulunan ama kesin tanısı "C61", 'Prostat malign neoplazmı (prostat kanseri)' olmayan 118 hasta daha seçilerek toplam 236 hastalık veri setine ulaşılmıştır.

Veri setinin WEKA programına yüklenebilmesi için arff formatında olması gerekmektedir. WEKA programı 'Arff Viewer' seçeneği ile ".cvs" uzantılı dosyaları açabilmektedir. Bu nedenle, öncelikle MS Excel ortamda edinilen veriler ".xls"

formatından ".cvs" (comma-separeted variables) (virgülle ayrılmış veriler) formatına dönüştürülmüştür. Bu format türündeki veriler Ek-1'de sunulmuştur

Veri madenciliği algoritmalarından Lojistik Regresyon Analizi (LRA), Yapay Sinir Ağları (YSA) Ve Sınıflandırma Ve Regresyon Ağaçları (C&RT) yöntemlerinin karşılaştırılması için prostat kanseri verileri kullanılmıştır.

3.1. C&RT ALGORİTMASI

Bilimsel çalışmalardan elde edilen verilerin analizinde sınıflama ve regresyon ağaçları, kümeleme, ayırma ve lojistik regresyon analizlerini içeren sınıflama yöntemleri ve regresyon modelleri sıklıkla kullanılmaktadır [36]. Ancak bu tür modellerin gerektirdiği varsayımlar pek çok alanda istatistiksel analiz olanaklarını kısıtlamaktadır. İncelenen veri seti üzerinde hiçbir varsayım gerektirmemesi nedeniyle, sınıflama ve regresyon ağaçları (C&RT) bu tür parametrik yöntemlere karşı güçlü bir alternatif olarak ortaya çıkmaktadır [32].

Breiman ve arkadaşları tarafından 1984 yılında geliştirilen çok sayıdaki açıklayıcı (x) değişkeni ile yanıt (y) değişkenine karar vermede kullanılan istatistiksel bir tekniktir. C&RT hem kategorik hem de sürekli değişkenleri kullanarak sınıflama ve regresyon problemlerinin çözümünde karar ağaçlarını kullanan parametrik olmayan istatistiksel bir metottur. Ele alınan bağımlı değişken kategorik ise yöntem sınıflama ağaçları (Classification Tree), sürekli ise regresyon ağaçları (Regression Tree) olarak adlandırılmaktadır [37]. Bu yönüyle C&RT, hem çoklu regresyon analizini hem de bağımlı değişkenin kategorik olduğu durumlarda kullanılan lojistik regresyon analizini kapsamaktadır.

Yapılan çalışmalarda kullanılan C&RT algoritması, her aşamada ilgili kümeyi kendinden daha homojen olan iki alt kümeye ayırarak ikili karar ağaçları oluşturan bir yapıya sahiptir. Diğer bir ifadeyle C&RT, iki yavru düğümü oluşturup bütün bağımsız değişkenleri kullanarak veriyi alt gruplara ayırmak üzerine kurulmuştur. En iyi bağımsız değişken safsızlık (impurity) ve değişim ölçülerindeki (gini, twoing, en küçük kareler sapması) değişkenliği kullanarak seçilir. Burada amaç hedef değişkene ilişkin mümkün olabilen en homojen veri alt gruplarını üretmektir [5].

C&RT, sadece bağımlı değişken ile bağımsız değişken arasındaki ilişkinin yapısını araştırmakla kalmayıp, aynı zamanda bağımsız değişkenlerin birbirleri ile olan etkileşimlerini de ortaya koymaya çalışmaktadır. C&RT algoritmasının, bağımsız değişkenlerin bağımlı değişkenle ilişkisini değerlendirmede ve model içindeki etkileşim yapısını çözümlemede önemli avantajları mevcuttur [37,38].

C&RT'ın sahip olduğu algoritma, benzerlik gösteren değişkenlerin aynı ağaç düğümünde toplanmasına dayalı olup, bütün oluşturduğu alt dalları bağımlı değişken olan kök düğüme bağlamayla son bulmaktadır [38]. C&RT analizi genellikle 3 adımdan oluşmaktadır. Birinci adım veri setini tanımlayan maksimum ağacın oluşturulmasıdır. İkinci adım; oluşturulan ağaçlar içerisinden bağımlı değişkenle önemli ilişkisi olan ağaçları seçmek için yapılan budama işlemi ve son adım ise en uygun ağaç yapısının seçimidir [37].

3.1.1. Maksimum Ağacın Oluşturulması

Maksimum ağaç, ağacın kökünde başlayan bir ikili bölme işlemi kullanan yapıdır. Ağacın kökü, veri seti içerisindeki her nesneyi içermekte ve her bir seviyede kendine özgü iki alt düğüm halinde bölünen bir ana düğüm olarak düşünülmektedir.

Sonraki adımda, her alt grup bir ana grup olmaktadır. Her bölünme bir alt gruptaki tüm nesnelerin benzer bağımlı değişken değerlerine sahip olacak şekilde seçilen bir açıklayıcının değeri ile tanımlanmaktadır [37,38].

Sürekli değişkenlerin bölünmesi x_i'nin seçilmiş bağımsız değişken ve a_j'nin onun bölünme değeri olan "$x_i < a_j$" ile ifade edilmektedir.

Bir bölünme ve onun bölünme değeri için en uygun tanımlayıcıyı seçmek için C&RT, içinde tüm tanımlayıcıların ve tüm bölünme değerlerinin düşünüldüğü bir algoritma kullanmakta ve test koşulunun ne kadar iyi uygulandığını belirlemek için ana düğümün safsızlık derecesini alt düğümlerin safsızlık derecesiyle karşılaştırmaktadır. Ana ve alt düğümlerin safsızlıkları arasındaki fark ne kadar büyükse test koşulu o kadar daha iyi olduğundan, ana düğüm (t_p) ve alt düğümler (t_L ve t_R) arasındaki safsızlık ölçüsünü en iyi azaltan bölünme seçilmektedir. Matematiksel olarak bu durum aşağıdaki gibi ifade edilmektedir [37]:

$$\Delta i(s, t_p) = i_p(t_p) - PLi(t_L) - PRi(t_R) \qquad (3.1)$$

Burada i safsızlığı, s aday bölünme değerini ve PL ile PR sırasıyla sağ ve soldaki alt düğümlerdeki nesnelerin bölünmelerini ifade etmektedir. Bu eşitlikte $\Delta i(s, t)$ değerini maksimize edecek s değerinin seçilmesi amaçlanmakta ve t_p düğümünde bütün kayıtların katılımıyla hesaplanan bu değer, C&RT ağacında gelişme (improvement) kavramı ile ifade edilmektedir. C&RT algoritması ağacı geliştirirken $\Delta i(s, t_p)$'yi maksimize eden bir test koşulu seçtiğinden ve $i_p(t_p)$ bütün test koşulları için aynı olduğundan, $\Delta i(s, t_p)$'yi maksimize etmek alt düğümlerin safsızlık ölçülerinin ağırlıklı ortalamalarını minimize etmekle eşdeğer olmaktadır [37].

Her bir düğümün her aşamada ikiye ayrıldığı C&RT algoritmasında, her bir bölünme noktasının belirlenmesinde Gini, Twoing gibi en iyi bölmeyi seçmek için geliştirilen söz konusu safsızlık ölçütlerinden Gini indeksi kullanılmaktadır. Gini indeksi aşağıdaki gibi hesaplanmaktadır [40].

1) Her nitelik değerleri ikili olacak biçimde gruplanmakta ve bu şekilde elde edilen sol ve sağ bölünmelere karşılık gelen sınıf değerleri gruplandırılmaktadır.
2) Her bir nitelikle ilgili olarak sol ve sağ taraftaki bölünmeler için $Gini_{sol}$ ve $Gini_{sağ}$ değerleri;

k: Sınıfların sayısı,

T: Bir düğümdeki örnekler,

T_{sol}: Sol düğümdeki örneklerin sayısı,

$T_{sağ}$: Sağ düğümdeki örneklerin sayısı,

L_i: Sol düğümde i kategorisindeki örneklerin sayısı.

R_i: *Sağ* düğümde *i* kategorisindeki örneklerin sayısı olmak üzere;

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

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

şeklinde hesaplanmakta ve her j niteliği için, eğitim verisindeki satır sayısı *n* olmak üzere genel *Gini* indeks değeri ise;

$$Gini_j = \frac{1}{n} \left(T_{sol} \times Gini_{sol} + T_{sag} \times Gini_{sag} \right)$$

formülü ile hesaplanmaktadır.

3.2. YAPAY SİNİR AĞI MODELLERİ

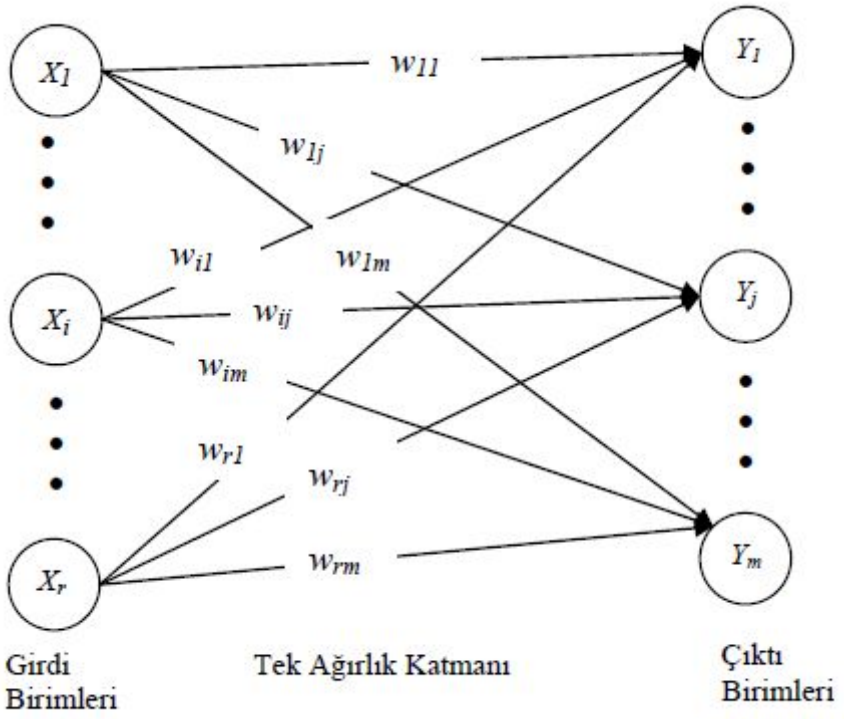
Sinir ağları, tek katmanlı ya da çok katmanlı olarak sınıflandırılırlar. Katman sayısını belirlerken, girdi birimi bir katman olarak sayılmaz; çünkü bunlar üzerinde hiçbir hesaplama işlemi yapılmaz.

3.2.1. Tek Katmanlı Algılayıcılar

Tek katmanlı yapay sinir ağları giriş ve çıkış katmanlarından oluşur. Girdi ve çıktı katmanlarında birden fazla giriş ve çıkış değeri bulunmaktadır. Giriş katmanındaki her giriş değerini çıkış katmanına bağlayan sinaptik bağlantılar mevcuttur. Her bağlantı bir ağırlık değerine sahiptir. Aynı zamanda ağın çıktısının sıfır olmasını engelleyen bias sapma değeri mevcuttur.

Tek katmanlı YSA örnek sınıflandırma, tanıma, örnek ilişkilendirme ve bunun gibi diğer problemlerin çözülmesinde kullanılabilir. Örnek sınıflandırma problemlerinde, her giriş vektörü (örnek, numune) belirli sınıflara ait olabilir ya da olmayabilir. Basit olarak, bir sınıfa üye olma sorusu göz önünde bulundurulur. Çıkış birimi için +1 cevabının alınmasıyla örneğin o sınıfa üye olduğu, -1 cevabı alınırsa, örneğin o sınıfa üye olmadığı belirlenir. Bu tip durumlarda, her bir sınıf için bir çıkış birimi vardır. Örnek tanımlama, örnek tanıma uygulamasımn bir çeşididir. Örneklerin ilişkili hatırlanması ise daha farklıdır [6].

Aşağıda Şekil 3.5’ de tek ağırlık katmanlı YSA için örnek verilmiştir. Bu ağ ileri beslemeli ağlara birer örnektir.



Şekil 3.1. Tek Ağırlık Katmanlı Bir Yapay Sinir Ağı

Tek katmanlı sinir ağlarının eğitilmesinde üç önemli yöntem vardır:

- Hebb Kuralı
- Perseptron Öğrenme Kuralı
- Delta Kuralı

3.2.1.1. Hebb Kuralı

Hebb kuralı, bir yapay sinir ağı için, en eski ve en basit öğrenme kuralı olarak bilinir. Hebb, öğrenmenin, sinaps uzunluklarını (ağırlıkları) değiştirerek meydana geleceğini önermiştir. Hebb'e göre, eğer birbiri ile bağlı iki nöronun her ikisi de aynı zamanda "aktif" ise, bu nöronlara uygun ağırlıkların artırılması gerekmektedir. Benzer olarak, eğer her iki nöron aynı zamanda "pasif" ise, ağırlıkların artırılması gerekir. Bu

durumda, daha güçlü bir öğrenme şekli meydana gelir. Geliştirilmiş Hebb kuralı ile tek katmanlı ileri beslemeli bir sinir ağının eğitilmesi bir Hebb ağını anlatır. Hebb kuralı, diğer özgül ağları eğitmek için de kullanılabilir. Tek katmanlı bir yapay sinir ağında birbiri ile bağlantılı nöronlardan bir tanesi giriş birimi, bir tanesi de çıkış birimi olacaktır (hiçbir giriş birimi birbiri ile bağlanmadığı için, herhangi bir çıkış birimleri de birbiri ile bağlı değildir [48,59].

3.2.1.2. Perseptron

Perseptronlar, YSA'nın öğrenebilir niteliğini taşıyan ilk modelidir. Hebb kuralından daha yetenekli bir öğrenme kuralıdır. Perseptron tekrarlı öğrenme algoritmasıdır ve çözümün varlığı durumunda yakınsama niteliğine sahiptir. Bu, perseptron modelinin en önemli özelliklerinden biridir.

Rosenblatt (1962) ve Minsky-Papert (1969, 1988) tarafından çeşitli perseptron modelleri tanımlanmıştır. Orijinal perseptronlar, duyumsal birimler, birleştirici birimler ve cevap birimleri olmak üzere nöronların üç durumuna sahiptirler. Örneğin, bir basit perseptron duyumsal ve birleştirici birimler için ikili aktivasyon, cevap birimi için ise +1, 0, veya -1 değerlerini üreten aktivasyon uygulayabilir.

Sınıflandırma problemlerinde eşik değerli aktivasyon fonksiyonu kullanılır:

$$f(y\_in) = \begin{cases} -1, & y\_in < -\theta \ ise \\ 0, & -\theta \le y\_in \le \theta \ ise \\ 1, & y\_in > \theta \ \ ise \end{cases} \tag{3.2}$$

Çıktı biriminin aktivasyonu y = f(y_in) şeklinde hesaplamr.

Birleştirici birimden cevap birimine giden bağlantıların ağırlıkları perseptron öğrenme kuralı ile ayarlamr. Her eğitim girişi için, sinir ağı, çıkış biriminin cevabını

hesaplar. Daha sonra sinir ağı, bu örnek için çıkış değeri ile hedeflenen çıkış arasındaki farkı karşılaştırarak bir hata oluşup oluşmadığını tespit eder. Yapay sinir ağı, hesaplanmış çıkış değeri "0" ve hedef değeri "-1" olan örnek için hatayı ayırt edemez, buna karşıt olarak hesaplanmış çıkış değeri "+1" ve hedef değeri "-1" olan örnek için hatayı ayırt edebilir. Bu durumlarda, hedef verinin işareti yönünde ağırlıkların işareti değiştirilmelidir. Bununla birlikte çıkış birimine "0" olmayan sinyaller gönderen bağlantılANN ağırlıkları ayarlanmalıdır. Eğer belirli bir eğitim giriş örneğinde hata oluşuyorsa, ağırlıklar

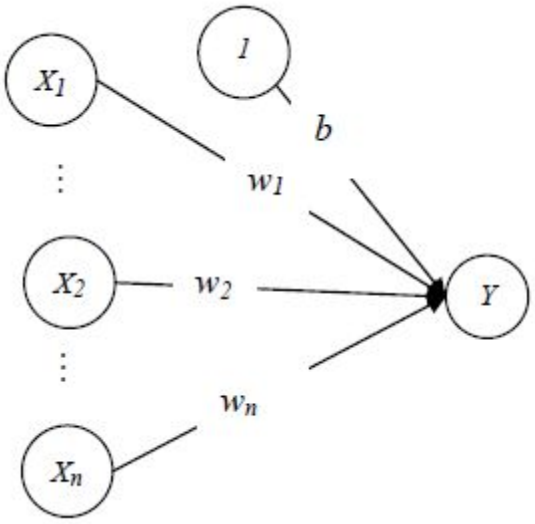
$$w_i(yeni) = w_i(eski) + \alpha t x_i \tag{3.3}$$

formülüne göre değiştirilmelidir.

Burada hedef değeri t ya "+l" ya da "-l"dir ve a öğrenme oranı katsayısıdır. Eğer hata oluşmadıYSA ağırlıklar değiştirilmemelidir. Eğitim işlemi hata oluşmayıncaya kadar devam etmelidir. Perseptron öğrenme kuralı yakınsama teoremine göre, eğer ağda tüm eğitim örnekleri için uygun ağırlıkların varlığına izin verilirse, bu ağırlıklar, eğitim sürecinde bütün eğitim örnekleri için elde edilebilir. Bu kuralın amacı, ağın tarn olarak doğru cevap veremediği eğitim örnekleri için ağırlıkları ayarlamaktır. Ayrıca, eğitim sonunda bu ağ sınırsız sayıdaki eğitim adımları için ağırlıkların değerlerini bulmalıdır [4,7,50].

3.2.1.2.1. Perseptron Algoritması

Şekil 3.6'de perseptronun mimarisi gösterilmiştir. Burada X_1 ,..., X_n girdi birimleri, Y çıktı birimi ve 1 sapma sinyalidir. b sapma ağırlığı, w_i (i = 1,...,n) ağırlıklardır.



Şekil 3.2. Basit Bir Perseptron Mimarisi

Sınıflandırma problemlerinde, sinir ağının görevi tüm giriş örneklerinin belirli bir sınıfa ait olup olmamasını belirlemektir. Sınıfa ait olma çıkışın "+1" değerine, ait olmama ise çıkışın "-1" değerine uygun olmasıyla belirlenir. Sınıflandırma işlemi yapılabilmesi için ağ, tekrarlı bir teknik ile eğitilir. Girdi ve hedefler ikili ve ya iki kutuplu olabilir. θ eşik değeri tüm birimler için değişmezdir. Sapma ve eşik değerinin her ikisinin aynı zamanda kullanılmasına ihtiyaç duyulmaktadır. Bu işlemin algoritması aşağıda verilmiştir. Bu algoritma, ağırlıkların başlangıç değerlerine ve öğrenme oranına tam olarak duyarlı değildir [6].

Adım 0 Ağırlıklar ve sapmalara başlangıç değerleri ata.

(Ağırlıkları ve sapma değeri kolaylık için "0" olarak ahnabilir.)

Öğrenme oranı olan α (0 < α < l)'yı ayarla. (kolaylık için, a, l'e eşitlenebilir.)

Adım 1 Durma koşulu yanlış iken , adım 2-6' yı uygula.

Adım 2 Her bir s:t öğrenme çifti için, 3-5 adımlarını uygula.

Adım 3 Giriş birimlerinin aktivasyonlANNı ayarla. $x_i = s_i$ i = 1, . . . , n

Adım 4 Her çıktı birimi için aktivasyonları hesapla.

$$y\_in_j = b_j + \sum_i x_i.w_{ij} \quad j = 1,....,m:$$

$$f(y\_in) = \begin{cases} -1, & y\_in < -\theta \ ise \\ 0, & -\theta \le y\_in \le \theta \ ise \\ 1, & y\_in > \theta \ ise \end{cases} \tag{3.4}$$

Adım 5 Ağırlıkları ve sapmalan ayarla.

$$eger \ \ t_j \neq y_j \ ise, \tag{3.5}$$

$$b_j(yeni) = b_j(eski) + t_j$$

$$w_{ij}(yeni) = w_{ij}(eski) + t_j x_i.$$

$$eger \ \ t_j \neq y_j \ ise,$$

$$b_j(yeni) = b_j(eski)$$

$$w_{ij}(yeni) = w_{ij}(eski).$$

Adım 6 Durma koşulu:

Eğer adım 2'de hiç bir ağırlık değişmezse dur; aksi duaimda devam et. Algoritmada çıktı birimlerinin sayısı m = 1 olabilir. Örneğin, mantıksal fonksiyonları gözden geçirirken çıktı biriminin sayısının bir olduğu kabul edilir. Eğitimden sonra, ağ her bir eğitim vektörünü doğru şekilde sınıflandırır.

Sınıflandırma ile ilgili perseptron eğitim algoritmasında, ayırma doğrusu yerine, pozitif cevaplar bölgesini sıfır cevaplar bölgesinden ayıran $w_1x_1 + w_2x_2 + b > \theta$ doğrusu ve negatif cevaplar bölgesini sıfır cevaplar bölgesinden ayıran $w_1x_1 + w_2x_2 + b < -\theta$ doğrusu olmak üzere iki ayırma doğrusu vardır [53,55].

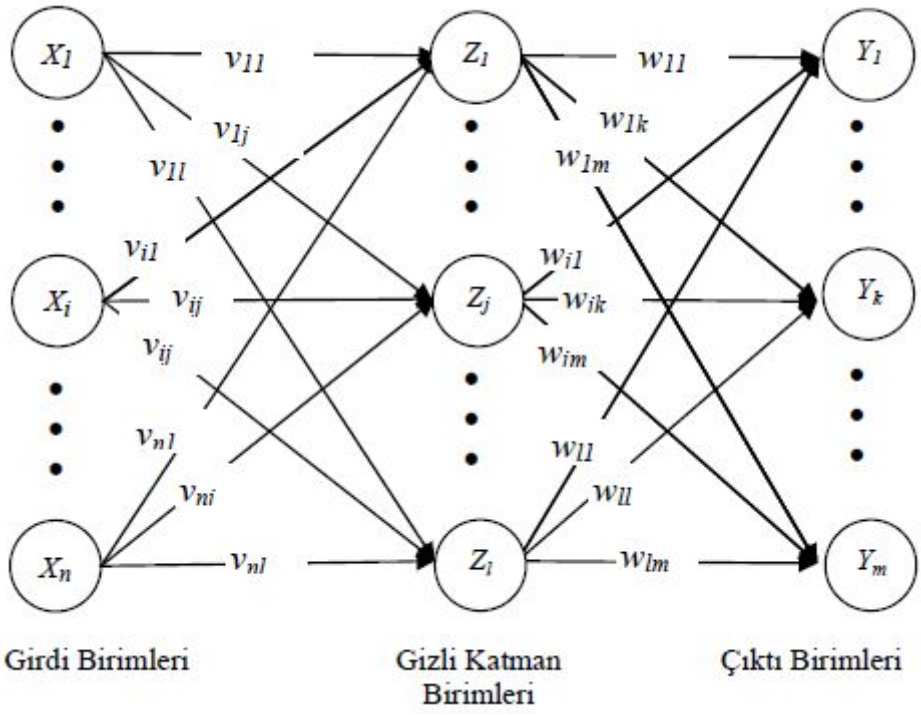
3.2.1.2.2. Delta

Delta kuralı, Widrow ve Hoff (1960) tarafından ADALINE için ortaya atılmış olan iteratif bir öğrenme sürecidir. Delta kuralında, tüm girdi numuneleri için çıktı ve hedef farkları karelerinin toplamının, başka bir ifadeyle, toplam hatanın küçültülmesine hedeflenmiştir. Uygun algoritmalarda her numune için gradient vektörünün ters yönünde ağırlıkların güncellenmesi yapılır. Bu durumda delta kuralı, nöron bağlantılarının ağırlıklarını, ağ girişi (y_in) ve ağın hedef çıkışı (t) arasındaki farkı en aza indirgeyecek şekilde değiştirir. Amaç, tüm eğitim numunelerinin hatalarını en aza indirgemektir. Ağırlık düzeltmeleri, çok sayıdaki eğitim numunesi ile beraber biriktirilebilir ve bu yığın güncelleştirilmesi olarak adlandırılır [6].

3.2.2. Çok Katmanlı Algılayıcılar

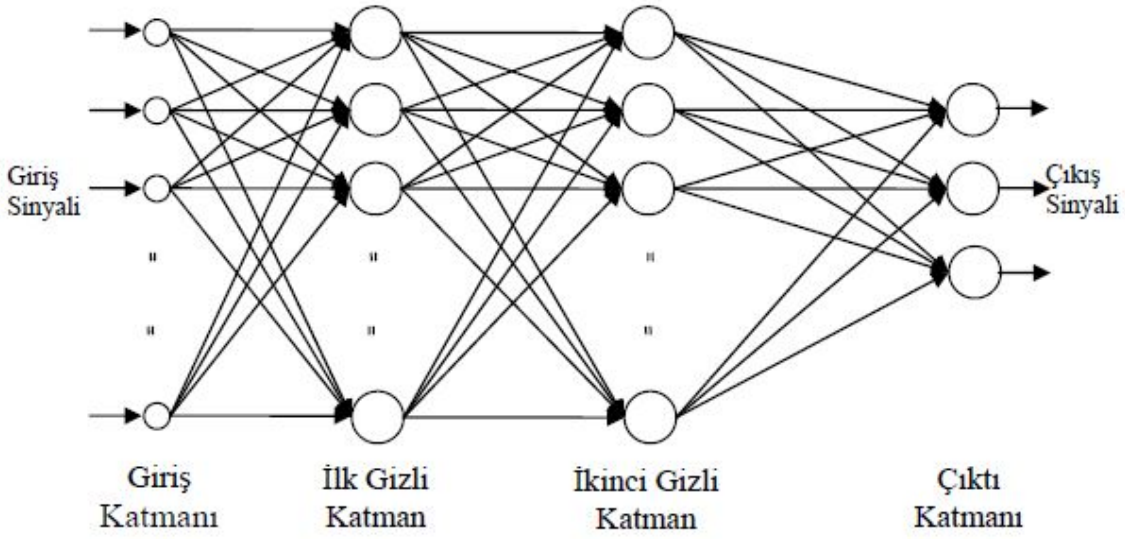
Tek katmanlı ağlar ayrılamayan problemlerin çözümünde başarısız olmaktadırlar. Bunun içinde bilim adamları çok katmanlı YSA modellerini incelemişlerdir. Burada önemli aşamalardan biri bu tip ağlar için akıllı bir eğitim algoritmasını geliştirmekti. 1986 yılında Rumelhart, Hinton ve Williams tarafından bu gerçekleştirildi. Benzer fikirler daha önce Werbos (1974), Parker (1985), Cun (1985) gibi bilim adamlarının yayınlamış olduğu makalelerde gözükmektedir. Standart geriye yayılım (backpropagation) olarak adlandırılan bu eğitim metodu hata kareler toplamının geriye yayılım yöntemiyle küçültülmesi fikrine dayanır ve genelleştirilmiş delta kuralını kullanır. Dolayısıyla bu yöntem her adımda hatanın küçültülmesi için, Widrow-Hoff eğitiminde olduğu gibi, gradient azalış yöntemini kullanır. Bu durumda gizli katmanda doğrusal olmayan aktivasyon fonksiyonlar, örneğin lojistik sigmoid fonksiyonu ve ona

uygun olarak genelleştirilmiş delta kuralı uygulanmaktadır [48,55]. Aşağıda Şekil 3.7' de tek katmanlı YSA için örnek verilmiştir.



Şekil 3.3. Tek Gizli Katmanlı İleri Beslemeli Çok Katmanlı Bir Yapay Sinir Ağı

Çok katmanlı algılayıcılar (ÇKA), bilgilerin girildiği girdi katmanı, bir veya daha fazla sinir hücresinden oluşan gizli katmanları ve bir çıktı katmanını içerir. Girdi sinyalleri ağ boyunca bir katmandan diğer katmana ileri yönde yayılırlar (Şekil 3.8).



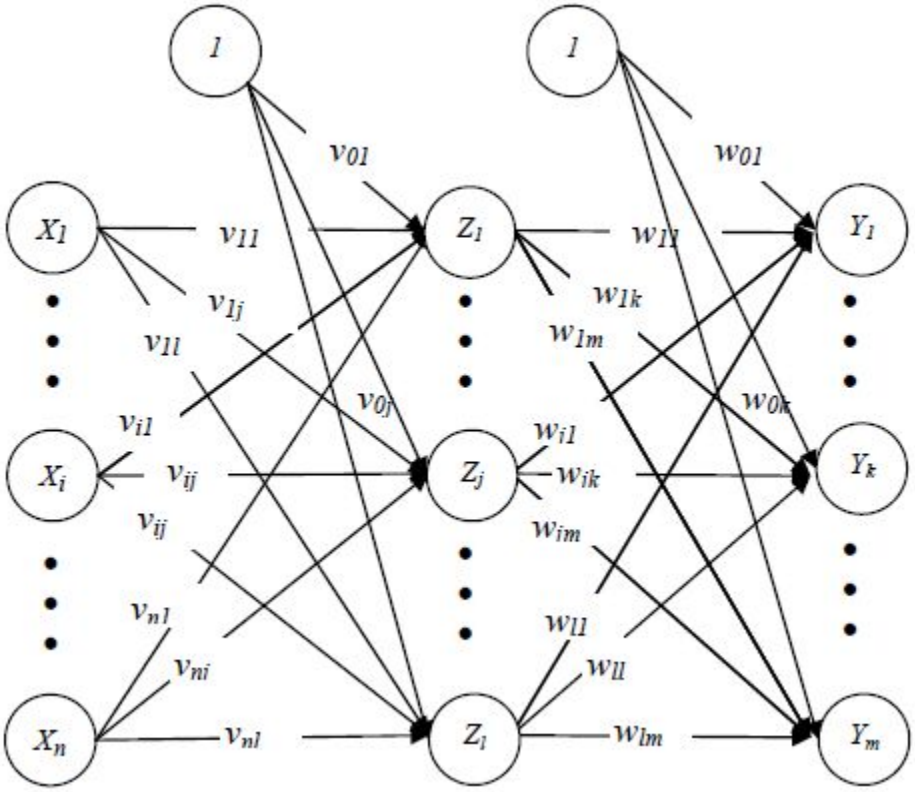
Şekil 3.4. Çok Katmanlı Algılayıcı

Ağın farklı katmanları boyunca ileri ve geri yayılım olarak adlandırılan iki geçiş bulunur. İleri yayılımda, bir girdi vektörü ağın giriş katmanına uygulanır ve bu girdinin etkisi ağda katmandan katmana yayılır. İleri yayılım sırasında ağın sinaptik ağırlıkları belirlidir. Geri yayılımda sinaptik ağırlıkların tümü bir hata-düzeltme kuralı ile uyumlu olarak düzenlenir. Bir hata sinyali üretmek için ağın gerçek çıktısı istenilen bir çıktıdan çıkartılır. Bu hata sinyali sinaptik bağlantıların tersi yönünde ağda geriye doğru yayılır. Sinaptik ağırlıklar ağın gerçek çıktısını istatistiksel anlamda arzu edilen çıktıya yakın hale getirmek için düzenlenir [4,5,7].

3.2.2.1. Geriye Yayılım Algoritması

Şekil 3.9' de bir gizli katmana sahip çok katmanlı ileri beslemeli bir ağ gösterilmiştir. Burada *X* girdi katmanı, *Z* gizli katman, *Y* çıktı katmanına ait birimlerdir. Gizli katmanın *j*. birimine dahil olan sapma ağırlıkları v_{0j} (j =1,...,l) , çıktı katmanının k. birimine dahil olan sapma ağırlıkları w_{0k} (k =1,...,m) ve sapmalara uygun birimlerin

girdi sinyalleri "1" olarak gösterilmiştir. Ağda sinyallerin yayılımı girdi birimlerinden gizli birimlere sonra ise, gizli birimlerden çıktı birimlerine doğru yönelmiştir. Bu nedenle ağ ileri beslemeli çok katmanlı ağ gibi göze alınmaktadır. Gizli birimin sayısı "1" olduğundan şekilde gösterilen ağ iki katmanlı bir yapay sinir ağıdır. Gizli birimlerin sayısının birden fazla olduğu yapay sinir ağ mimarileri de vardır [47,53].



Şekil 3.5. Tek Gizli Katmanlı İleri Beslemeli Çok Katmanlı Bir Yapay Sinir Ağı

3.2.2.2. Standart Geriye Yayılım Eğitim Algoritması

Standart geriye yayılım yöntemi üç aşamadan oluşmaktadır: ileri besleme, hatanın hesaplanması ve geriye yayılması, ağırlıkların güncellenmesi.

3.2.2.2.1. Geriye Yayılım Algoritma Çeşitleri

3.2.2.2.1.1. Momentum

Momentum geriye yayılımda, ağırlık değişiminin yönü o anki eğimle bir önceki eğimin kombinasyonu şeklindedir. Bu eğim azaltma yönteminin değiştirilmiş bir şeklidir ve bazı eğitim verileri, eğitim verilerinin büyük bir çoğunluğundan farklılık gösteriyorsa bu değişim iyi bir avantaj sağlar. Eğer hiç alışılmamış bazı veriler kullanılacaksa bu değişikliği küçük bir öğrenme oranı ile kullanmak iyi olacaktır. Bununla birlikte eğitim verileri benzer olsa da bu değişiklik kullanılarak yaklaşmanın hızı arttırılabilir.

Momentumu kullanmak için bir veya daha önceki eğitim numunelerinin ağırlıkları saklanmalıdır.

Örnek olarak, geri yayılımın momentumlu basit bir biçimi, *t+1* eğitim adımının yeni ağırlıkları *t* ve *t-1* eğitim adımlarındaki ağırlıkları temel alır. Momentumlu geri yayılımın ağırlık güncelleme formülü,

$$w_{jk}(t+1) = w_{jk}(t) + \alpha\delta_k z_j + \mu[w_{jk}(t) - w_{jk}(t-1)] \tag{3.6}$$

$$\Delta w_{jk}(t+1) = \alpha\delta_k z_j + \mu\Delta w_{jk}(t)$$

$$v_{ij}(t+1) = v_{ij}(t) + \alpha\delta_j x_i + \mu[v_{ij}(t) - v_{ij}(t-1)]$$

$$\Delta v_{jk}(t+1) = \alpha\delta_j x_i + \mu\Delta v_{ij}(t)$$

Şeklindedir.Momentum katsayısı değeri olan μ, 0-1 aralığında sınırlandırılmıştır [54,55].

3.2.2.2.1.2. Eşlenik Eğim Algoritması

Birçok eğitim algoritmasında ağırlık güncellenmesinin belirlenmesinde bir öğrenme oranı kullanılmaktadır. Çoğu eşlenik eğim algoritmalarında ağırlık güncellemesi her bir tekrarda yapılmaktadır. Ağ performans fonksiyonunu minimize eden bu işlem için eşlenik eğim yönleri boyunca bir arama yapılmaktadır. Birçok arama fonksiyonu vardır. Bazı arama fonksiyonlan, belirli eğitim fonksiyonları için çok uygundur [6,48].

3.2.2.2.2. Geri Yayım Mantığı

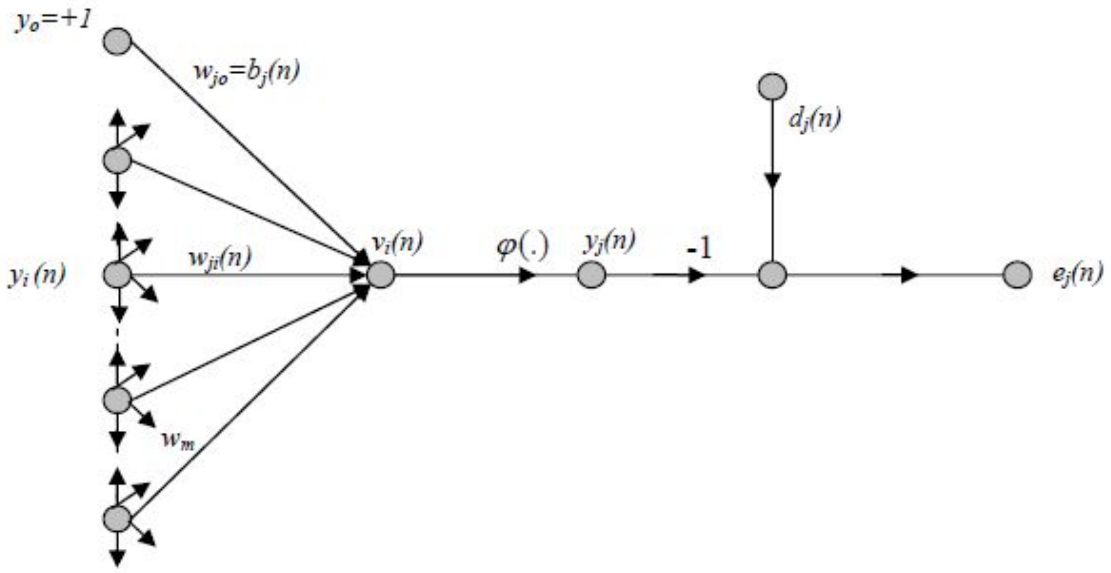
Geri yayılım eğitim yönteminin en genel doğası, geri yayılım tarafından eğitilen çok katmanlı ileri beslemeli bir sinir ağının, çeşitli alanlardaki problemleri çözmek için kullanabileceğini ifade etmektedir. Örneğin, İngilizce harfleri ve bu harflerden oluşan kelimeleri yüksek sesle okuyabilen NETtalk uygulaması, geri yayılımlı ağlarla gerçekleştirilmiştir. Bu tür ağları kullanan uygulamalarla, neredeyse her alanda karşılaşılabilir ve bu uygulamalar, doğadaki çeşitli problemler için sinir ağlarını kullanır. Bu problemler, verilen bir girdi grubu için fonksiyon atamayı içerir. Bu girdiler, hedef çıkışların belirli bir grubuna girer. Amaç, çok farklı YSA yapılarında olduğu gibi, aşağıda değinilen şu iki yetenek arasındaki dengeyi elde etmek için YSA'yı, eğitmektir. Bunlardan ilki, girdi numunelerine doğru bir şekilde cevap verebilme yeteneğidir. Burada söz konusu olan girdi numuneleri, eğitim ya da ezberleme için kullanılırlar. İkincisi ise, YSA'nın, sistem içine verilen girdilere karşı uygun çıktıları verebilme yeteneğidir. Bir YSA'nın geri yayılım yoluyla eğitilmesi üç aşamadan oluşur [6].

1. Girdi eğitim örneğinin ileri beslenmesi.

2. İlişkili hatanın hesaplanması ve geri yayılımı.
3. Ağırlıkların ayarlanması.

Tek katmanlı bir sinir ağı, fonksiyon atamalarda herhangi bir şekilde sınırlandırılmasına rağmen, bir ya da daha çok saklı katmana sahip çok katmanlı bir YSA, sürekli bir fonksiyon atamayı rasgele bir doğrulukla öğrenebilecektir. Tek bir saklı katmandan daha fazla katmana sahip YSA mimarisi, birçok uygulama için daha yararlı olacaktır [60].

Yapay sinir ağlarında eğitim işlemine başlarken ağırlık ve bias değerleri rasgele verilir. Ağa örnekler sunulur ve bilgi ileri doğru yayılır, çıktı katmanındaki nöronlarda hata değeri bulunur [55].



Şekil 3.6. Geri Yayım Mantığı

Çıktı nöronunda n. eğitim basamağında oluşan hata (e), istenilen çıktıdan (d) hesaplanan çıktının (y) çıkarılması ile hesaplanır.

$e_j(n) = d_j(n) - y_i(n)$, j nöronu bir çıktı düğümüdür.

Elde edilen bu hataların anlık hata enerjileri $e_j^2(n)/2$ ifadesi ile hesaplanır. Toplam hata enerjisi bütün nöronlardaki anlık hata enerjilerinin toplanması ile elde edilir. Hata kareler ortalaması, toplam hata enerjilerinin (ε_{ort}) bulunup eğitim küme sayısı N'ye göre normalize edilmesi ile hesaplanır [6].

$$E(n) = \varepsilon_{ort} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\sum_{j\in C} e^2 j(n) \tag{3.7}$$

Öğrenme sürecinin amacı hatayı minimize edecek ağın serbest parametrelerini yani ağırhk ve bias değerlerini ayarlamaktır. Bu işlem yapılırken eğitim setlerinin hepsi ağa bir kez sunulur, serbest parametreler ondan sonra ayarlanır ya da sunulan her örnekten sonra ağırlık ve bias değerleri güncellenir. Ağırlıklara yapılan düzenlemeler ağa sunulan her örnek için hesaplanan hatayla uyumlu olarak yapılır.

Geri besleme algoritmasında ağırlık ve bias değerlerini güncellemek için düzeltme değeri $(\Delta w_{ji}(n))$, hatanın ağırlıklara göre kısmi türevinin alınması $(\partial\varepsilon(n)/\partial w_{ji}(n))$ ile hesaplanır.

$$\frac{\partial\varepsilon(n)}{\partial w_{ji}(n)} = -e_j(n)\varphi'(v_j(n))y_i(n) \tag{3.8}$$

$$\Delta w_{ji}(n) = -\eta\frac{\partial\varepsilon(n)}{\partial w_{ji}(n)}$$

$$\delta_j(n) = -e_j(n)\varphi'(v_j(n))$$

Yukardaki eşitlikte kullanılan η, geri besleme algoritmasının öğrenme oranı parametresidir, eksi işareti ağırlık uzayındaki hatayı düşürecek ağırlık değişimi için yön arayan gradyan azalanı ifade eder. Lokal gradyan $(\delta_j(n))$, ilgili nöron için karşılık gelen hata sinyali $e_j(n)$ ve ilgili aktivasyon fonksiyonunun türevinin $\varphi'(v_j(n))$ çarpımına eşittir [59,48,53].

Düzeltme değeri, lokal gradyanların kullanımı ile aşağıdaki eşitlik ile ifade edilir.

$$\Delta w_{ji}(n) = -\eta \delta_j(n) y_i(n) \tag{3.9}$$

Ağ eğitim esnasında hata yüzeyi sabit bir noktada takılı kalabilir, minimum hataya yakınsayamayabilir. Bunu engellemek için düzeltme değerine, momentum katsayısı olarak ifade edilen bir değer eklenir.

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) - \eta \delta_j(n) y_i(n) \qquad \alpha \text{ momentum katsayısı} \tag{3.10}$$

Elde edilen bu düzeltme değeri eski ağırlıklara eklenerek yeni ağırlık değerleri elde edilir. Bu işlem doğru ağırlık değerleri elde edilene kadar devam eder [6,55,60, 61].

3.3. LOJİSTİK REGRESYON MODELİ

3.3.1. Lojistik Regresyon Modelinin Oluşturulması

Lojistik regresyon modeli oluşturulurken, doğrusal regresyonda olduğu gibi maksimum olabilirlik tahmini yöntemi kullanılır. (x_i, y_i) gibi *n* tane bağımsız gözlem eşinin olduğu varsayıldığında y_i iki düzeyli bağımlı değişkeni ve x_i 'de *i*' inci denek için bağımsız değişkenin değerini temsil etsin. Sonuç değişkeni için 0 ve 1 kodlarının sırasıyla belirli bir olayın varlığını ya da yokluğunu temsil ettiği varsayılsın. Lojistik regresyon modelini kestirebilmek için bilinmeyen β_0 ve β_1 parametrelerini tahmin etmek gerekmektedir [44].

$$(E(Y \setminus x) = \beta_0 + \beta_1 x) \tag{3.11}$$

Eğer Y, 0 ve 1 olarak kodlandıYSA, π(x) ifadesi x verildiğinde Y' nin l'e eşit olma koşullu olasılığını vermektedir. [π(x) = P(Y = l\ x)] $_x$ [1- π(x)] değeri verilen herhangi bir x için Y'nin 0'a eşit olma koşullu olasılığını göstermektedir.

[l - π{x) = P(Y = 0 / x] $_x$ (x_i, y_i) Çiftinin y_i = 1 olduğu zaman olabilirlik fonksiyonuna katkısı π(x_i) iken, y_i = 0 olduğu zaman olabilirlik fonksiyonuna katkısı 1-π(x_i) kadardır. (x_i, y_i) Çiftinin olabilirlik fonksiyonuna katkısını ise aşağıdaki ifade yardımıyla tanımlanmıştır [41,44].

$$\xi(x_i) = \pi(x_i)^{yi}\left[1-\pi(x_i)\right]^{1-yi} \tag{3.12}$$

Gözlemlerin birbirlerinden bağımsız oldukları varsayıldığı için, olabilirlik fonksiyonunu $E(Y \setminus x) = \beta_0 + \beta_1 x$ eşitliğindeki terimlerin çarpılmasıyla elde edilir.

$$\int(\beta) \quad \prod_{i=1}^{n} \varsigma(x_i) \tag{3.13}$$

En çok olabilirliğin temel ilkesinde β tahmininin $g(x) = \ln\left[\dfrac{\pi(x)}{1-\pi(x)}\right]$ eşitliğini maksimum yaptığı vurgulanır, öte yandan log olabilirlik fonksiyonu ise:

$$L(\beta) = \ln\left[1(\beta)\right] \quad \sum_{i=1}^{n}\left\{y_i \ln\left[\pi(x_i)\right] + (1-y_i)\ln\left[1-\pi(x_i)\right]\right\} \tag{3.14}$$

L(β) ' yı maksimum yapan β değerini bulabilmek için, L(β) ' nın β_0 ve β_1 'e göre türevini aldıktan sonra, elde edilen ifadeler 0'a eşitlenmelidir. Sonuçta elde edilen en çok olabilirlik eşitlikleri aşağıdaki gibidir.

$$\prod_{i=1}^{n}\left[y_i - \pi(x_i)\right] = 0$$
$$\prod_{i=1}^{n} x_i\left[y_i - \pi(x_i)\right] = 0 \tag{3.15}$$

Yukarıdaki eşitlikten elde edilen β' nın değeri, en çok olabilirlik tahmincisi olarak adlandırılır ve β olarak gösterilir. Örnek olarak, π(x_i)' nin en çok olabilirlik

tahmini $\pi\{x_i)$ile gösterilir. Bu değer, x' in x_i gibi bir değere eşit olduğu verildiği zaman, Y nin l'e eşit olma koşullu olasılığının tahminini verir [41,44].

$$\prod_{i=1}^{n} \not\equiv_i \quad \prod_{i=1}^{n} \hat{\pi}(x_i) \tag{3.16}$$

3.3.2. Çoklu Lojistik Regresyon Modeli

Birden çok bağımsız değişkenin yer aldığı Lojistik modellere çok değişkenli lojistik regresyon adı verilir. Yapısal olarak bu modelin diğer çok değişkenli regresyon modellerinden farkı olmayıp regresyon katsayılarının yorumlanması farklıdır. Çoklu lojistik regresyon modelinde genel eğilim modeldeki katsayıların tahmini ve onların önem testi şeklinde olmaktadır.

x' = (x1, x2, ….xp) vektörü ile gösterilen, *p* tane bağımsız değişken toplandığı varsayılsın. Şu an için bu değişkenlerin her birinin en azından aralıklı ölçekle (sürekli) ölçüldüğü varsayılsın. Bağımlı değişkeninin mevcut olduğu (Y=1) zaman ki koşullu olasılık P(Y = l\x) = πfx/e eşittir. Çoklu lojistik regresyon modelinin lojiti aşağıdaki eşitlik ile verilmiştir.

$$g(x) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p + ... \tag{3.17}$$

bu durumda

$$\pi(x) = \frac{e^{g(x)}}{1 + e^{g(x)}} \tag{3.18}$$

Eğer bazı bağımsız değişkenler kesikli, nominal ölçekli ise, o zaman bu değişkenleri sürekli değişkenlermiş gibi modele dahil etmek yanlış olacaktır. Çeşitli düzeyleri göstermek için kullanılan sayılar sadece tanımlayıcıdırlar ve bunların herhangi bir sayısal değeri yoktur. Bağımsız değişkenler sayısal olarak sınıflandırıldığı zaman

çeşitli dizayn değişkenlerinin (kukla değişken) kategorik olan bu değişkenleri temsil etmesi için kullanılması gerekmektedir [4,6,8].

Genel olarak, eğer nominal bir değişken k kategoriye sahipse, o zaman *k*-1 dizayn değişkenine ihtiyaç vardır. *j*'ninci bağımsız değişken (xj) *kj* kategoriye sahip olsun. *kj* – 1 dizayn değişkeni D_{ju} olarak katsayıları da β_{ju} , u =1,2,…,*kj* -1 olarak belirtilmiş olsun. Sonuç olarak *J*'ninci değişkeni kesikli olan *p* değişkenli model için lojit aşağıdaki gibidir [6,8].

$$g(x) = \beta_0 + \beta_1 x_1 + ... + \sum_{u=1}^{k_j - 1} \beta_{jk} D_{ju} + \beta_p x_p ... \tag{3.19}$$

3.3.3. Çoklu Lojistik Regresyon Modelinin Kurulması

Birbirinden bağımsız *n* tane (x_i , y_i), *i*=1,2,…,*n* gözlem eşinin olduğu varsayılsın. Tek değişkenli modelde olduğu gibi modelin kurulması için tahmin vektörünün $\beta' = (\beta_0 , \beta_1, \ldots , \beta_p)$ elde edilmesi gerekmektedir. Çok değişkenli durumda, tek değişkenli durumda olduğu gibi tahmin metodu en çok olabilirlik olacaktır. Log olabilirlik fonksiyonunun *p*-1 katsayısına göre türevi alınarak, *p*+1 tane olabilirlik denklemi elde edilebilmektedir [8].

$$\sum_{i=1} \left[y_i - \pi(x_i) \right] = 0 \tag{3.20}$$

$$\sum_{i=1} x_{ij} \left[y_i - \pi(x_i) \right] \quad 0, j \quad 1, 2, ..., p, ...$$

3.3.4. Lojistik Regresyon Modelinde Katsayıların Yorumlanması

Karar vermemiz gereken ilk adım, "bağımlı değişkenin hangi fonksiyonu bağımsız değişkenler ile doğrusal bir fonksiyon oluşturmaktadır?" sorusudur. Bu fonksiyona link fonksiyonu adı verilir. Doğrusal regresyon modelinde link fonksiyonu I

(identity-Birim) matristir. Çünkü bağımlı değişken parametreleri ile doğrusaldır. Lojistik regresyon modelinde ise link fonksiyonu lojit transformasyondur [41].

$$g(x) = \ln\left\{\frac{\pi(x)}{[1-\pi(x)]}\right\} \quad \beta_0 + \beta_1 x \tag{3.21}$$

3.3.5. Modelde İkiden Fazla Bağımsız Değişkenin Olduğu Durum

Sürekli bir değişken için katsayının yorumlanması amacıyla geliştirilecek metot için lojitin değişkenle doğnısal olduğu varsayılacaktır. Lojitin sürekli değişkenle (x) doğrusal olması varsayımı altında lojit için eşitlik g(x) = $\beta_0 + \beta_1$ 'dir. Eğim katsayısı (β_1) x'deki "1" birimlik artışın log odds değerinde meydana getireceği değişimi verir. x'deki "c" birimlik bir değişim için log odds değeri lojit farktan elde edilmektedir.

g(x + c) -g(x) = c β_1.Karşılık gelen odds oranı lojit farkın üssü alınarak elde edilir.

ψ (c) = ψ (x + c, x) = exp(c β_1) ψ (c) 'nin tahmini için %100(1-α) güven aralığının uç noktaları $\exp\left[c\hat{\beta}_1 \pm z_{1-\alpha} c\hat{S}E(\hat{\beta}_1)\right]$ olarak verilmiştir [6].

3.4. YÖNTEMLERİN KARŞILAŞTIRILMASI

Yöntemlerin birbirlerine karşı olan sınıflama başarımlarının karşılaştırılması için aşağıdaki ölçütler kullanılmıştır.

Düzensizlik matrisi: Yakınsaklık matrisi olarak da adlandırılır. Doğru olarak sınıflandırılan örneklerin sayısı bu matrisin köşegeni üzerindeki elemanlarının toplamına eşittir. Doğru olarak sınıflandırılan kayıt yüzdesi bize madencilik algoritmalarını karşılaştırma olanağı sunmaktadır.

Kesinlik: Gerçekte herhangi bir sınıfa ait olan kayıtların hangi oranda sınıflandırma algoritması tarafından o sınıfa atandığı gösterir.

F-Ölçütü: Kesinlik ve duyarlılık ölçütlerini beraber değerlendirmek için, her iki değerin harmonik ortalaması.

Kappa istatistiği: Tahmin doğruluğunun ölçüsüdür. Kappa katsayısı, aynı nesneyi derecelendiren iki gözlemci arasındaki uyumu test etmek amacıyla kullanılır. Örneğin, aynı bireylerin röntgenini değerlendiren iki klinisyenin, lezyon bulgularının karşılaştırılmasında birbirleriyle ne düzeyde uyum içinde olduğu kappa değeri ile gösterilebilmektedir. Kappa katsayısı 0-1 aralığında değer alır ve buna göre,

0,93-1: mükemmel,

0,81-0,92: çok iyi,

0,61-0.80: iyi,

0,41-0,60: orta düzeyde

0,21-0,40: ortanın altında ve

0,01-0,20: zayıf uyumu tanımlamaktadır [67].

ROC Eğrisi: ROC eğrisi, tanı testlerinin performanslarının değerlendirilmesi ve kıyaslanması için yaygın olarak kullanılan bir yöntemdir. Bu yöntem, bir tıbbi testin en uygun duyarlılığını ve en uygun özgüllüğünü belirlemek için optimum kesim noktalarının belirlenmesini sağlar. ROC eğrisi yöntemi, tanı testinin sınıflandırma performansının iki boyutlu grafiksel gösterimidir. Sınıflandırmanın doğruluğu, ROC eğrisi altında kalan alanın (AUC) büyüklüğüne bağlıdır [68].

ROC eğrisi altında kalan alan, $A = f(b/\sqrt{1+a^2})$ eşitliğiyle hesaplanır. Burada F kümülatif standart normal dağılım fonksiyonu, a eğim ve b sabit katsayılardır. ROC eğrisi altındaki alan, 0 ile 1 arasında değer almaktadır [69].

3.5. WEKA Programı

Weka, Yeni Zelanda'daki Waikato Üniversitesi tarafından geliştirilmiş olup "Waikato Environment for Knowledge Analysis" kelimelerinin baş harflerinin kısaltmasıdır. Weka başta Yeni Zelanda"da tarımsal verinin işlenmesi amacıyla geliştirilmiştir. Bununla birlikte sahip olduğu makine öğrenme yöntemleri ve veri mühendisliği kabiliyeti öyle hızlı ve köklü bir şekilde gelişmiştir ki, şimdi veri madenciliği uygulamalarının tüm formlarında yaygın olarak kullanılmaktadır [65].

Weka, bir öğrenen makineler algoritmaları koleksiyonu olduğu gibi yeni algoritmaların geliştirilmesi için de çok uygundur. GNU (General Public License) altında yayınlanmış, Java dilinde kodlanmış, açık kaynaklı bir yazılımdır. Ayrıca WEKA, Windows, Linux ve Macintosh gibi farklı işletim sistemleri üzerinde çalışabilen bir programdır [66]. Weka Grafiksel Kullanıcı Ara yüzü, WEKA'nın grafiksel çevresine erişim için kullanılmaktadır.

Weka pencerenin alt kısmında ise dört adet seçenek bulunmaktadır:

- ***Simple CLI:*** WEKA komutlarının direkt olarak işlenmesine olanak sağlayan basit bir komut satırı ara yüzü sağlar.
- ***Explorer:*** Verinin WEKA ile keşfi için bir ara yüzdür. Bu ara yüzde VM ile sınıflandırma, kümeleme ve birliktelik kuralı uygulamaları kolaylıkla gerçekleştirilmektedir. Weka Explorer ile Bayes sınıflayıcısı, karar ağaçları, karar kuralları, regresyon, yapay sinir ağları gibi sınıflandırma algoritmaları; K-

ortalama, Cobweb gibi kümeleme algoritmaları; Apriori gibi birliktelik kuralları kolaylıkla uygulanabilmektedir. Weka Explorer‟ da önişleme, sınıflama, kümeleme, birliktelik kuralları, özellik seçme ve görselleştirme panelleri bulunmaktadır.

Önişleme*: Veri* dosyalarının yüklendiği, veri tabanının seçildiği ve verinin çeşitli yollarla değiştirildiği keşif sürecinin ilk adımıdır.

Sınıflama: Sınıflandırma ve regresyon algoritmalarının uygulanıp değerlendirildiği paneldir. Sınıflandırma fonksiyonları, kuralları, karar ağaçları, Bayes ağları, sinir ağları gibi sınıflandırma algoritmaları bu panelde yer almaktadır.

Kümeleme: K-ortalama, cobweb gibi kümeleme algoritmalarının yer aldığı paneldir.

Birliktelik kuralları: Verilerden birliktelik kurallarının çıkarıldığı paneldir.
Özellik seçme: Veri kümesindeki ilişkili verilerin seçildiği paneldir.
Görselleştirme: Özellikler arasındaki ilişkiler iki boyutlu grafiklerle izlenebildiği paneldir.

- ***Experimenter:*** Deneylerin gerçekleştirilmesi ve öğrenme planları arasındaki istatistiksel testleri yürüten bir ara yüzdür. Bir veri setine farklı yöntemleri uygulayarak ya da aynı tekniği farklı parametrelerle tekrarlayarak, tek seferde birden fazla deneyin gerçekleştirilmesine izin veren bir araçtır.
- ***Knowledge Flow:*** Weka veri madenciliği paketi ile sağlanan fonksiyonelliğin alternatif bir ara yüzüdür. Bu ara yüz temel olarak Explorer ile aynı işlevleri sürükle-bırak ara yüzü ile yerine getirmektedir. Experimenter tarafından

desteklenmeyen ek özellikleri ve experimenter de bulunan bazı eksik özellikleri ile gelişmekte olan bir bölümdür.

4. BULGULAR

4.1. TANIMLAYICI İSTATİSTİKLER

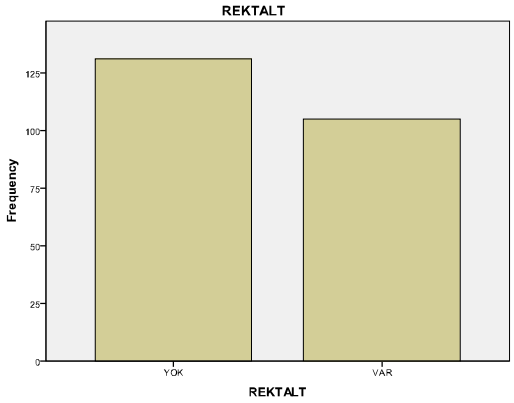
Öncelikle tüm veri seti için daha sonra ise prostat kanseri tanısı konmuş ve prostat kanseri tanısı konmamış hastalar için tanımlayıcı istatistiklere bakalım.

Tablo 4.1. Sürekli değişkenler (Yaş ve PSA) İçin Tanımlayıcı İstatistikler (n=236)

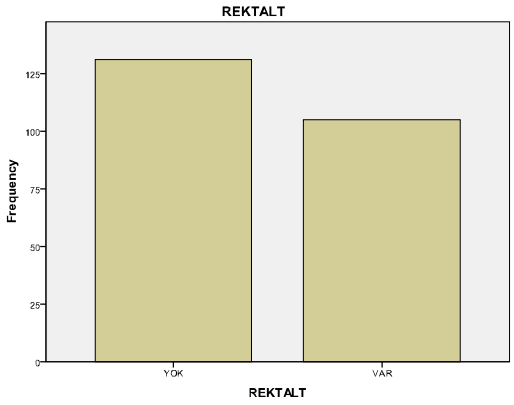
	Min.	**Maks.**		**SS**
YAŞ	40	85	65.22	8.09
PSA	0.037	151.0	15.70	28.59

Tablo 4.2. Kategorik Değişkenler (Rektal Tuşe ve Genetik Yatkınlık) İçin Tanımlayıcı İstatistikler (n=236)

		n	**%**
Rektal Tuşe	**Pozitif**	105	44.5
	Negatif	131	55.5
Genetik Yatkınlık	**Var**	18	7.6
	Yok	218	9.4



Şekil 4.1. Rektal Tuşe Sonuçlarının Dağılımı



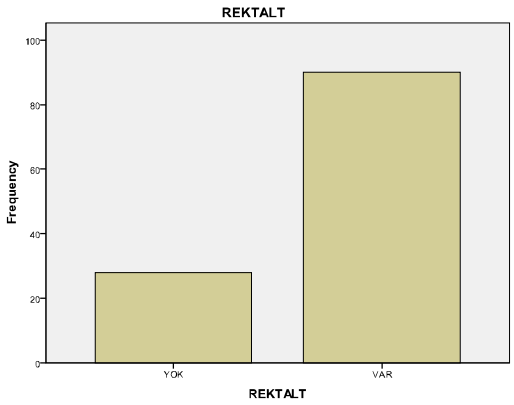
Şekil 4.2. Genetik Yatkınlık Sonuçlarının Dağılımı

Tablo 4.3. Prostat Kanseri Tanısı Durumuna Göre Sürekli Değişkenlerin (Yaş ve PSA) Dağılımı

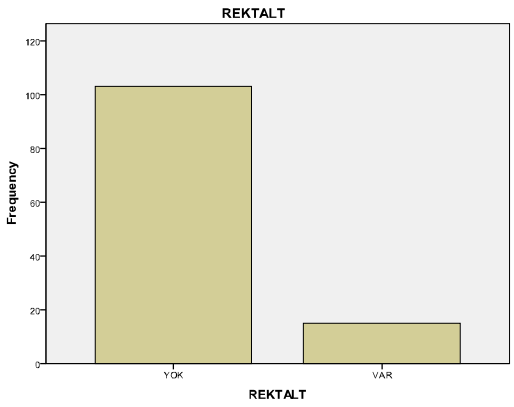
	Prostat Kanseri		**Normal**		**t**	**p**
	Ort	**SS**	**Ort**	**SS**		
YAŞ	69.29	7.23	61.15	6.78	8.911	<0.001
PSA	27.78	36.59	3.62	2.98	8.911	<0.001

Tablo 4.4. Prostat Kanseri Tanısı Durumuna Göre Kategorik Değişkenlerin (Rektal Tuşe ve Genetik Yatkınlık) Dağılımı

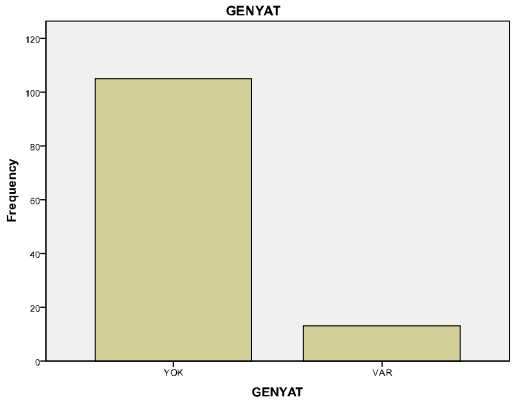
		Prostat Kanseri		Normal		χ^2	p
		n	%	n	%		
Rektal Tuşe	**Pozitif**	90	76.3	15	12.7	96.510	<0.001
	Negatif	28	23.7	103	87.3		
Genetik Yatkınlık	**Var**	13	11.0	5	4.2	3.849	0.084
	Yok	105	89.0	113	95.8		



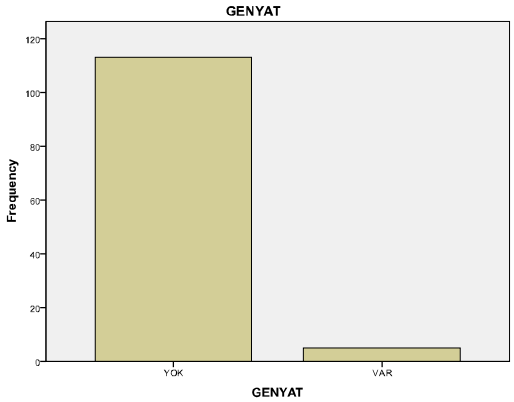
Şekil 4.3. Prostat Kanseri Tanısı Koyulan Hastaların Rektal Tuşe Sonuçları



Şekil 4.4. Prostat Kanseri Tanısı Konulmayan (Normal) Hastaların Rektal Tuşe Sonuçları



Şekil 4.5 Prostat Kanseri Tanısı Koyulan Hastaların Genetik Yatkınlık Durumları Dağılımı



Şekil 4.6. Prostat Kanseri Tanısı Konulmayan (Normal) Hastaların Genetik Yatkınlık Durumları Dağılımı

4.2. LOJİSTİK REGRESYON ANALİZİ

Aşağıda WEKA programına ait Lojistik Regresyon Analizi yöntemi sonuçları verilmiştir.

Tablo 4.5. Odds Oranları (LRA)

LRA	
Değişken	**Odds Oranı**
YAŞ	1.125
PSA	1.220
RT	14.679
GP	0.312

Tablo 4.6. Sınıflandırma Sonuçları-I (LRA)

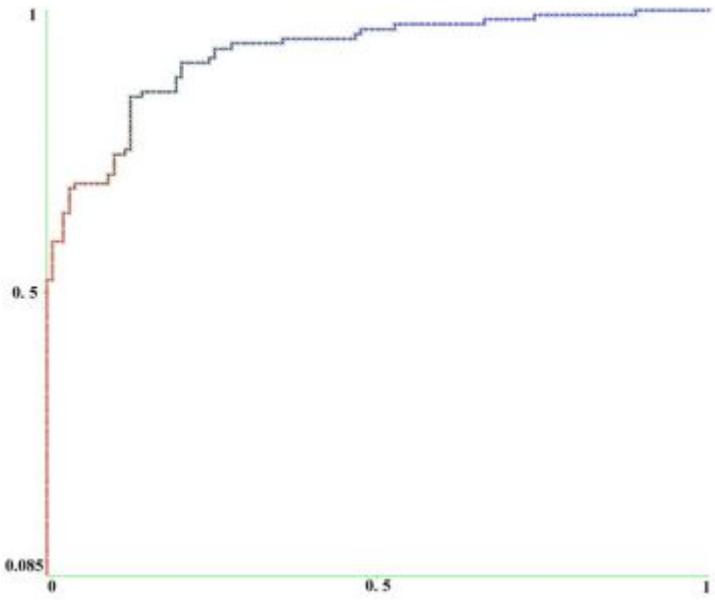
LRA	
N	236
Doğru Sınıflandırma	198 (%83.89)
Yanlış Sınıflandırma	38 (%16.11)
Kappa	0.678
Ortalama Mutlak Hata (MAE)	0.211
Ortalama Karesel Hata (RMSE)	0.334
Göreceli Mutlak Hata (REA)	%42.22
Göreceli Karesel Hata (RRSE)	%66.81

Tablo 4.7. Sınıflandırma Sonuçları-II (LRA)

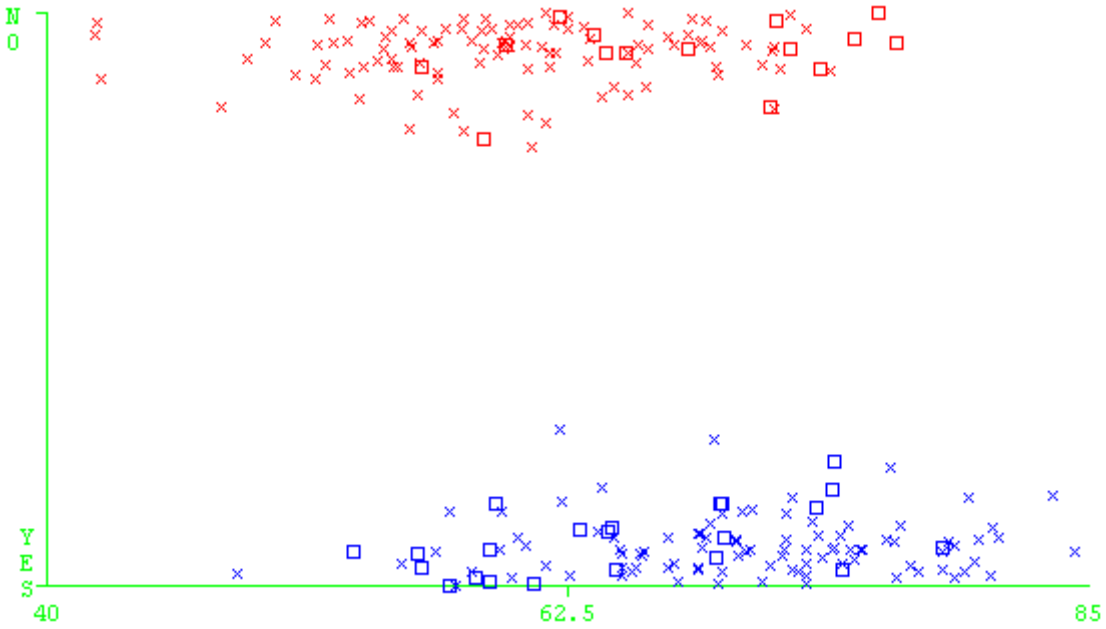
LRA			
Kesinlik	**Duyarlılık**	**F-Ölçütü**	**AUC**
%86.40	%80.50	%83.30	0.924

Tablo 4.8. Düzensizlik Matrisi Sonuçları (LRA)

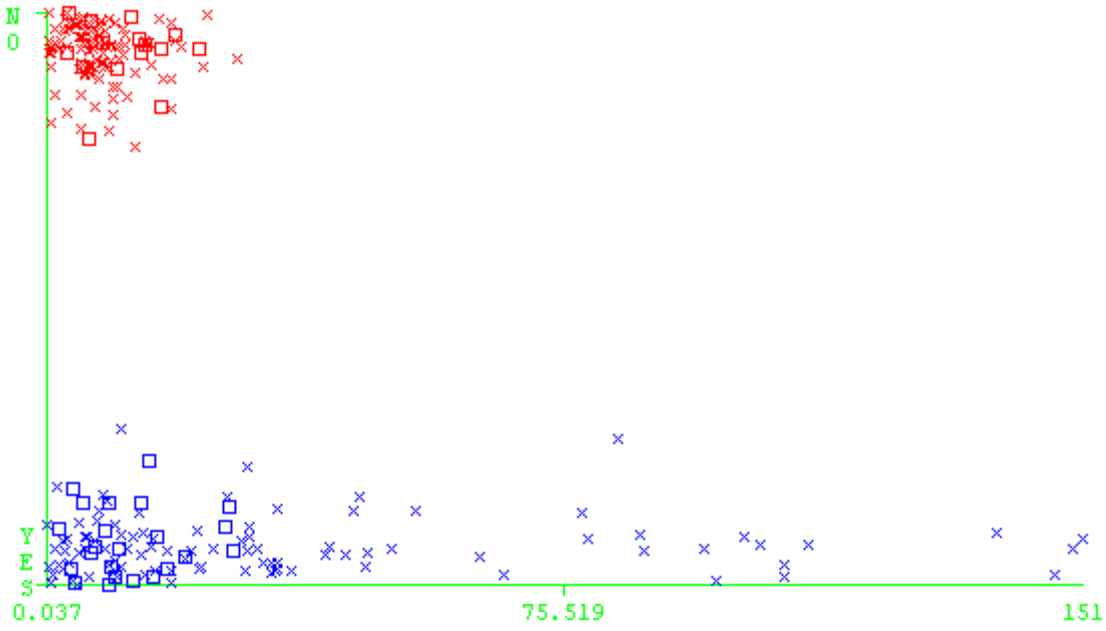
LRA		
a	**b**	**Sınıflama**
95	23	A=YES
15	103	B=NO



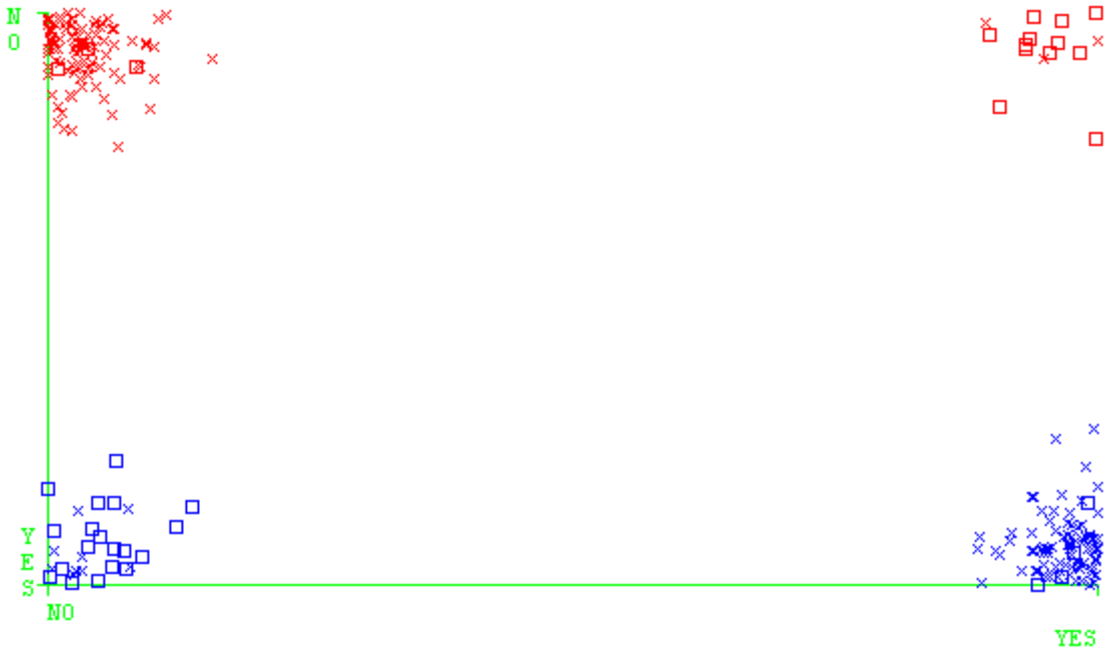
Şekil 4.7. LRA İçin ROC Eğrisi



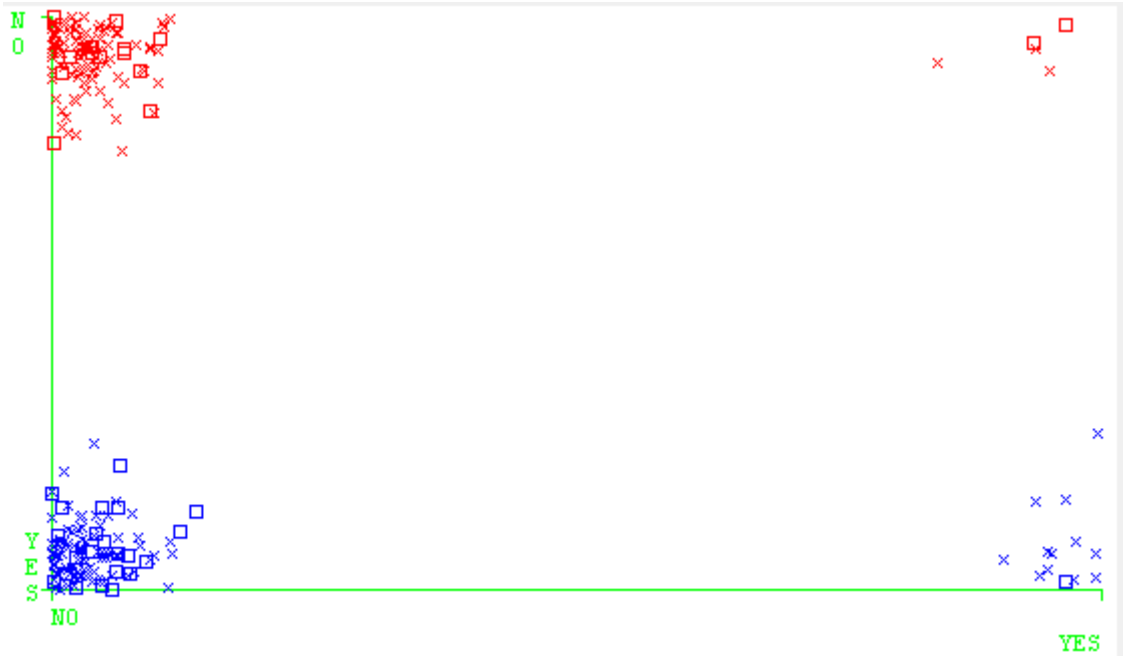
Şekil 4.8. Yaş İçin Yanlış Sınıflandırmaların Gösterimi (LRA)



Şekil 4.9. PSA İçin Sınıflandırma Hatalarının Gösterimi (LRA)



Şekil 4.10. Rektal Tuşe İçin Sınıflandırma Hatalarının Gösterimi (LRA)



Şekil 4.11. Genetik Yatkınlık İçin Sınıflandırma Hatalarının Gösterimi (LRA)

4.3. SINIFLANDIRMA VE REGRESYON AĞAÇLARI

Weka programında model oluştururken kullanılabilecek pek çok karar ağaçları algoritmaları mevcuttur. ADTree, BFTree, Decision Stump, FT, C&RT, C&RTgraft, LADTree, LMT, RBTree, RandomForest, RandomTree, RepTree algoritmaları Weka programı içerisinde mevcuttur. Çalışmada C&RT algoritması kullanılmıştır.

Tablo 4.9. Sınıflandırma Sonuçları-I (C&RT)

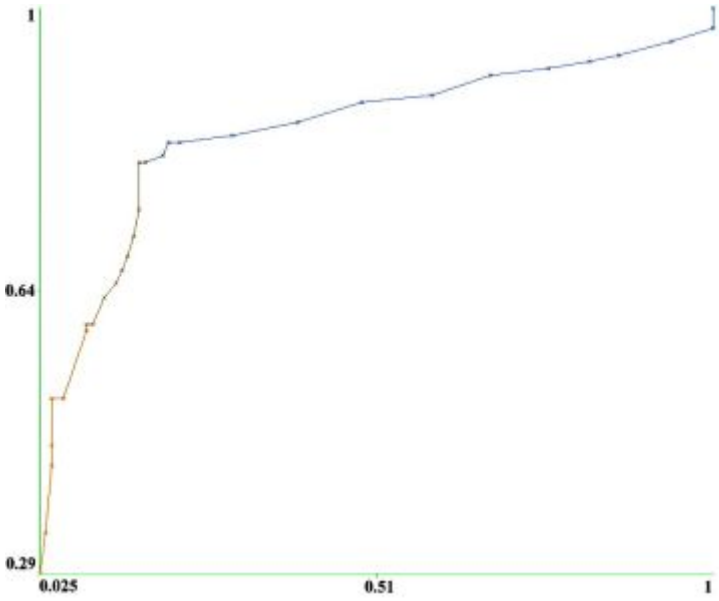
C&RT	
Doğru Sınıflandırma	193 (%81.78)
Yanlış Sınıflandırma	43 (%18.22)
Kappa	0.636
Ortalama Mutlak Hata (MAE)	0.246
Ortalama Karesel Hata (RMSE)	0.395
Göreceli Mutlak Hata (REA)	%49.19
Göreceli Karesel Hata (RRSE)	%78.95
Toplam Örnek Sayısı	236

Tablo 4.10. C&RT Sınıflandırma Sonuçları-II (C&RT)

C&RT			
Kesinlik	**Duyarlılık**	**F-Ölçütü**	**AUC**
%82.60	%80.50	%81.50	0.828

Tablo 4.11. Düzensizlik Matrisi Sonuçları (C&RT)

C&RT		
a	b	Sınıflama
95	23	A=YES
20	98	B=NO



Şekil 4.12 C&RT İçin ROC Eğrisi

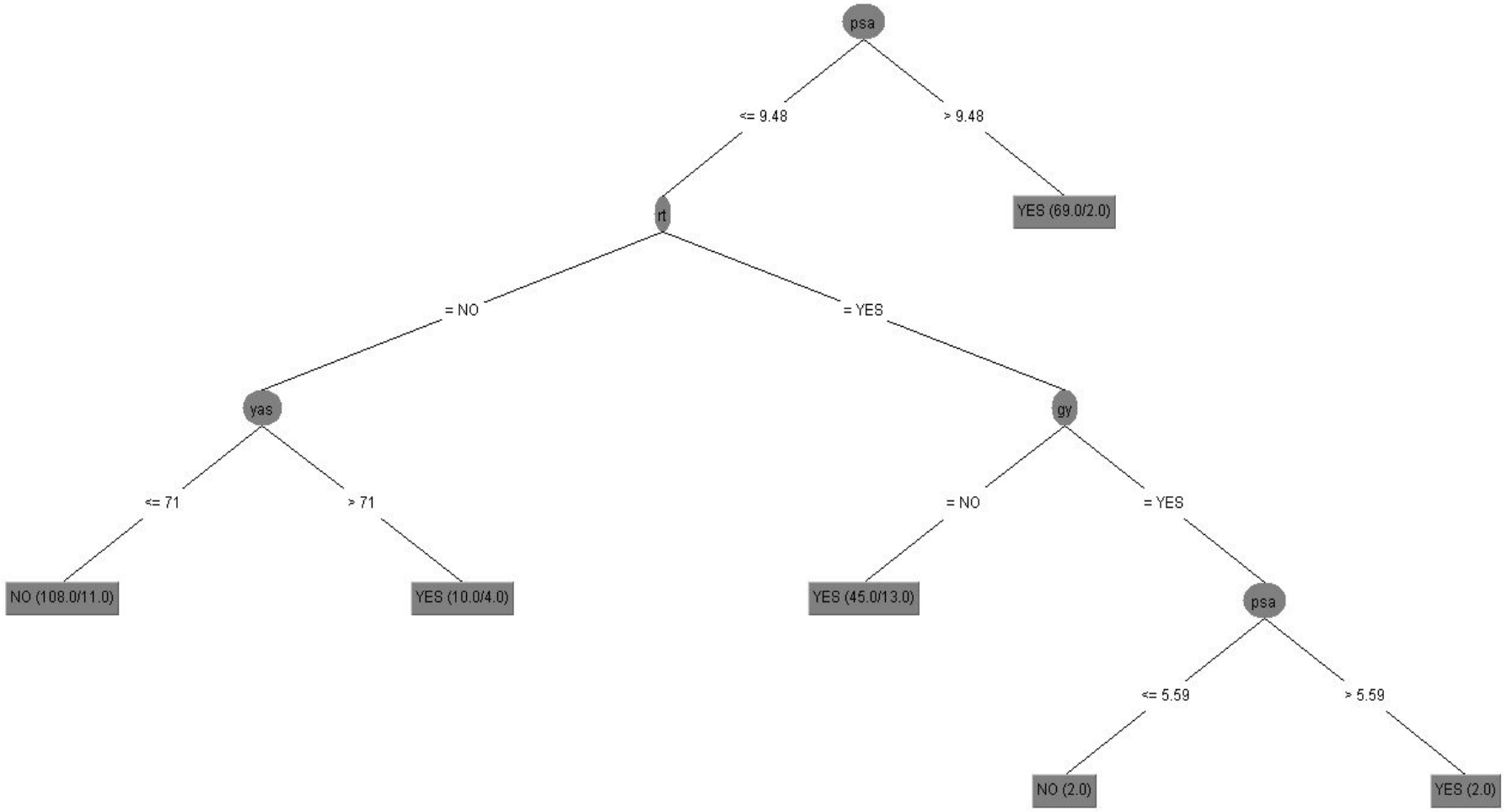
C&RT Algoritması

PSA <= 9.48
| rt = NO
| | yaş <= 71: NO (108.0/11.0)
| | yaş > 71: YES (10.0/4.0)
| rt = YES
| | gy = NO: YES (45.0/13.0)
| | gy = YES
| | | PSA <= 5.59: NO (2.0)
| | | PSA > 5.59: YES (2.0)
PSA > 9.48: YES (69.0/2.0)

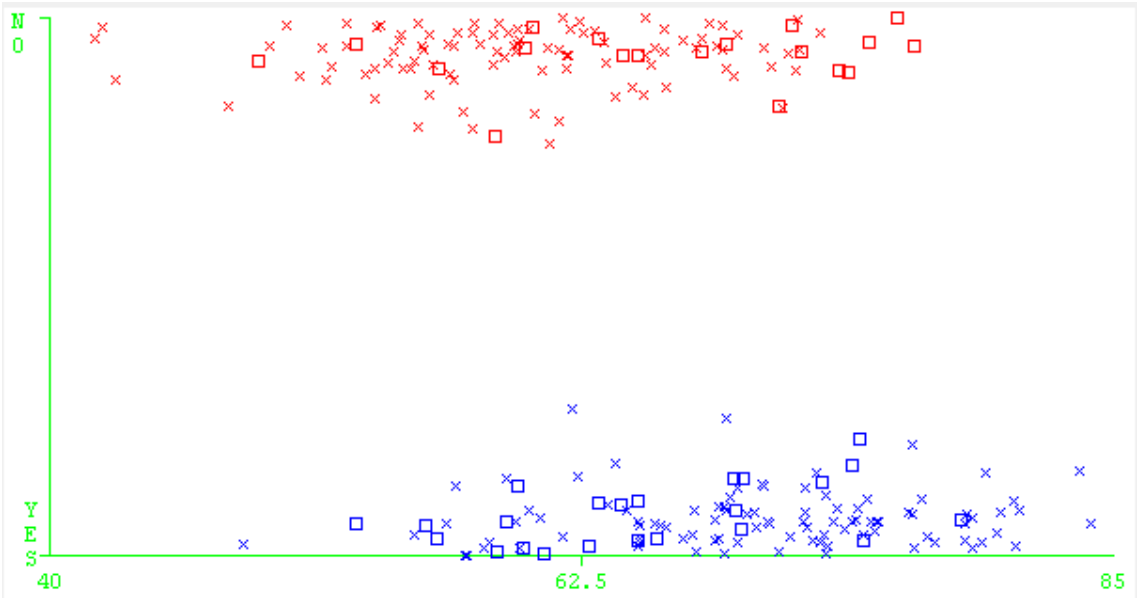
Yukarıda C&RT budanmış ağaç algoritması verilmiştir. Bu karar ağacının boyutu 11, çıkış sayısı ise 6'dır.

Veri seti için C&RT yönteminin ürettiği algoritma şu şekildedir.

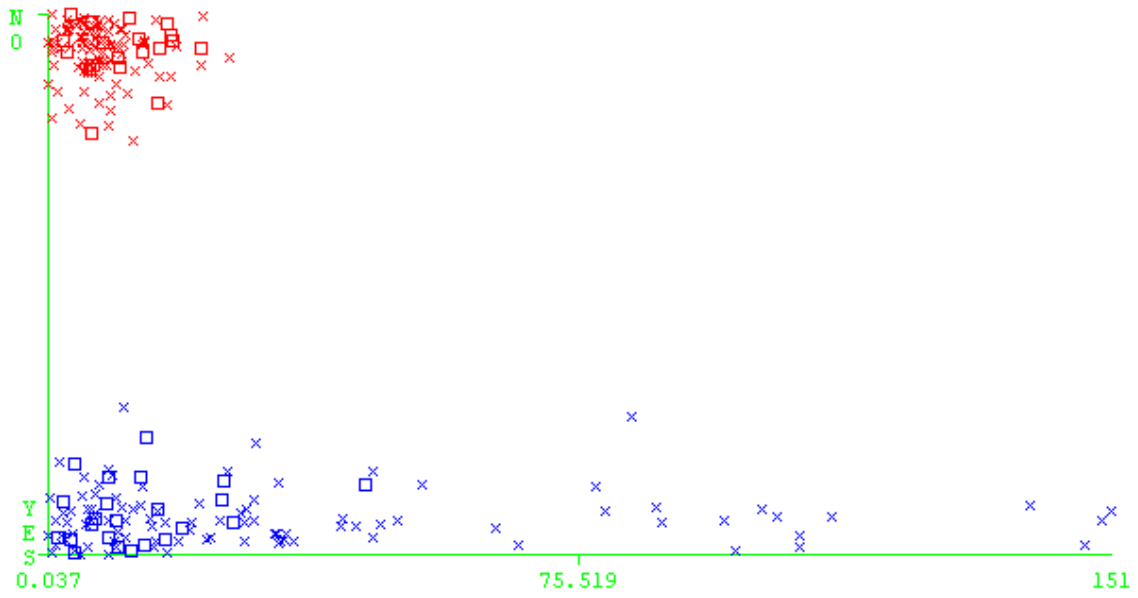
i. **PSA** değeri 9,48'den büyük eşit olanları kanserdir.
ii. **PSA** değerleri 9,48'den küçük olanların;
    a. **RT** değeri negatif olanlardan;
        i. **Yaşı** 71'den büyük olanlar kanser,
        ii. **Yaşı** 71'den küçük eşit olanlar kanser değil,
    b. **RT** değeri pozitif olanlardan;
        i. **Genetik Yatkınlığı** olmayanlar kanser,
        ii. **Genetik Yatkınlığı** olanlardan;
            1. **PSA** değeri 5,59'dan büyük olanlar kanser,
            2. **PSA** değeri 5,59'dan küçük eşit olanlar hasta değildir.



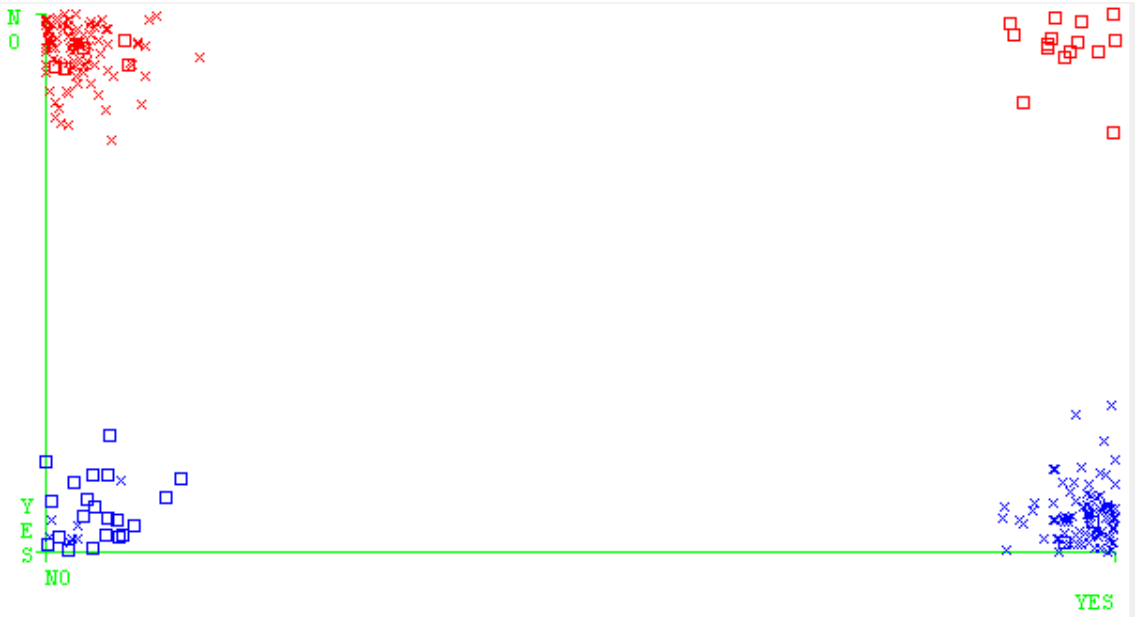
Şekil 4.13. C&RT - Algoritması İçin Ağaç Gösterimi



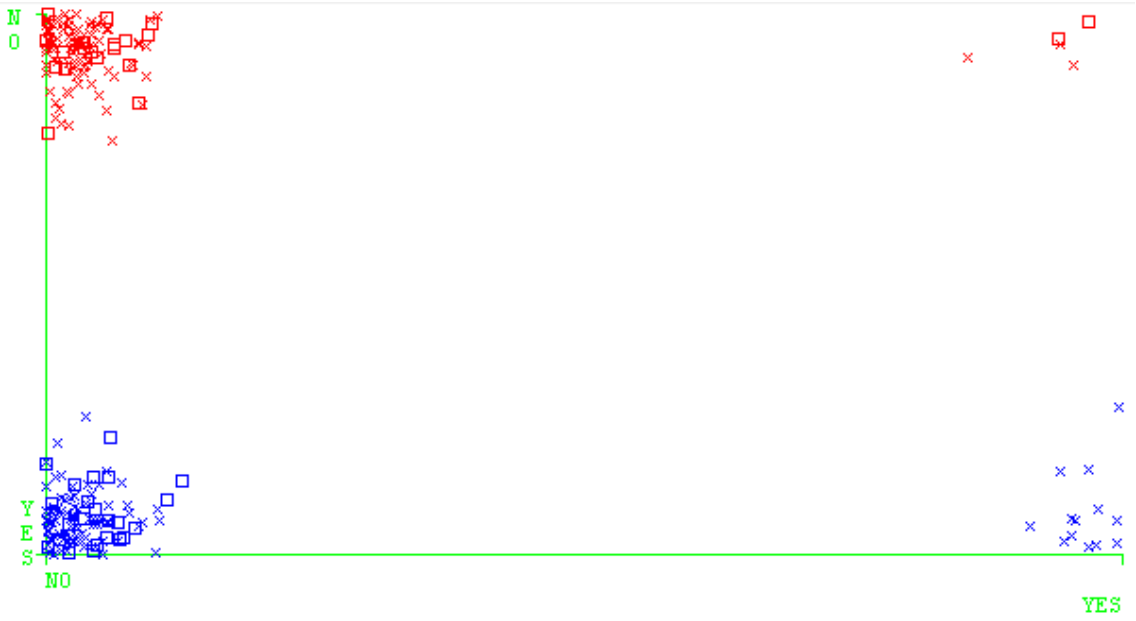
Şekil 4.14. Yaş İçin Yanlış Sınıflandırmaların Gösterimi (C&RT)



Şekil 4.15. PSA İçin Yanlış Sınıflandırmaların Gösterimi (C&RT)



Şekil 4.16. Rektal Tuşe İçin Yanlış Sınıflandırmaların Gösterimi (C&RT)



Şekil 4.17. Genetik Yatkınlık İçin Yanlış Sınıflandırmaların Gösterimi (C&RT)

4.4. YAPAY SİNİR AĞLARI

Aşağıda WEKA programına ait Yapay Sinir Ağları analizinin sonuçları verilmiştir.

Tablo 4.12. Sınıflandırma Sonuçları-I (YSA)

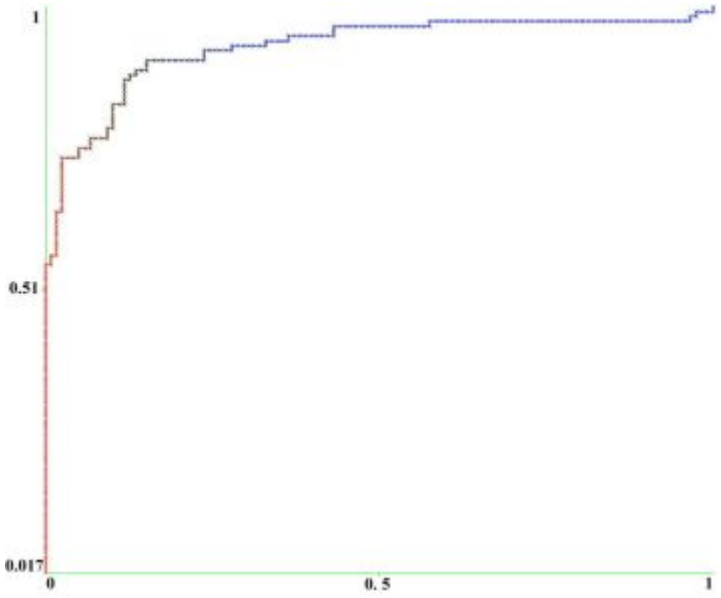
YSA	
Doğru Sınıflandırma	206 (%87.29)
Yanlış Sınıflandırma	30 (%12.71)
Kappa	0.746
Ortalama Mutlak Hata (MAE)	0.191
Ortalama Karesel Hata (RMSE)	0.311
Göreceli Mutlak Hata (REA)	%38.15
Göreceli Karesel Hata (RRSE)	%62.09
Toplam Örnek Sayısı	236

Tablo 4.13. YSA Sınıflandırma Sonuçları-II (YSA)

YSA			
Keskinlik	Duyarlılık	F-Ölçütü	AUC
%85.50	%89.80	%87.60	0.929

Tablo 4.14. Düzensizlik Matrisi Sonuçları (YSA)

YSA		
a	B	Sınıflama
106	12	A=YES
10	100	B=NO



Şekil 4.18 YSA için ROC Eğrisi

Tablo 4.15. Düğüm 0 İçin Ağırlıklar

Sigmoid Düğüm 0	
Girdiler	Ağırlıklar
Eşik	-0.122
Düğüm 2	-2.806
Düğüm 3	6.522
Düğüm 4	2.976

Tablo 4.16. Düğüm 1 İçin Ağırlıklar

Sigmoid Düğüm 1	
Girdiler	Ağırlıklar
Eşik	0.122
Düğüm 2	2.806
Düğüm 3	-6.523
Düğüm 4	-2.976

Tablo 4.17. Düğüm 2 İçin Ağırlıklar

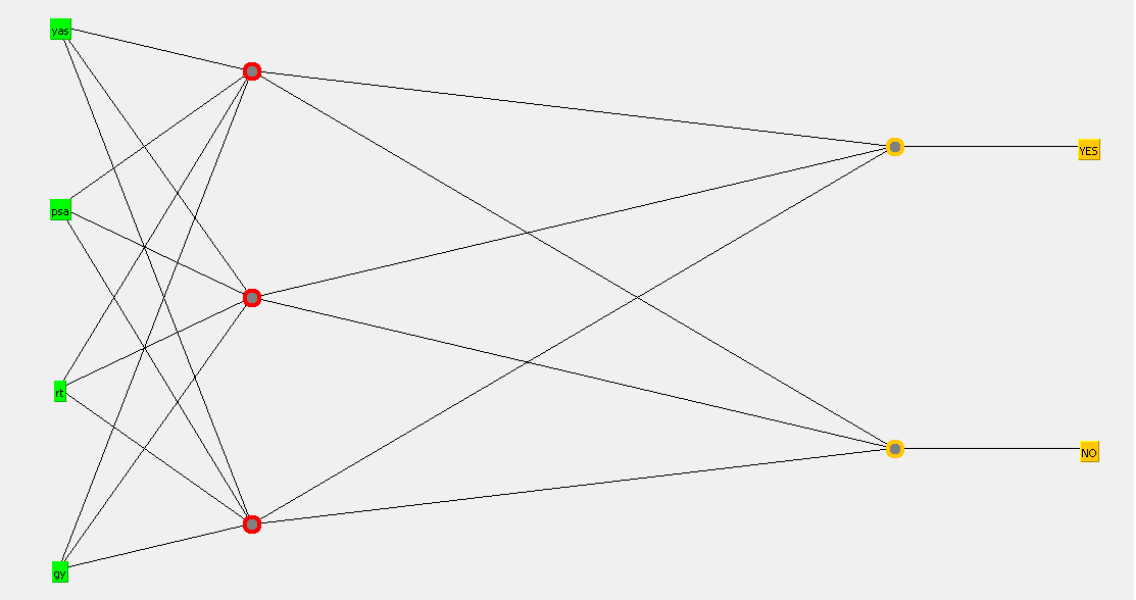
Sigmoid Düğüm 2	
Girdiler	Ağırlıklar
Eşik	-4.401
Attrib YAS	-7.481
Attrib PSA	-10.345
Attrib RT	-1.256
Attrib GY	1.671

Tablo 4.18. Düğüm 3 İçin Ağırlıklar

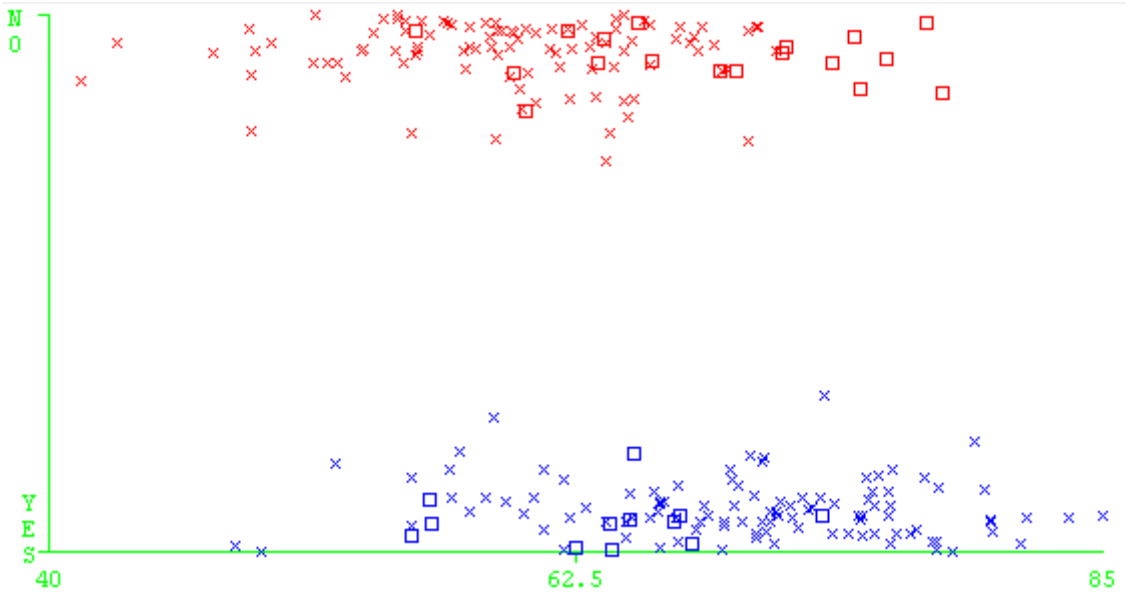
Sigmoid Düğüm 3	
Girdiler	Ağırlıklar
Eşik	9.148
Attrib YAS	-2.291
Attrib PSA	11.942
Attrib RT	0.395
Attrib GY	0.463

Tablo 4.19. Düğüm 4 İçin Ağırlıklar

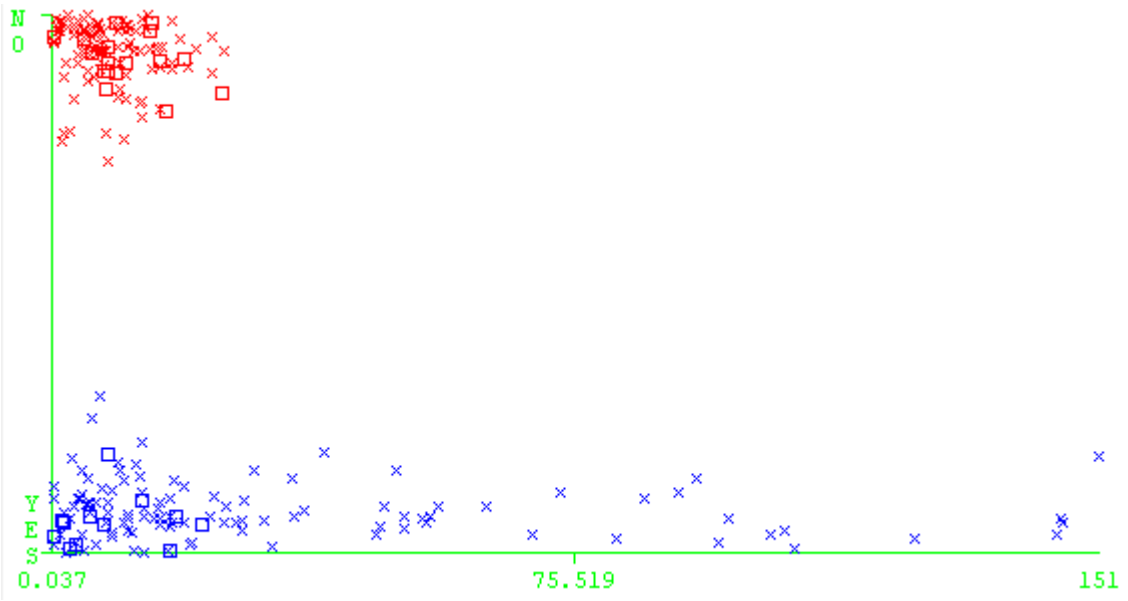
Sigmoid Düğüm 4	
Girdiler	Ağırlıklar
Eşik	8.652
Attrib YAS	11.297
Attrib PSA	9.073
Attrib RT	3.913
Attrib GY	-0.643



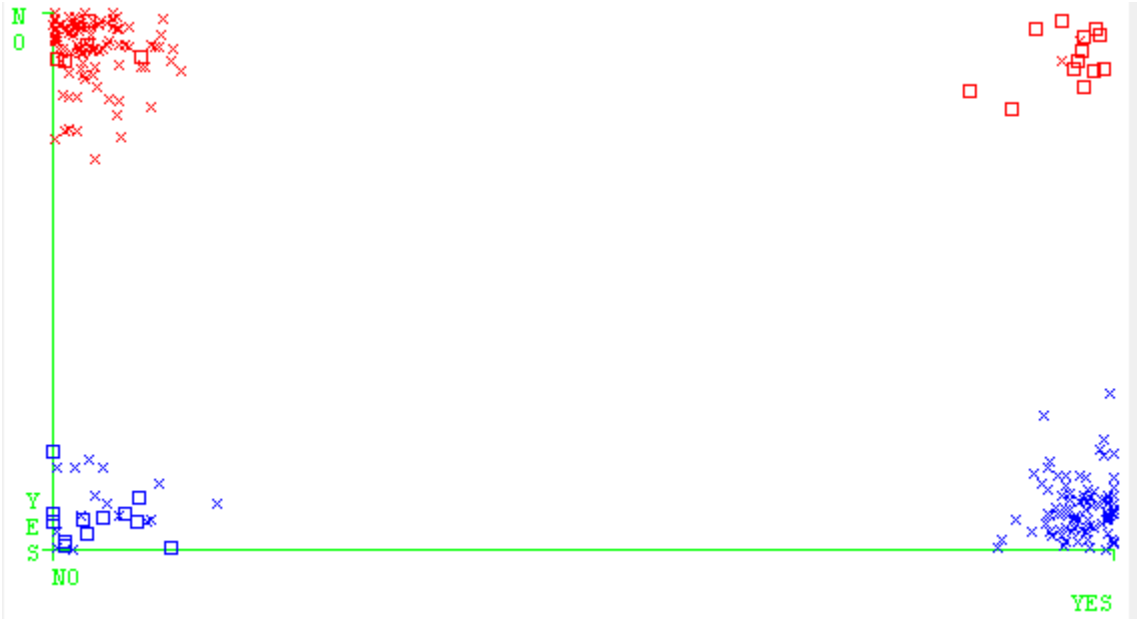
Şekil 4.19 Yapay Sinir Ağı Modeli



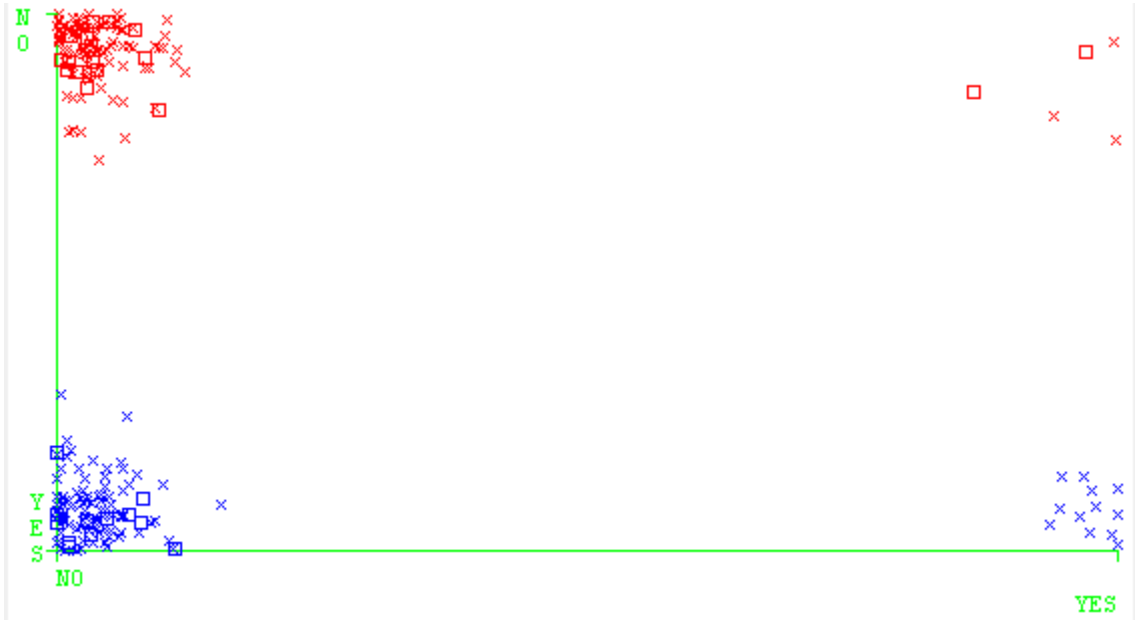
Şekil 4.20 Yaş İçin Yanlış Sınıflandırmaların Gösterimi (YSA)



Şekil 4.21 PSA İçin Yanlış Sınıflandırmaların Gösterimi (YSA)



Şekil 4.22 Rektal Tuşe İçin Yanlış Sınıflandırmaların Gösterimi (YSA)



Şekil 4.23 Genetik Yatkınlık İçin Yanlış Sınıflandırmaların Gösterimi (YSA)

4.5. YÖNTEMLERİN KARŞILAŞTIRILMASI

LRA, C&RT ve YSA yöntemlerini karşılaştırabilmek için aşağıda genel tablo oluşturulmuştur.

Tablo 4.20. Genel Karşılaştırma Tablosu

Yöntem	Doğru Sınıflandırma Oranı (%)	Hatalı Sınıflandırma Oranı (%)	Kappa İstatistiği	F-Ölçütü (%)	AUC
Lojistik	83.90	16.10	0.678	83.3	0.924
C&RT	81.78	18.2	0.636	81.5	0.828
YSA	87.29	12.71	0.746	87.6	0.929

5. TARTIŞMA ve SONUÇ

Veri madenciliği sınıflandırma modellerinden Lojistik Regresyon Analizi, Yapay Sinir Ağları ve Sınıflandırma ve Regresyon Ağaçları'nın Tokat Gaziosmanpaşa Üniversitesi Tıp Fakültesi hastanesi veritabanından çekilen Üroloji Polikliniği hastalarına ait veri seti üzerinde karşılaştırılması yapılmıştır. Bu veri seti oluşturulurken 'prostat kanseri' kesin tanısı konmuş hastalar yaklaşık beş yıllık bir veri yığını içinden süzülmüştür. Sonuç olarak kullanabileceğimiz 118 adet kesin tanısı prostat kanseri olan hasta bulunmuştur. Bu hastalara ait muayene notları içerisinden prostat kanseri teşhisinde kullanılan yaş, genetik yatkınlık, rektal tuşe kontrolü ve PSA değerleri tespit edilmiştir. Bu 118 adet hastaya ek olarak yine aynı veritabanından muayene notlarında yukarıdaki parametrelere sahip 118 adet prostat kanseri tanısı olmayan hasta tespit edilerek toplamda 236 adet hastaya ait bir veri seti elde edilmiştir.

Çalışmada bağımlı değişken prostat kanseri tanısı olarak alınmış, bağımsız değişkenler ise yaş, genetik yatkınlık, rektal tuşe kontrolü ve PSA değeri olarak tesbit edilmiştir. LRA için belirlenen bağımsız değişkenler YSA için girdi olarak kabul edilmiş ve çok katmanlı yapay sinir ağı modeli buna göre oluşturulmuştur. Yine aynı şekilde bağımlı değişken prostat kanseri üzerine bağımsız değişkenler ile ağaç oluşturulmuş ve C&RT analizi gerçekleştirilmiştir. Sonuçların regresyon ağacı diyagramı şeklinde görüntülenmesi sınıflandırmanın anlaşılabilir ve yorumlanabilir olması bakımından önemlidir. LRA da, bağımsız değişkenlerin aldığı değerler ile sınıflayıcı ya da sıralayıcı ölçek yapısında olan birimlerin, bağımlı değişkene göre sınıflandırılması yapılabilmektedir. Ayrıca LRA yardımı ile bağımsız değişkenler ile bağımlı değişken arasındaki ilişki risk yönünden incelenebilmektedir. Genel olarak YSA ise beynin işlevini yerine getirme yöntemini modellemek için tasarlanan

matematiksel bir sistem yardımı ile oluşturulan model üzerinden sınıflama işlemi yapmaktadır. Karmaşık yapısı nedeniyle YSA'nın oluşturulması ve değerlendirilmesi diğer yöntemlere göre daha zor olduğu görülmüştür.

Bu çalışmada, modellerin oluşturulması için ücretsiz bir yazılım olan ve veri madenciliği algoritmaları üzerinde geniş bir yelpazede analiz şansı sunan WEKA programı tercih edilmiştir.

Tanımlayıcı istatistikler Tablo 4.3. Prostat Kanseri Tanısı Durumuna Göre Sürekli Değişkenlerin (Yaş ve PSA) Dağılımı'na göre prostat kanserli grup ile olmayan grup arasında yaş ortalamaları bakımından anlamlı bir fark olduğu gözlemlenmektedir ($p<0.001$).Yaş arttıkça prostat kanserine yakalanma riski artmaktadır. Literatüre göre ise 70 yaş üzerine çıkıldığında en yüksek hastalanma oranı ile karşılaşılmaktadır. Yine aynı tabloda prostat kanserli grup ile olmayan grup arasında PSA düzeyi bakımından anlamlı bir fark bulunmuştur ($p<0.001$). PSA düzeyi arttıkça prostat kanseri vakaları daha fazla görülmektedir.

Tanımlayıcı istatistikler Tablo 4.4. Prostat Kanseri Tanısı Durumuna Göre Kategorik Değişkenlerin (Rektal Tuşe ve Genetik Yatkınlık) Dağılımı'na göre hastaların prostat kanseri durumu ile rektal tuşe durumları arasında anlamlı bir ilişki gözlemlenmektedir ($p<0.001$). Rektal tuşe kontrolü pozitif olan hastalarda prostat kanseri olma yüzdesi olmayanlara göre daha fazladır. Yine aynı tabloya göre genetik yatkınlık ile prostat kanseri arasında istatistiksel açıdan anlamlı bir ilişki bulunamamıştır.

Prostat kanseri teşhisi konulan hastalar için genetik yatkınlık oranına bakıldığında %11'lik bir oran vardır. Literatür çalışmasında bu oranın genellikle %1 ile

%5 oranında değiştiği görülmektedir. Bu açıdan çalışma yapılan hastalar için ortaya çıkan genetik yatkınlık oranı normallerin üzerindedir.

Yöntemlerin sınıflandırma başarıları açısından veri seti üzerinde en iyi sınıflandırmayı Yapay Sinir Ağları (YSA) gerçekleştirmiştir.%87.29 doğruluk ve 0.929 AUC ile Yapay Sinir Ağları algoritmasına girilen kayıtlar diğer yöntemlere göre daha doğru şekilde sınıflandırılmıştır. İkinci sırayı ise %83.90 doğruluk ve 0.924 AUC ile Lojistik Regresyon Analizi yöntemi, üçüncü sırayı ise %81.78 doğruluk ve 0.828 AUC ile C&RT Karar Ağacı algoritması almıştır.

Yine yöntemlerin Kappa İstatistiği yönünden karşılaştırılmasında YSA 0,746 ile ilk sırada, LRA 0.678 ile ikinci ve C&RT 0.636 ile üçüncü sıradadır.

Literatüre baktığımızda ise yaptığımız çalışma sonucu ile paralellik görmekteyiz. Karşılaştırdığımız yöntemler ile yapılan birçok çalışmada güvenirliği ile YSA ön plana çıkmıştır.

Bartfay ve arkadaşları [69] YSA ve LRA kullanarak yaptıkları çalışmada doğru sınıflandırma oranların karşılaştırmayı amaçlamışlar beş farklı lojistik regresyon ve üç faklı YSA modeli arasında doğru sınıflandırma oranları en iyi olan LRA ve YSA modellerini almışlar ve LRA için doğru sınıflandırma oranını %65, YSA için doğru sınıflandırma oranını ise %67 olarak hesaplamışlardır.

Karakış'ın[7] yaptığı çalışmada, meme kanseri hastalarının koltuk altı lenf nod durumlarını belirleyen SLNB ve AD ameliyatları olmaksızın, her hastanede kolaylıkla elde edilebilir olan klinik ve patolojik verilerinin girildiği YSA' nın, hastaların koltuk altı lenf nod durumunu belirlemesi amaçlanmıştır. Çalışma için Ankara Numune Eğitim ve Araştırma Hastanesi ve Ankara Onkoloji Eğitim ve Araştırma Hastanesi'ne başvuran

ve meme kanseri 270 kişinin verileri kullanılmıştır. Lojistik regresyon ve seçilen YSA modelleri kıyaslandığında YSA değerleri daha başarılı olduğu görülmüştür.

Ocakoğlu'nun[6] yaptığı çalışmada, lojistik regresyon analizi ve yapay sinir ağlarının sınıflama etkinliklerini karşılaştırmayı amaçlamaktadır. Lojistik regresyon analizi ve yapay sinir ağları yöntemleri, bireylerin sınıflandırma oranlarına göre karşılaştırılmıştır. Buna göre YSA modelleri ile sınıflandırmanın LRA kullanılarak yapılan sınıflandırmadan daha iyi sonuçlar verme eğiliminde olduğu ayrıca yine aşırı eğitme, mimarinin hatalı oluşturulması vb. problemleri olmayan YSA modellerinin daha iyi öngörü performansı sağlayabildiği görülmüştür.

Kullanılabilecek istatistik analizleri karşılaştırıldığımızda farklı ölçütlere göre farklı analizlerin başarılı olduğu görülmüştür.

Doğru sınıflama oranlarına göre en iyi modelin Yapay Sinir Ağları sonra Lojistik Regresyon Analizi ve en son olarak C&RT bulunmuştur. Buradaki çalışmamızda kanser vakalarının değerlendirilmesi yapılırken bir model yerine birden fazla model kullanılmasının belki bir çözüm olabileceği bulunmuştur. Çünkü açıklanma yüzdelerine bakıldığında farklı veri setlerinde farklı sonuçlar bulunabileceği düşünülmektir. Çalışmamız için seçicilik bakımından her ne kadar Yapay Sinir Ağaçları ile elde edilen sonuçların daha iyi olduğu elde edilmiş olsa bile daha fazla bilgi toplayarak yeni bir karar alma yoluna gidebilir.

KAYNAKLAR

1. Akpınar H., "Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği", İ.Ü. İşletme Fakültesi Dergisi 2000, 29: 1-22

2. Köktürk F., Ankaralı H., Sümbüloğlu V., "Veri Madenciliği Yöntemlerine Genel Bakış", Türkiye Klinikleri J Biostat 2009, 1(1): 20-5

3. Türe M., Ömürlü K.. "Sınıflandırma Yöntemlerinin Performanslarının Karşılaştırılmasına İlişkin Simülasyon Çalışması", (2009).

4. Güneri N., Apaydın A., "Öğrenci Başarılarının Sınıflandırılmasında Lojistik Regresyon Analizi ve Sinir Ağları Yaklaşımı", Ankara (2008).

5. Kurt İ., Türe M., "Tıp Öğrencilerinde Alkol Kullanımını Etkileyen Faktörlerin Belirlenmesinde Yapay Sinir Ağları ile Lojistik Regresyon Analizi'nin Karşılaştırılması", Trakya Üniversitesi Tıp Fak Dergisi 2005, 22(3):142-153.

6. Ocakoğlu G., "Lojistik Regresyon Analizi ve yapay Sinir Ağları Yöntemlerinin Sınıflama Özelliklerini Karşılaştırılması ve Bir Uygulama", Yüksek Lisans Tezi, Uludağ Üniversitesi, Bursa (2006).

7. Karakış R., "Yapay Sinir Ağları ve Lojistik Regresyon Yöntemleri ile Meme Kanseri Koltuk Altı Lenf Durumunun Belirlenmesi", Yüksek Lisans Tezi, Gazi Üniversitesi, Ankara (2009).

8. Kıran Z., "Lojistik regresyon ve C&RT Analizi Yöntemleriyle Sosyal Güvenlik Kurumu İlaç Provizyon Sistemi Üzerinde Bir Uygulama", Yüksek Lisans Tezi, Gazi Üniversitesi (2010).

9. Wu X., Kumar V., QuinlANN J., Ghosh J., Yang Q., "Top 10 Algorithms In Data Mining". Knowledge of Information Systems 2008, 14: 1-37.

10. Sabzevari H., Soleymani M., Noorbakhsh E., “A Comparison Between Statistical and Data Mining Methods for Credit Scoring in Case of Limited Available Data”, Eleventh ANNual APRIA Conference 2007.

11. Zurada J., Lonial S., “Comparison Of The Performance Of Several Data Mining Methods For Bad Debt Recovery In The Healthcare Industry”, The Journal of Applied Business Research 2005, 21: 37-53.

12. Kaya E., Bulun, M., Arslan, A., “Tıpta Veri Ambarları Oluşturma ve Veri Madenciliği Uygulamaları”, Akademik bilişim 2003, Çukurova Üniversitesi, Adana, (2003)

13. Berry M., Linoff G., “Data Mining Techniques for Marketing Sales and Customer Support”, John Wiley & Sons, 1997 2-12.

14. Giudici P., “Applied Data Mining: Statistical Methods for Business and Industry 1st ed.”, John Wiley & Sons, England, 1-15, 85-110 (2003).

15. Holsheimer M., Siebes A., “Data mining: The search for knowledge in databases.”, Technical Report , CWI, Netherlands, 12 (1994).

16. Jacobs P., ”Data Mining: What general managers need to know”, Harvard Management Update, 4 (10): 8-9 (1999).

17. Fayyad, U., Piatetsky-Shapiro G., Smyth P., “From Data Mining to Knowledge Discovery in Databases,” American Association for Artificial Intelligence, 3(17): 37-54 (1996).

18. Hand, J., “Data mining: statistics and more ?”, The American Statistician, 52: 112-118 (1998).

19. Altıntaş, Y., "Veri Madenciliğinin Tıpta Kullanımı Ve Bir Uygulama : Hemodiyaliz Hastaları İçin Risk Seviyelerine Göre Risk Faktörlerinin Etkileşimlerinin İncelemesi", Yüksek Lisans Tezi, Gazi Üniversitesi 2010.

20. Han J., Kamber M., "Data Mining Concepts and Techniques 2nd Ed.", Editor : Jim Grey, The Morgan KaufmANN Series in Data Management Systems, Morgan KaufmANN 2,8,12,14,15,29,30,398-403, (2006)

21. Çetinyokuş, T., "Veri Küplerinin Bütünleşik Kullanımına Yönelik Yeni Bir OLAP Mimarisi", Yüksek Lisans Tezi, Gazi Üniversitesi 2008.

22. KDnuggets, "In what industries/sectors were your data mining clients in 2007-2008?", 2008.

23. Kantardzic M., "Chapter 9: Artificial Neural Networks Chapter 1-1.4", DataMining Concepts, Models, Methods and Algorithms, John Wiley & Sons,(2003)

24. Hastie T., Tibshirani R., Friedman J., "The Elements of StatisticalLearning; Data Mining, Inference and Prediction", Springer Series in Statistics, New York, USA, 533 (2001).

25. Maimon R., "Data Mining & Knowledge Discovery Handbook" ,Springer, 334 (2005).

26. Bigus J.P., "Data Mining With Neural Networks: Solving BusinessProblems from Application Development to Decision Support",McGrawHill, (1996).

27. Silahtaroğlu, G., "Kavram ve Algoritmalarıyla Temel Veri Madenciliği", Papatya Yayıncılık Eğitim, İstanbul, 33, 45-47, 58 (2008).

28. Aydoğan, F., " E-Ticarette Veri Madenciliği Yaklaşımlarıyla Müşteriye Hizmet Sunan Akıllı Modüllerin Tasarımı ve Gerçekleştirimi", Hacettepe Üniversitesi, Ankara, (2003)

29. Berry, M. J., Linoff, G. S., "Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management 2nd ed.", Wiley, USA, (2004).

30. Pehlivan, G., "Chaid Analizi ve Bir Uygulama", Yıldız Teknik Üniversitesi, İstanbul, (2006).

31. Thomas, Lyn. C., "A Survey of Credit and Behavioral Scoring: Forecasting Financial Risk of Lending to Consumer", International Journal of Forecasting, 16 (2): 149–172 (2000).

32. Temel, G. O., Çamdeviren, H., Akkuş, Z., "Sınıflama Ağaçları Yardımıyla Restless Legs Syndrome (RLS) Hastalarına Tanı Koyma", İnönü Üniversitesi Tıp Fakültesi Dergisi, 12 (2): 111-117 (2005).

33. Argüden, Y., Erşahin, B., "Veri Madenciliği : Veriden Bilgiye, Masraftan Değere", ARGE Danışmanlık, 48-63, (2008)

34. Vahaplar, A., " Bir Coğrafi Veri Madenciliği Uygulaması", Yüksek Lisans Tezi, Ege Üniversitesi, İzmir, (2003)

35. Masseglia, F., Poncelet, P., Teisseire, M., "Using Data Mining Techniques on Web Access Logs to Dynamically Improve Hypertext Structure", ACM Sigweb Newsletter, 8 (3): 1-19 (1999).

36. Teng, J., Lin, K., Ho, B., "Application of Classification Tree and Logistic Regression for The Management and Health İntervention Plans in ACommunity-Based Study", Journal of Evaluation in Clinical Practice, 13 : 741-748 (2007)

37. Deconinck, E., Hancock, T., Coomans, D., Massart, D.L., Heyden, Y.V., "Classification of drugs in absorption classes using the classification and regression trees (C&RT) methodology", Journal of Pharmaceutical and Biomedical Analysis, 39 : 91–103 (2005).

38. Teng, J., Lin, K., Ho, B., "Application of Classification Tree and Logistic Regression for The Management and Health İntervention Plans in A Community-Based Study", Journal of Evaluation in Clinical Practice, 13 : 741-748 (2007)

39. King R. D., Feng, C., Sutherland, A., "StatLog: Comparison of Classification Algorithms on Large Real-World Problems; Applied Artificial Intelligence", Vol 9, Issue 3 , 289-333 (1995)

40. Ediz B., "Lojistik Regresyon-Ayırma Analizi, Ayrımsama Sorunu ve Kalp Hastalarında Lojistik Model Yardımıyla Risk Ölçütlerinin Belirlenmesi", Uludağ Üniversitesi, Bursa (1997).

41. Tatlıdil H., "Uygulamalı Çok Değişkenli İstatistiksel Analiz", Engin Yayınları, Ankara, 11, 122, 252–260 (1992).

42. Hosmer, D. W., Lemeshow, S., "Applied Logistic Regression", John Wiley & Sons, New York, 5-50 (1989).

43. Freeman, D.H., "Logistic Regression", Applied Categorical Data Analysis, Marcel Dekker Inc., New York, 238 (1987).

44. Şahin, Ş.Ö., "Yapay Sinir Ağları Yardımı ile Dinamik Bir Senaryo Analizi", İTÜ, İstanbul, (2001).

45. Efe, M.Ö., Kaynak O., "Yapay Sinir Agları ve Uygulamaları", Yüksek Lisans Tezi, İstanbul Boğaziçi Üniversitesi, s.1, (2000).

46. Tosun S., "Sınıflandırmada Yapay Sinir Ağları Ve Karar Ağaçları Karşılaştırması: Öğrenci Başarıları Üzerine Bir Uygulama", İTÜ, (2007).

47. Fausett L.V., "Fundamentals Of Neural Networks", Printice-Hall Inc., New Jersey,s. 40-45, (1994).

48. Öztemel E., "Yapay Sinir Ağları", Papatya Yayıncılık, İstanbul, s.32-33, (2003).

49. Sağıroğlu Ş., Beşdok E., Erler M., "Mühendislikte Yapay Zeka Uygulamaları I: Yapay Sinir Ağları", Ufuk Kitap, Kayseri, (2003).

50. Seven A., "Yapay Sinir Ağları ile Doku Sınıflandırma", İ.T.Ü, İstanbul, (1993).

51. Judith E. D., "NeuralNetwork Architectures An Introduction", Van Nostrand Reinhold, NewYork, s. 1-6, (1990).

52. Taş, E., "Yapay sinir ağlarında momentumlu dik iniş ve eşlenik gradyan eğitim algoritmalarının karşılaştırılması", Yüksek Lisans Tezi, Anadolu Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir, 5-61 (2005).

53. Haykin, S., "Neural networks: a comprehensive foundation", Prentice Hall, USA, 1–50, 117-156,156-256 (1999).

54. Elder, J.F., Abbot, D. W., "A Comparison of Leading Data Mining Tools; Fourth International Conference on Knowledge Discovery& Data Mining", New York, (1998).

55. Elmas Ç., "Yapay Sinir Ağları", Seçkin Yayıncılık, Ankara, s.31-32, (2003).

56. Öztemel E., "Yapay Sinir Ağları", Papatya Yayıncılık, İstanbul, s.49, (2003).

57. Efe, M.Ö., Kaynak O., "Yapay Sinir Agları ve Uygulamaları", İstanbul Boğaziçi Üniversitesi, s.7, (2000).

58. Bishop C.M., "Neural Networks For Pattern Recognition", Clarendon Press, Oxford, (1995).

59. Wang S., "An Adaptive Approach To Market Development Forecasting", Neural Computing & Applications 8, s.3-8, (1999).

60. Binici E., "Java ile yapay zeka mekanizmasına sahip bir ağ yönetim sistemi geliştirilmesi", Yüksek Lisans Tezi, Ege Üniversitesi, İzmir, 28-48 (2006).

61. Wigle D.T., Turner M.C., Gomes J., “Role of hormonal and other factors in human prostate cancer”. J Toxicol Environ Health B Crit Rev 11 (3-4): 242-59, (Mart 2008).

62. "Cancer Facts & Figures 2008", American Cancer Society, (2008).

63. Huggins C., Steven R.E., Hodges C.V.,” Studies on prostatic cancer. Arch. Sug.” 43:209-223, (1941).

64. Frank E., Hall M., Holmes G., Kirkby R., Pfahringer B., Witten, I.H., “WEKA: A Machine Learning Workbench for Data Mining”, University of Waikato, New Zealand, 7-10 (2004).

65. Witten I.H., Frank E., “Data Mining: Practical Machine Learning Tools and Techniques 2nd ed.”, Morgan Kaufmann, USA, 365-415 (2005).

66. Dawson Saunders B, Trapp Robert G., “Basic& Clinical Biostatistics”, London, s:32-33, 116, (1994).

67. Hosmer D.W., Lemeshow S., “Applied logistic regression”. 2nd ed. New York: John Wiley & Sons; (2000).

68. Dirican A. Evaluation of the diagnostic test’s performance and their comparisons. Cerrahpasa J Med ;32:25-30 , (2001).

69. Bartfay et al., “Comparing the predictive value of neural network models to logistic regression models on the risk of death for small-cell lung cancer patients”, European Journal of Cancer Care 15 , s:115–124, (2006).

EKLER

Ek 1. Tokat Gaziosmanpaşa Üniversitesi Tıp Fakültesi Hastanesi Üroloji Polikliniği'ne 01.01.2005 - 31.05.2011 tarihleri arasında başvuran 236 hastaya ait veri seti. (Ekteki veri setinin MS Excel .cvs formatıdır)

age,PSA,rt,gy,pk

70,7.62,NO,NO,YES 68,33.27,NO,NO,YES 73,6.65,YES,NO,YES 62,115.6,YES,NO,YES

59,0.037,NO,NO,YES 72,151,YES,NO,YES 66,43.57,YES,NO,YES 58,3.29,YES,NO,YES

82,151,YES,YES,YES 73,4.3,YES,NO,YES 63,4.53,YES,NO,YES 69,12.47,NO,NO,YES

80,14.5,YES,NO,YES 57,10.27,YES,YES,YES 73,43.02,YES,YES,YES 74,3.88,NO,NO,YES

68,6.29,NO,NO,YES 66,7.44,YES,YES,YES 77,100,YES,NO,YES 68,12.85,YES,NO,YES

61,49.16,NO,NO,YES 73,6.74,NO,NO,YES 70,10.84,YES,NO,YES 67,1.92,NO,NO,YES

77,16.89,YES,YES,YES 59,3.94,YES,NO,YES 80,9.78,YES,NO,YES 73,8.22,YES,NO,YES

60,4.73,YES,NO,YES 72,1.53,YES,NO,YES 78,13.37,YES,NO,YES 66,100,YES,NO,YES

68,74.1,YES,NO,YES 80,26.62,YES,NO,YES 57,37.35,YES,NO,YES 78,100,YES,NO,YES

75,1.53,NO,NO,YES 77,1.4,NO,NO,YES 71,47.39,YES,NO,YES 75,20.66,YES,NO,YES

55,45.87,YES,NO,YES 64,1.41,YES,NO,YES 74,23.2,NO,NO,YES 60,9.81,NO,NO,YES

48,32.07,YES,YES,YES 63,11.31,YES,NO,YES 50,16.31,NO,NO,YES 71,4.5,YES,NO,YES

62,17.87,YES,NO,YES 63,11.65,YES,NO,YES 80,8.8,YES,YES,YES 70,9.09,YES,NO,YES

76,54.99,YES,NO,YES 60,0.587,YES,NO,YES 77,5.12,NO,NO,YES 73,85.53,YES,NO,YES

57,3.3,YES,NO,YES 70,0.91,YES,NO,YES 72,13.48,YES,NO,YES 73,24.98,YES,NO,YES

79,88.14,NO,NO,YES 55,3.21,YES,NO,YES 60,0.789,NO,NO,YES 66,15.7,YES,NO,YES

70,151,YES,NO,YES 79,3.32,YES,NO,YES 67,8.73,NO,NO,YES 67,77.29,YES,NO,YES

78,3.95,YES,NO,YES 65,8.31,NO,NO,YES 72,1.39,YES,NO,YES 76,0.375,YES,NO,YES

64,1.73,YES,NO,YES 75,54.99,YES,NO,YES 64,100,YES,NO,YES 65,6.59,YES,NO,YES

75,26.79,YES,YES,YES 74,7.69,YES,NO,YES 75,68.18,NO,NO,YES 75,9.59,YES,YES,YES

76,48.13,YES,NO,YES 77,31.21,YES,NO,YES 78,3.9,YES,NO,YES 72,29.91,YES,NO,YES

78,10.43,YES,NO,YES 75,0.44,YES,NO,YES 75,35.9,YES,NO,YES 72,151,YES,NO,YES

68,6.41,YES,NO,YES 70,3.8,YES,NO,YES 62,10.29,YES,YES,YES 85,13.48,YES,YES,YES

57,5.31,NO,NO,YES 63,4.5,NO,NO,NO 60,1.48,NO,NO,NO 66,0.636,NO,NO,NO

74,24.48,YES,NO,YES
71,0.36,YES,NO,YES
75,99.63,YES,YES,YES
64,7.79,NO,NO,YES
67,5.7,NO,NO,YES
70,28.81,YES,NO,YES
66,14.69,YES,NO,YES
67,22.04,NO,NO,YES
72,4.39,NO,NO,YES
67,54.11,YES,NO,YES
80,14.79,YES,NO,YES
68,29.31,NO,NO,YES
77,81.4,YES,NO,YES
68,100,YES,NO,YES
62,13.93,YES,NO,YES
53,2.36,YES,NO,NO
56,3.53,NO,NO,NO
66,5.73,NO,NO,NO
58,0.99,NO,NO,NO
57,4.8,NO,NO,NO
70,1.38,YES,YES,NO
58,1.58,NO,NO,NO
64,7.38,NO,NO,NO
70,1.26,NO,NO,NO
51,2.84,NO,NO,NO
70,2.04,NO,YES,NO
60,1.64,NO,NO,NO
67,3.3,NO,NO,NO

75,2.57,YES,NO,NO
65,0.896,NO,NO,NO
57,1.14,NO,NO,NO
62,0.81,NO,NO,NO
66,16.43,NO,NO,NO
64,2.73,NO,NO,NO
62,2.1,NO,NO,NO
63,8.24,NO,NO,NO
58,0.482,NO,NO,NO
67,2.35,NO,NO,NO
48,0.515,NO,NO,NO
73,3.47,NO,NO,NO
65,3.02,YES,NO,NO
67,5.05,NO,NO,NO
67,2.58,YES,NO,NO
43,0.671,NO,NO,NO
68,3.05,NO,NO,NO
67,3.8,NO,NO,NO
62,3.91,NO,YES,NO
56,6.33,NO,NO,NO
58,0.576,NO,NO,NO
58,1.5,NO,NO,NO
70,1.68,NO,NO,NO
60,18.3,NO,NO,NO
60,2.47,NO,NO,NO
65,4.26,NO,NO,NO
65,3.29,NO,NO,NO
66,3.29,NO,NO,NO

62,0.621,NO,NO,NO
55,2.67,NO,NO,NO
70,4.73,NO,NO,NO
65,0.397,YES,NO,NO
68,6.98,NO,NO,NO
66,6.66,NO,NO,NO
76,3.76,YES,NO,NO
48,8.63,NO,NO,NO
62,1.99,NO,NO,NO
55,2.04,NO,NO,NO
63,1.06,NO,NO,NO
57,0.91,NO,NO,NO
69,5.76,NO,NO,NO
61,0.575,NO,NO,NO
68,3.06,NO,NO,NO
53,3.04,NO,NO,NO
54,5.3,NO,NO,NO
72,3.8,YES,YES,NO
59,6.62,YES,NO,NO
63,3.97,YES,NO,NO
65,9.48,NO,NO,NO
60,7.83,NO,NO,NO
56,1.93,YES,NO,NO
57,2.8,NO,NO,NO
58,6.84,NO,NO,NO
68,3.38,NO,NO,NO
76,4.32,NO,NO,NO
69,6.23,YES,NO,NO

49,0.869,NO,NO,NO
65,0.55,NO,NO,NO
54,6.32,NO,NO,NO
63,8.26,NO,NO,NO
59,9.19,NO,NO,NO
55,3.5,NO,NO,NO
57,2.96,NO,NO,NO
52,1,NO,NO,NO
58,1.82,NO,NO,NO
59,1.49,NO,NO,NO
72,6.54,NO,NO,NO
59,3.04,YES,NO,NO