



T.C.
BİRÜNİ ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ

BİYOİSTATİSTİK ANABİLİM DALI
BİYOİSTATİSTİK YÜKSEKLİSANS PROGRAMI

YAPAY SİNİR AĞLARI, LOJİSTİK REGRESYON VE DESTEK
VEKTÖR MAKİNESİ İSTATİSTİK YÖNTEMLERİNİN
SINIFLANDIRMADAKİ KARŞILAŞTIRILMASI

DENİZ TEZER

TEZ DANIŞMANI

Prof. Dr. M. Yusuf Çelik

İSTANBUL

2018

ONAY SAYFASI

Biruni Üniversitesi Sağlık Bilimleri Enstitüsü Biyoistatistik Anabilim Dalında Deniz TEZER tarafından hazırlanan “Yapay Sinir Ağları, Destek Vektör Makineleri ve Lojistik Regresyon Yönteminin Sınıflamada Başarılarının Kıyaslanması” adlı tez çalışması aşağıdaki jüri tarafından YÜKSEK LİSANS tezi olarak kabul edilmiştir.

Tez Savunma Tarihi: 03.08.2018

Jüri Üyesi;
(Danışman) Prof. Dr. M. Yusuf ÇELİK Biruni Üniversitesi,
Tıp Fakültesi, Biyoistatistik ve Tıp Bilişimi
Bölümü



Jüri Üyesi; Prof. Dr. Fatma ÇELİK Biruni Üniversitesi,
Sağlık Bilimleri Fakültesi, Beslenme ve Diyetetik
Bölümü



Jüri Üyesi; Prof. Dr. Handan ANKARALI İstanbul Medeniyet
Üniversitesi, Tıp Fakültesi, Tıp Bilişimi Bölümü



Tez hakkında alınan jüri kararı, Biruni Üniversitesi Sağlık Bilimleri Enstitüsü Yönetim Kurulu tarafından onaylanmıştır.



Doç. Dr. Leman ŞENTURAN
Sağlık Bilimleri Enstitü Müdürü

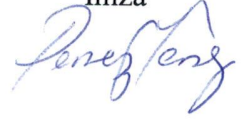
I.BEYAN

Bu tezin bana ait olduğunu, tüm aşamalarında etik dışı davranışımın olmadığını, içinde yer alan bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, kullanmış olduğum bütün bilgilere kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin yürütülmesi ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

Adı Soyadı

Deniz Tezer

İmza



II. TEŐEKKÜR SAYFASI

Bu tezin hazırlanması aŐamasında yardımlarını esirgemeyen, bana alıŐmamın her aŐamasında deneyim ve bilgileri ile yol gÖsteren danıŐman hocam Prof. Dr. M. Yusuf elik'e, hocam Prof. Dr. Fatma elik'e, sevgisi ve desteĐi ile hep yanımda olan aileme teŐekkür ediyorum.

DENİZ TEZER

III. İÇİNDEKİLER

İÇİNDEKİLER	SAYFANO
İÇ KAPAK	I
ONAY SAYFASI	II
I.BEYAN	III
II. TEŞEKKÜR	IV
III. İÇİNDEKİLER SAYFASI	V
IV. SİMGE VE KISALTMALAR LİSTESİ	VII
V.TABLO LİSTESİ	VIII
VI. ŞEKİL LİSTESİ	IX
1.ÖZET VE ANAHTAR KELİMELEER	1
2.ABSTRACT	2
3.GİRİŞ VE AMAÇ	3
4.GENEL BİLGİLER	6
4.1.Yapay Sinir Ağları	6
4.1.1.Tarihi Gelişim	7
4.1.2. İnsan Sinir Hücresi ve YSA Benzerliği	9
4.1.3. Basit Yapay Sinir Ağ Topografisi	10
4.1.4. YSA'da Sınıflamanın Yapılması	11
4.1.5. YSA Öğrenme Yöntemleri	12
4.1.5.1. Destekleyici Öğrenme	14
4.1.5.2.Danışmanlı Öğrenme	14
4.1.5.3.Danışmansız Öğrenme	14
4.1.6. YSA Aktivasyon Fonksiyonları (Etkinlik İşlevi)	16
4.1.7. YSA Kullanılan Kavramaları	18
4.1.7.1.Momentum	18
4.1.7.2.Öğrenmede İterasyon	18
4.1.7.3. Sorgulama	18
4.1.8. YSA Temel Model Yapısı	19
4.1.8.1. İleri Beslemeli Ve Çok Katmanlı Sinir Ağ Modelleri	19
4.1.8.2.Geri Beslemeli Ağlar	20
4.1.9. Çok Bilinen Ağ Yapıları ve Genel Özellikleri	24
4.2. Destek Vektör Makinası	28
4.2.1. Doğrusal Destek Vektör Makinası	29
4.2.2. Doğrusal Olmayan Destek Vektör Makinası	34
4.2.3. DVM'de Kullanılan Kernel Çekirdek Fonksiyonları	35
4.3. Logistik Regresyon Analizi	37
4.3.1. Lojistik Regresyon Yapısal Özellikleri	39
4.3.2. Logisitik Regresyon Teknikleri	41
4.3.2.1.İkili (Binary) Lojistik Regresyon Tekniği	42
4.3.2.2.Multinomial (Kategorik) Lojistik Regresyon Tekniği	42
4.3.2.3. Sıralı (Ordinal) Lojistik Regresyon Tekniği	42

4.3.2.4. Kernel Lojistik Regresyon	43
4.3.3. Lojistik Regresyon Tahmin Yöntemleri	45
4.3.3.1. En Küçük Kareler Yöntemi (RILS)	45
4.3.3.2. En Çok Olabilirlik Yöntemi (ML)	46
4.3.3.3. Minimum Lojit Ki-kare Yöntemi (MLCS)	46
4.3.4. Lojistik Regresyon Modelinde Kat Sayılarının Önemlilik Testleri	47
4.3.4.1. Olabilirlik Oran (Likelihood Ratio) Testi	47
4.3.4.2. Wald Test İstatistiği	48
4.3.4.3. Skor Test İstatistiği	48
4.3.5. Lojistik Regresyonda Model Uyumu	49
5. GEREÇ VE YÖNTEMLER	50
5.1. Veri Seçimi	51
5.2. Veri Çoğaltma Yöntemleri	52
5.3. Araştırma Yöntemlerini Karşılaştırmada Kullanılacak Ölçütler	54
5.4. Veriler ve Özellikleri	55
6. BULGULAR	58
6.1. Yapay Sinir Ağları Sınıflama Sonuçları	58
6.2. Destek Vektör Makineleri Sınıflama Sonuçları	61
6.2.1. SMO Fonksiyonlarına Göre DVM Sınıflama Başarı Dağılımı.	63
6.2.2. Libsvm Fonksiyonlarına Göre Sınıflama Başarı Oranı	64
6.3. Lojistik Regresyon Modeli Sınıflama Sonuçları	66
6.3.1. Lojistik Regresyon Sınıflama Başarı Oranı	66
6.3.2. Simple Lojistik Regresyon Sınıflama Başarı Oranı	68
6.3.3. Kernel Lojistik Regresyon Sınıflama Başarı Oranı	69
6.4. YSA, DVM ve Kernel Lojistik Regresyon Sınıflama Başarısı	71
7. TARTIŞMA	72
8. SONUÇ VE ÖNERİLER	76
9. KAYNAKÇA	78
10. EKLER	85
EK 1: WEKA programından elde edilmiş veri seti	85
11. ÖZGEÇMİŞ	91
12. İNTİHAL RAPORU	92

IV.SİMGE VE KISALTMALAR LİSTESİ

YSA : Yapay Sinir Ağları

DVM : Destek Vektör Makineleri

LR : Lojistik Regresyon

WEKA : Waikato Environment for Knowledge Analysis (İstatistik Paket Programı)

SMO: Ardışık Minimal Optimizasyon

SVM: Support Vector Machine

V. TABLOLAR LİSTESİ

Tablo 1: Sınıf Değişkenlerinin Sınıf Sayısı ve Bias'a (Yanlılık) Bağlı, Standart Sapma Ve Ortalamalarını Değişimi Tablosu.....	56
Tablo 2:Normal Dağılım Tablosu.....	57
Tablo 3: Oto-korelasyon Tablosu.....	57
Tablo 4: Çok Katmanlı Sinir Ağı Doğru Sınıflama Tablosu (Multilayer Perseptron).....	59
Tablo 5.Yapay Sinir Ağları En İyi Modelin Sınıflama Başarı Değeri.....	60
Tablo 6: YSA Modeli Karışıklık (Confusion) Matrisi.....	60
Tablo 7: Ardışık Minimal Optimizasyon (SMO) Fonksiyonlarına Göre DVM Sınıflama Tablosu.....	63
Tablo 8: Libsvm Fonksiyonlara Göre Sınıflama Tablosu.....	64
Tablo 9: Destek Vektör Makinesi En İyi Modelin Sınıflama Başarı Değerleri.....	65
Tablo 10:Destek Vektör Makinesi Karışıklık (Confusion) Matrisi.....	65
Tablo 11: Lojistik Regresyon Sınıflama Başarı Değerleri.....	66
Tablo 12:Lojistik Regresyon Kat Sayıları	67
Tablo 13: Simple Lojistik Regresyon Sınıflama Başarı Değerleri.....	68
Tablo 14: Kernel Lojistik Regresyon Sınıflama Başarı Değerleri.....	69
Tablo 15:Kernel Lojistik Regresyon Yöntemi En İyi Modelin Sınıflama Başarı Değerleri.....	70
Tablo 16: Kernel Lojistik Regresyon (Normalized Polikernel) Karışıklık (Confusion) Matrisi.....	70
Tablo 17: Yapay Sinir Ağları, Destek Vektör Makinesi Ve Kernel Lojistik Regresyon Sınıflama Başarılarının Karşılaştırılması	71
Tablo 18: Yöntemlere İlişkin Karışıklık Matrisi.....	71

VI. ŐEKİLLER DİZİNİ

Őekil 1: Nöron Yapısı ve Sinaptik Aralık.....	9
Őekil 2: Tek Katmanlı İleri Beslenmeli Sinir Ađı.....	10
Őekil 3: Üç Boyutlu verinin YSA sınıflanması.....	12
Őekil 4: Öğrenme Yöntemleri Őeması.....	14
Őekil 5: Aktivasyon Fonksiyonları.....	16
Őekil 6: İleri Beslemeli Çok Katmanlı Yapay Sinir Ađı	20
Őekil 7: Geri Beslemeli Çok Katmanlı Hopfield Ađı.....	21
Őekil 8: Yapay Sinir Ađları Őeması	23
Őekil 9: Hopfield Ađ Yapısı.....	24
Őekil 10: Destek Vektör Makineleri ve Marjin Deđerleri.....	29
Őekil 11: Destek Vektör Makineleri ve Bias Deđerleri.....	30
Őekil 12: Destek Vektör Makineleri Sınır Deđerleri ve Sınıflama.....	31
Őekil 13: Destek Vektör Makineleri Maksimum Marjin Geniřliđi.....	32
Őekil 14: Doğrusal Olmayan Destek Vektör Makineleri Gösterimi.....	35
Őekil 15: YSA modelinin görsel sunumu.....	60
Őekil 16: Lojistik Regresyon (Normalized Polikernel) ROC analizinin gösterimi...	70

1.ÖZET

Yapay Sinir Ağları, Destek Vektör Makineleri ve Lojistik Regresyon yöntemleri parametrik olmayan test yöntemleridir. Sonuç değişkeninin ikili olduğu veri setlerinde sınıflayıcı olarak kullanılır. Yapay sinir ağları insan beyninin çalışma prensibinden yola çıkılarak oluşturulmuştur. Ağ yapısı içinde yer alan sinir düğümleri giriş verilerinin istatistiksel bir takım fonksiyonlar ve eşik değerler ile işlendiği ve çıktıya dönüştüğü esas işlem merkezleridir. Geri Beslemeli Yapay Sinir Ağları algoritmasındaki döngüler her verinin birçok tekrarla düğümden geçmesini ve işlenmesini sağlayarak ağın hata değerlerini küçültür ve kendini eğiterek daha doğru sonuçlar vermesini sağlar. Yapay Sinir Ağları yöntemi çıktı değerlerini sürekli güncellediği için bilgisayar hafızasında daha az yer kaplar. Tekrarlı ölçümlenen veri yığınlarının analizinde başarılı sonuçlar veren iyi bir sınıflayıcı algoritmadır. Destek Vektör Makineleri klasik doğrusal veri ayrıştırma yönteminden farklı olarak sarmal ayrıştırıcı özelliğe sahip bir sınıflayıcıdır. Yine klasik ayrıştırıcılardan farklı olarak üç boyutlu verilere ait değerleri bir uzay düzlemi yaratarak sınıflama yeteneği gösterir. Lojistik Regresyon Yöntemi bir olayın gerçekleşmesine katkıda bulunan değişkenleri saptar ve bunların modele kattığı olasılık değerini hesaplar. Kernel fonksiyonlar yönteme denek bazında olasılık hesaplama yeteneği kazandırır. Bu olasılık değerlerinin toplamı eşik değer gibi sınıfların sınır değerlerini belirler. Sonuç olarak söz konusu verinin gerçekleşme ihtimali o verinin 'Var' ya da 'Yok' sınıfına atanmasını sağlar. Beklenen sonuç ve mevcut sonucun tutarlılığı veri setinin doğru sınıflama değerini yansıtır. Lojistik Regresyon yöntemi kolay açıklanabilir ve risk hesapladığı için sağlık verilerinin analizinde avantajlı bir yöntemdir. Yapay Sinir Ağları, Destek Vektör Makineleri ve Lojistik Regresyon yöntemlerinin sınıflama başarıları WEKA (Waikato Environment for Knowledge Analysis) paket programı ile karşılaştırıldı.

Anahtar Kelimeler (Keywords): Destek Vektör Makineleri, Kernel Çekirdek Fonksiyonu, Lojistik Regresyon Yöntemi, WEKA, Yapay Sinir Ağları

Danışman (Advisor): Prof.Dr. M.Yusuf ÇELİK Biruni Üniveristesi

2.ABSTRACT

A COMPARISON OF CLASSIFICATION ACHIEVEMENTS OF ARTIFICIAL NEURAL NETWORKS, SUPPORT VECTOR MACHINES AND LOGISTIC REGRESSION METHODS

Artificial Neural Networks, Support Vector Machines and Logistic Regression methods are nonparametric test methods. When the data set have a binary dependent variable they are used for classification. Artificial Neural Network is modelling from the behavior of human brain. The hidden node in the network is the main processing center which uses some statistical functions and threshold values to convert the input variable to the output variable. The cyclic structure of Back Propagation Neural Network provides to process to take the input variable again and again in to the nodes to minimize the errors and provides to network to train itself for best results. Because as The Artificial Neural Network updates the output values continuously, the method takes a small space from the computer memory. In the analysis of continuous measured data stacks it is a successful algorithm for classification. Support Vector Machines differ from the classic linear classification methods by their spiral classification properties. And also SVM differs from classic classification method by creating a space plane for classification tree dimension data values. The Logistic Regression method identifies the variables which contribute to the event and calculates the probability values of those variables for the current model. Kernel functions give the ability to calculate the probabilities of cases to the method. The sum of this probability values determine the boundary of class as a threshold value. As a result, the possibility of occurrence for current data provides the data to be assted to class 'Present' or to class 'Absent'. The consistency of expected and current results of data set shows the correct classification rate. Being easy to explain the statistical results and calculating odds make Logistic Regression the favored method in analysing the healthy datas. The achievement classification of Artificial Neural Network, Support Vector Machines and Logistic Regression Methods have been compared by WEKA (Waikato Environment for Knowledge Analysis) package program.

Advisor: Prof.Dr. M.Yusuf ÇELİK Biruni Üniveritesi

Keywords: Artificial Neural Network, Kernel Function, Logistic Regression, Support Vector Machines, WEKA

3.GİRİŞ

Tıp dünyası verilerinin analizi ve değerlendirilmesinde; hastalıkların mortalite ve morbidite oranları ile risk faktörlerini belirlemek, doğru teşhis koymak, tedavi etkinliklerini ölçmek, tahmin yöntemlerini kullanarak hastalık profilaksisi oluşturmak gibi klinik ve epidemiyolojik alanlarda sıkça istatistik biliminden faydalanmıştır.

Bilgisayar destekli istatistik yöntemleri; karar vermeyi kolaylaştıracak etkin sınıflama algoritmalarını geliştirmeyi amaçlamıştır. Bu algoritmaların her birinin farklı parametreler kullanıyor olması veya farklı veri türlerinde farklı duyarlılığa sahip olması bizi bu araştırmaya yönlendirmiştir.

Veri analizinde; değişkenler ölçüm biçimi yönünden kantitatif ise regresyon ve tahmin analizi, kalitatif değişken ise sınıflama analizleri yapılmalıdır. Yapay Sinir Ağları ve Destek Vektör Makineleri sınıflayarak, Lojistik Regresyon ise regresyon yöntemi kullanarak tahminlemede bulunurlar. (Kuyucu, 2012)

Bilgisayar destekli veri analizinde Destek Vektör Makineleri sınıfları negatif ve pozitif şeklinde ikili ayırım yaparak belirlerken Lojistik Regresyon ve Yapay Sinir Ağları k En Yakın Komşu ve Karar Ağacı sınıflandırma olasılık dağılımlarından yararlanarak sınıflama yapar.

Yapay Sinir Ağları ve Lojistik Regresyon biyotıp alanında en yaygın kullanılan yöntemlerdir. Destek Vektör Makineleri ise daha çok ticaret sektöründe yaygın kullanıma sahiptir.

MEDLINE yayın dizininde; 28,500 Lojistik Regresyon, 8,500 Yapay Sinir Ağları, 1300 K En Yakın Komşu, 1.100 Karar Ağacı ve 100 tane Support Vector Machine çalışması mevcuttur. (Dreiseitl v.d. 2002)

Lojistik Regresyon Analizi Odds oranlarına bakarak en yüksek risk faktörlerini ve sınıf değişkenini saptar. Sınıflandırmayı bir F fonksiyonu ve aynı zamanda bir α vektörü parametresi kullanarak ifade etmeye çalışır. X_i verileri bize toplanabilir bir vektör bileşeni şeklinde sunulur. Böylece x ve α 'yı basit doğrusal bir vektör şeklinde yazmaya olanak sağlar.

Yapay Sinir Ağları ise her değişkenin kat sayılarını hesaplayarak girdi ve çıktı değişkeni oluşturmak koşulu ile sınıflandırma yapar. Yapay Sinir Ağları doğrusal bir

sınıflama yöntemi değildir, bu durum Lojistik Regresyona göre sınıflama yaparken modeli daha esnek kılar. Lojistik Regresyon ve Yapay Sinir Ağlarının ikisi de sınıflama yaparken maksimum (likely hood estimation) olasılık durumunu kullanır. (Bissocat v.d., 2016)

Destek Vektör Makineleri sınıfların vektörel uzunluklarını hesaplayarak, verileri pozitif ve negatif iki kategori şeklinde sınıflara atayan, kernel fonksiyondan yaralanan, doğrusal olmayan, esnek veri setlerini ayrıştırma yöntemidir.

Lojistik Regresyon, Yapay Sinir Ağları ve Destek Vektör Makineleri yöntemlerinde kullanılan fonksiyon, yarı parametrik veya parametrik olmayan model özelliğindedir. Bu nedenle Lojistik Regresyonun katkıları yazılırken kestirim noktaları ve güven aralıkları, Yapay Sinir Ağlarında sinir düğümlerinin ağırlık parametreleri, Destek Vektör Makinelerinde her değişkene ait pozitif ve negatif değerler belirtilmelidir.

Yapay Sinir Ağlarında ve Destek Vektör Makinelerinde, Lojistik Regresyonda kullanılan belirleyicilik katsayıları (significiant parameter) kullanılamaz. Burada otomatik alaka belirleme (relevance determination) ya da duyarlı analiz (sensitive analizis) kullanılır. (Elmas 2003)

Modeller geliştirilirken sınıflandırmada iki ana fikir öne çıkmaktadır. Bunlardan biri diskriminant (ayrımcılık) diğeri ise kalibrasyon (kalibre etmek) yaklaşımıdır.

Diskriminant yaklaşımı verilerin en iyi nasıl iki sınıfa ayrışacağını ve $F(x, \alpha)$ olasılık tahmininin gerçek olasılık tahmini $P(x, y)$ 'ye ne kadar yakınsadığını doğrulamaya çalışır. İşlem yapılırken genellikle gerçek verinin küçük bir parçası alınarak test veya doğrulama amacı için kullanılır. Diskriminasyonda genel ölçümler; duyarlılık (sensitivity), seçicilik (spesifity), doğruluk (accuracy) ve roc eğrisi (ROC CURVE) ile desteklenir ve tek bir istatistiki modelin açıklayıcılığından daha üstün kabul görür.

Lojistik Regresyonda; parametrelerin seçimi, değişkenlerin seçimi, belirleyicilik katsayıları araştırmada modeli anlaşılır kılarken Yapay Sinir Ağlarında ve Destek Vektör Makinelerinde eşik değerlerin yazılması ile bu mümkündür.

Yapay Sinir Ağlarında aşırı uyum problemi; node sayısı düşürülerek, ağırlıklar çürütülerek, erken durdurma ile mümkündür. Bayesian yaklaşım

kullanılarak araştırma sonuçları zenginleştirilebilir. Geri beslemeli sinir ağlarından (Backpropagation) kaçınılmalıdır. Gradient ya da Newton metodlarının modeli yavaşlatacağı unutulmamalıdır. (Şeker 2013)

Destek Vektör Makinelerinde; '0' değeri 'false' olarak atanmalı, veri kümesi büyükse ağırlıkları limitleme (shrinkage) uygulanmalı, hata değeri (lambda) belirlenmelidir. (Akkuş v.d., 2008)

Araştırmada sonuçlarının kalibrasyonu sağlanmalı ve discriminant gücü açıklanmalıdır. Bu iki koşulun yerine getirilmesi uygulanan yöntemlerin bilimsel geçerliliğini belirleyen ve araştırmayı güçlü kılan gerekliliklerdir.

'Black box model' olarak ifade edilen Yapay Sinir Ağları ve Destek Vektör Makineleri yorumdan ziyade sınıflamanın üstün olduğu yerlerde daha çok tercih edilmelidir. Bunların dışındaki modeller ise 'white box model' diye tanımlanır ve çıkan sonuçlar bu modelde yorumlanabilir. Lojistik Regresyon bunlardan biridir. (Dreisetl 2002)

YSA esnek ve doğrusal olmayan modellerde, gürültülü ve tamamlanmamış veri setlerinde hata toleransı ve genelleme yeteneği nedeniyle 'robust test' gibi kabul edilir. (Eftekhari v.d., 2005)

DVM esnek ve doğrusal olmayan ancak kayıp veri içermeyen modellerde ve sadece sınıflama gerekli durumlarda tercih edilmelidir.

Lojistik Regresyon ikili (dichotomus) yanıt değişkeninin risk analizinin yapılmasının istendiği, küçük ve doğrusal yapıdaki veri seti içeren modellerde tercih edilmelidir.

Çalışılacak veri türünün özellikleri yöntemi belirlemede ve en doğru sınıflama oranına ulaşma noktasında en önemli faktördür.

4.GENEL BİLGİLER

4.1.Yapay Sinir Ağları

Akıllı; genetik yolla aldığımız iç güdüsel davranışlarımızın doyurulmuşluk oranı ve çevredeki dışsal uyarıların zenginliği ile yaşam koşullarına bağlı olarak geliştirebildiğimiz yeteneğimizdir. Aynı şekilde, günümüz yazılım teknolojisi imkanları; aklın geliştirilebilir özelliğinden esinlenilerek makinelerin öğrenebileceğini varsaymış ve yapay zeka algoritmaları üretmeyi amaçlamıştır.

Klasik algoritma yönteminde matrisler oluşturulur ve her hücreye piksel adı verilir. Pikseller ekranda bir noktaya karşılık gelir ve ikili sayı sistemine dönüştürüldüklerinde alacağı değerler '0' yani boş değer veya '1' yani dolu değer olarak kaydedilir. Her piksel bir noktaya denk gelir ve ekranda görüntünün oluşmasını sağlar. Tüm veriler ikili sayı sistemine (Binari) dönüştürülerek 'ASCII kodlama sistemi' oluşturulur. Bu dönüşüm tüm analog verilerin dijital veriye dönüşmesini veya dijital verinin analoga dönüşmesini sağlayan, bilgisayar sisteminin hafıza yeteneği yani bilginin depolanmasına olanak veren yazılım mantığının en kaba açıklamasıdır. (Bishop, 2005). Yapay zeka bu birebir karşılaştırmanın bir adım ötesinde; daha önce hiç karşılaşılmamış veya hatları düzgün olmayan bir karakterin (girdi örüntüsü) matristeki dağılık yayılımını kendi kendine hata küçültme yöntemi yani öğrenme yeteneği ile derleyerek karakteri tanıma ya da eşleştirme yeteneğine sahiptir. İki yazılımın bu büyük farklılığı klasik algoritmada verinin tek tek ve düzenli bir şekilde akış gerekliliği ve statik yöntemle mevcut kodlarla (ASCII kodları) karşılaştırma yapma zorunluluğuna karşılık yapay zekada veriler paralel olarak algılanır ve girişte oluşabilecek gürültüye (örneğin bozuk bir resim veya veri kümesindeki karışık değerler gibi..) rağmen ağıllı dinamik ve değişebilir yeteneği sayesinde veriler çıkışta tanımlanıp bir sınıfa atfedilebilir.

Yapay sinir ağlarını, diğer bilgisayar yazılımlarından ayıran bir başka özelliği ise genel bilgisayar yazılımlarında oluşturulan veriler, bilgisayarın depolama birimi olan RAM yada ROM'larda biriktirilirken yapay sinir ağları bilgiyi kendi ağ yapısı içinde (algoritmasında) tutmak zorundadır çünkü bilgi sürekli güncellenmekte ve değerler her eğitim işleminde yeniden ağırlıklandırılmaktadır.

4.1.1.Tarihi gelişim

Yapay Sinir Ağları 1943 yılında Warren McCulloch ve Walter Pitts tarafından, basit bir sinir ağı modeli oluşturma ile başlamıştır. 1949 yılında ise Donald Hebb, bilgisayarların öğrenmesi ile ilgili bir kavram ortaya atmıştır. 1950 yılında IBM laboratuvarında ilk sinir ağı simülasyon çalışması başarısızlığa uğramıştır ancak 1957 yılında Frank Rosentblatt, 'Bir sineği harekete geçiren şeyin sineğin beyni değil de gözüyle algıladıklarıdır' mantığından yola çıkarak 'Kavrama Düğümü' adı verilen ilk sinir ağının oluşmasını sağlamış, -tek katmanlı, eğitilebilen, tek çıkışı olan- perseptron makinasını çalıştırmayı başarmıştır. 1959 da Bernard Widrow ve Marcian Hoff ADALİNE(Adaptive linear yöntem) ağını geliştirerek telefonlardaki cızırtıların yok olmasını sağlamış ve böylece yapay sinir ağları gerçek yaşamın içinden bir problemi çözmeyi başarmıştır. Marvin Minsky ve Seymour Paper tarafından 1969 yılında tek katmanlı makinelerin 'XOR' problemlerini çözemediği gözlemlenmiş ve çok katmanlı sinir ağları oluşturulmaya çalışılmış, ancak sinir ağlarının ağırlıklarının nasıl düzenleyebileceği bulunamamıştır. 1970'de Fukussima, NEOCOGNITRON modelini geliştirmiş ve bu model ile şekil ve örüntü tanılamayı sağlamıştır. (Elmas, 2003)

Yapay sinir ağlarının kendi kendine öğrenme yeteneğinin olması nedeniyle toplumda bilgisayarın insanlığı ele geçireceği kanısının veya korkusunun ortaya çıkması bu alanda çalışmaların yavaşlamasına neden olmuştur.

John Josep Hopfield, 1982 yılında yapay zeka alanındaki çalışmalarını beynin bir modeli olarak değil de bir matematik algoritması şeklinde ve bir makine düzeneğinde sunmuş ve çalışmaların önünü açmıştır. 1984 yılında Teuvo Kohonen danışmansız öğrenme ağını ve 1986'da Rumelhard ile McClelland karmaşık ve çok katmanlı ağlarda geri yayılım algoritmasını üretmişlerdir. Bu alanda ilk uluslararası konferans 2000 kişiye yakın bir katılımı ile, 1987 yılında Elektrik Elektronik Mühendisliği Enstitüsü (Institute of Electrical Electronik Engineering) tarafından Amerika'da yapılmış ve yapay sinir ağları yaşamın içine girmiştir. (Elmas, 2003)

1988 yılında Broomhead ve Lowe 'Radyal Tabanlı Fonksiyonları' geliştirmiş ve filtreleme konusunda başarılı sonuçlar elde edilmiştir. Spect daha gelişmiş olan Probabilistik Ağları (PNN) ve Genel Regresyon Ağlarını (GRNN) geliştirmiştir. (Gönül vd.,2015)

Yapay sinir ağlarının avantajları:

- Yapay Sinir ağları paralellik yapısına sahiptir. Bu özelliği ile herhangi bir nöron işlevini kaybetse dahi diğer nöronlar çalışmaya devam eder.
- Doğrusal olmayan ağ fonksiyonları uygulanabilir.
- Yapay sinir ağları; üzerinde çalıştığı örneklerden yola çıkarak genelleme yapar ve daha önce karşılaşmadığı yeni verileri sınıflayabilir.
- Bir uzman gibi çalışabilir, kendisine yüklenen bilgileri ağın tamamında saklama özelliği ile; ilişkilendirme, tahmin, sınıflama, örüntü tamamlama, algılama, modelleme yapabilir.
- Eksik bilgilerle çalışabilir, geleneksel optimizasyon teknolojileri ile çözülemeyen problemleri genetik algoritmalar yardımı ile cevapları bir araya getirerek yeni çözümler oluşturur.
- Bulanık mantık yöntemi ile kesin rakamlarla ifade edilemeyen problemlerde belirsiz bilgileri işleyebilme ve hata toleransına sahiptir.
- Değişik zeka tekniklerini bir araya getirebilen ve bağımsız çalışabilen esnek programlara sahiptirler. (Jodas v.d. 2013; Yakut v.d. 2014; Tunç 2012)

Yapay sinir ağlarının dezavantajları:

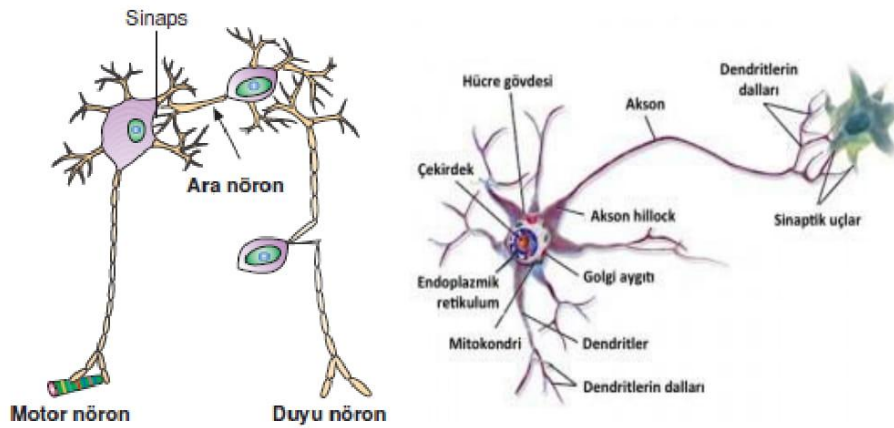
- Sadece numerik bilgilerle çalışabilir.
- Ağın ezberleme özelliği ya da çoklu doymuşluk nedeniyle verileri ayrıştırma yeteneğinden uzaklaşabilir ve kaba sonuçlar verebilirler.
- Ağın eğitilmesi çok uzun zaman alabilir. (Jodas 2013; Yakut v.d. 2014; Tunç 2012)

YSA endüstride uçuş simülasyonu, yedek pilot uygulaması, rota belirleme, navigasyon takip sistemleri, stok hesaplama, boya malzemesinin renk karışımından tutun da bir makinenin hangi teknik parçalarla daha iyi performans sağlayacağına kadar bir çok alanda kullanılmıştır. Ayrıca, bankacılıkta kazancı yüksek yatırımları sınıflama, müşteri profili belirleme ve banka kredisi verme gibi konularda; Askeri alanda bir silahın girilen koordinatlara göre hedefi yakalama düzeyi, kimlik doğrulama, suçlu teşhisi, insan profili oluşturma ve suça eğilimli karakter saptama, konuşma-ses tanımlama, hava tahmini, trafik yoğunluğu gibi konularda; Eğitimde, başarı kriterlerini belirlemede; Tıpta hastalık tanımlama, ölümcül travma sınıflaması, ameliyat sonrası beyin hasarları sınıflaması, ve transplantasyon başarısı gibi daha

binlerce hastalığın teşhisinde etkenlerini sınıflama ve tedavi performansını değerlendirme çalışmalarında, EKG, EEG, MR ve radyolojik imaj yöntemlerinin doğru çalışma prensibi geliştirme ve doğru teşhis koyma çalışmalarında da kullanılmıştır.

4.1.2. İnsan sinir hücresi ve YSA benzerliği

Yapay sinir ağları mimarisi gerçek insan nöron hücresinden esinlenerek oluşturulmuştur. İnsan vücudunda ortalama 100 milyar nöron olduğu ve her bir nöronun 25 bin ve 50 bin civarında diğer nöronlarla temasta olduğu varsayılmaktadır. Bir nöron hücresinde esas zihinsel aktivitenin oluşmasını sağlayan bir dentrit ve dentrite bağlı tek veya çoklu akson yapıları mevcuttur. Aksonlar iletiyi aktarma işlemini yapar. Aksonlar uyarılmış organdan duyuları alarak afferent sinir yolu ile beyne iletir, beyinde oluşan cevabı efferent sinir yolu ile efektör (uyarılmış) organa iletirler. Aksonlar arasında somut bir temas yoktur. Her nöron bir diğer nöronla sinaptik aralıkta karşılaşır ve nörotransmitter maddeler reseptörler aracılığı ile bir nörondan diğerine aktarılırlar. Adrenalin (hızlandırıcı) ve asetilkolin (yavaşlatıcı) gibi nörotransmitterler ilgili reseptörlere bağlanırlar. Reseptörler aktarım işlemini gerçekleştiren taşıyıcılardır. Alfa ve beta reseptörler adrenalini taşıırken muskorinik ve nikotinik reseptörler ise asetilkolini taşırlar. Sinir uçlarından nörotransmitterlerin salınımı ve aktarımı otonom aktivitelerin meydana gelmesini sağlar. Otonom aktiviteler vücudumuzdaki 'iç denge' dediğimiz tüm istemsiz faaliyetlerin düzenlenmesini sağlar.



Şekil 1: Nöron Yapısı ve Sinaptik Aralık

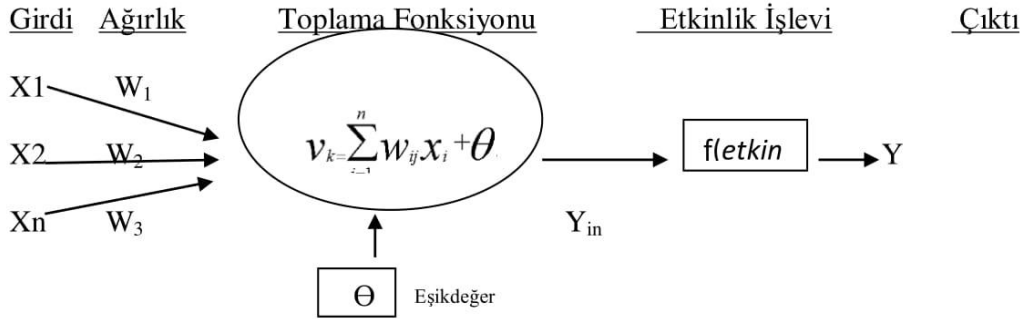
<http://www.dicle.edu.tr>

Duyu nöronu afferent siniri yani uyarıyı alan ve beyne ulaştırın, motor nöron ise efferent siniri yani beyinde oluşan cevabı uyarılmış organa taşıyan yolu temsil etmektedir. (Solomon,2002)

Bir yapay sinir hücresi modeli 5 bileşenden oluşur. Bunlar; girdiler, ağırlıklar, transfer fonksiyonu, aktivasyon fonksiyonu ve çıktılardır. (Bayata ve Hattatoğlu 2010)

Temel bir yapay sinir ağında; veri girişlerinin gerçekleştiği **giriş katmanı**, giriş katmanındaki veriyi taşıyan **ağırlıklar**, toplama işlevi ile eşik değerin uygulandığı ve bulunan sonuçların işlenip ölçeklendirme ve sınıflamanın yapıldığı **etkinlik işlevi katmanı (transfer fonksiyonu ve aktivasyon fonksiyonu)** ve **çıkış** katmanı bulunmaktadır. İnsan anatomik yapısı ile eşleştirmek gerekirse; giriş katmanı dentritleri, ağırlıklar aksonları, işlevsel birim sinaptik aralığı ve çıkış birimi elektriksel uyarıyı temsil eder.

4.1.3. Basit yapay sinir ağı topografisi



Şekil 2: Tek Katmanlı İleri Beslenmeli Sinir Ağı

(Elmas 2003)

X_1, X_2, \dots, X_n : Veri girişlerini simgeler ve dentritlerden esinlenmiştir

W_1, W_2, \dots, W_n : Ağırlıkları temsil eder ve aksonlardan esinlenmiştir.

V_k = Toplama Fonksiyonu:

$$V_k = \sum_{i=1}^n W_{ki} * x_i + \theta_j$$

W_{ki} =Her girdi değerine ait ağırlıkları (K=Eğitim tekrar sayıdır.)

x_i =Girdi değeri

Θ :Eşik Değeri, uyarının oluşma koşulu ve sinaptik aralıktaki nörotransmitter maddenin eşik değeri gibi algılana bilir.

$f(\text{Etkinlik})$ = Aktivasyon Fonksiyonu, Doğrusal eşik Değerin Aktarım İşlevi

Y_{in} =Toplam sinyal,

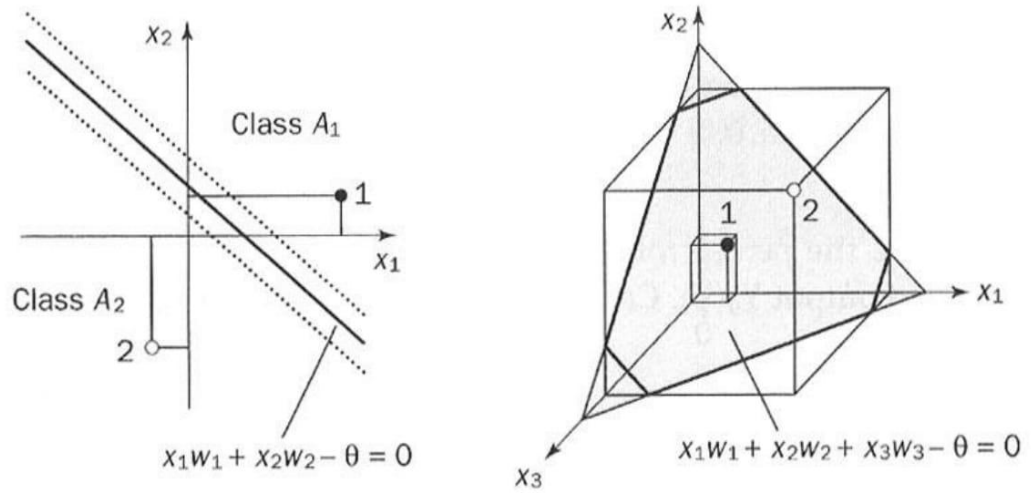
Y =Çıkış, sinaptik aralıkta oluşun sinirsel kararı temsil eder.

(Elmas,2003)

4.1.4. YSA'da sınıflamanın yapılması

Sınıflandırmada esas amaç verinin hangi sınıfa ait olduğunun en az emek ve en güvenilir sonuçlarla belirlenmesidir. C_1 ve C_0 şeklinde iki sınıfımızın olduğunu varsayarak; çıktı değeri $X_i = 1$ ise C_1 sınıfına dahil olmakla birlikte $X_i = 0$ ların C_0 sınıfına ait olduğu düşünülmüştür. Sistem; X_1, X_2, \dots, X_i diye ifade edilen ve piksel yoğunluklarını temsil eden girdi birimi değerlerini, y 'ler ise sınıf etiketine dayalı çıktı değerlerini oluşturur. Mevcut sınıflama deseninde yapay sinir ağları ağırlık vektörünü hesaplar, verilere ait histogramdan yararlanır ve eşik değerleri kullanarak toplanabilir parametre değerlerinden $y_k(x;w)$ fonksiyonuna ulaşır. Yapay sinir ağlarının önemli özelliklerinden biri non-linear olan verileri sınıflandırabilmesidir ve esas amaç en esnek fonksiyonu kullanarak (genelde Sigmoid $f(x) = \frac{1}{1 + e^{-x}}$) girdileri çıkışa yönlendirmektir. (Yavuz ve Deveci 2013)

Yapay sinir ağları veriyi kullanmadan önce ön işleme tabi tutar. Örneğin x 'e ait değeri iki boyutlu değil de hacim olarak karşılaştıracak ve sınıflayacak ise önce x 'in boyutlarını hacim cinsinden hesaplar.



Şekil 3: Üç Boyutlu verinin YSA sınıflanması

<https://en.wikipedia.org>

Önişlemin bir diğer katkısı; verileri farklı aralıklarla bölerek aynı anda birçok verinin işleme girmesi ve sistemin hızını yavaşlatmasını engellemektir. Verileri farklı aralıklara bölmek aynı zamanda veriyi hızla ait olduğu sınıfa göndermemizi de sağlar.

4.1.5. YSA öğrenme yöntemleri

Yapay sinir ağlarındaki verinin çıkışa yönlendirilmesi yani ilerleme sonucunda, 'öğrenme' yada 'eğitim' dediğimiz çıktı süreci gerçekleşmiş olur.

Yapay sinir ağlarında, iki sinirin birbirini ortaklaşa uyarması sonucunda, sinirler arasındaki bağlantı kuvveti yani ağırlıklar bir katman sinirinden diğer katman sinirlerine aktarılır ve bu değer aktarıldığı her yeni sinir katmanında o sinirin etkinlik işlemine tabi olur ve sinirin etkinlik değeri ile çarpılarak ağın çıkışına doğru ilerler. Çıkış değerleri yeni ağırlıkların belirlenmesi için ağın içinde gerekli görülen bir yere aktarılarak (ilgili katmanın ağırlığı veya sinir katmanı) ağın kendi içinde hafızalanmasını sağlar. En iyi çıkış değerine ulaşabilmek için ağın ağırlık değerlerinin bu şekilde güncellemesi faaliyetlerine 'ağın öğrenmesi' denir. En iyi ağırlık değeri hata değerinin sıfırlanması demektir ve tüm döngü sürecinde hataların toplanarak ağırlıkların yenilenmesine 'küme kipi' öğrenme denir. (Jodas v.d. 2013)

ÖĞRENME:

X_i sinir girişlerinden ağa girer. Ve

$$y_i = f(w, x) = f(w^T x) = f\left(\sum_{j=1}^n w_j x_j\right) \quad (\text{Ocakoglu 2006})$$

Burada

$w_i = (w_1, \dots, w_n)^T \in R^n$ ağırlık vektörüdür.

$f(w^T x)$ etkinlik ve ayın zamanda aktarım işlevidir. Ağ değerleri giriş vektörleri ile tanımlanır.

$$a\check{g} = w^T x = w_1 x_1 + w_2 x_2 + \dots + w_n x_n$$

$$y_i = a\check{g}$$

Ağın çıkış değeri ' θ ' yani eşik düzeyle karşılaştırılarak ilerleme sağlanır.

Aşağıda $\theta = 1$ olarak alınmıştır.

$$y_i = f(a\check{g}) = \begin{cases} 1 & \text{eğer } w^T x \\ 0 & \text{eğer farklı} \end{cases}$$

"Farklı" ifadesinden kastedilen ağın tasarlanışına göre farklılık gösterir. Yani ağın algoritmasında başka bir sinire aktarımı veya başka bir aktivasyon fonksiyonunun tanımlanması varsa ilerlenecek adım bu yönde olacaktır.

Ağa girişte tanımlanması istenilen diğer değerler ise

$$w_1 = \text{ilk ağ ağırlığı}$$

ϵ = Giriş hata değeri (genelde ilk adımda 0 tanımlanır.)

d = düzeltme katsayısı

Ağın $f(w^T x) \geq \theta$ değerine ulaştığını varsayarsak, ağ çıkışa yönelir. İstenilen çıkış değeri v_i ve gerçek çıkış değeri z_i ile ifade edilsin.

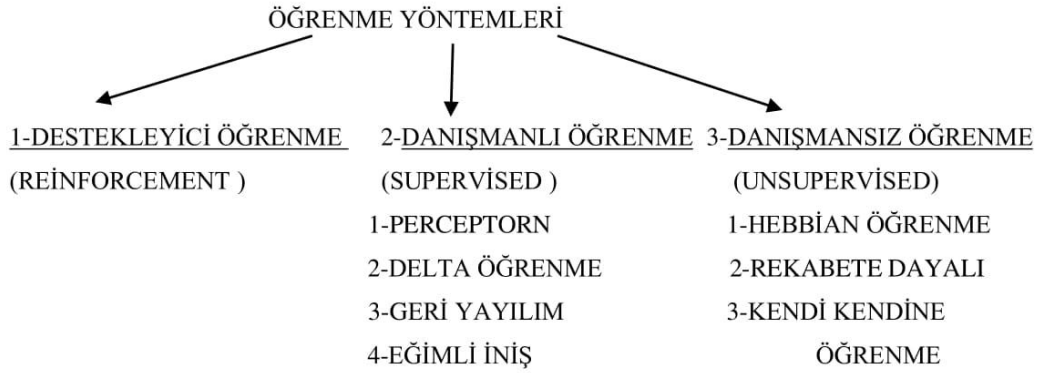
Hata $E_i = v_i - z_i$ formüle edilir.

Düzeltilme faktörü ise

$\Delta w_i = (E_i + \epsilon)d$ şeklindedir.

Eğer $z_i = 0$ ve $v_i = 1$ ise $w_{ij} = w_{ij} + \Delta w_i$

Eğer $z_i = 1$ ve $v_i = 0$ ise $w_{ij} = w_{ij} - \Delta w_i$ şeklinde yeni döngünün (İterasyon=t) ağırlık değerleri güncellenir. İlerleme yada öğrenme gerçekleşmiş olur. (Öztemel 2012)



Şekil 4: Öğrenme Yöntemleri Şeması

(Elmas 2003)

4.1.5.1. Destekleyici öğrenme

Destekleyici öğrenme yönteminin amacı; gerçek bir çıkış değeri vermek değil de ağın çıkış değerlerini ne kadar doğru verdiğine dair ağın performans değerini iyi veya kötü olarak dereceleme gayretindedir. Denetim hareketlerinin başarısı ile ilgilidir.

Danışmanlı öğrenme yönteminde ağa hem giriş verileri hem de çıkması istenilen sonuçlar yüklenir. Oysa bazı durumlarda ağın nasıl bir sonuca ulaşması gerektiği bilgisi ağa tanıtılamaz çünkü bu konuda elde edilmiş araştırma sonuçları mevcut olmayabilir. Ancak ağ sonucunun doğruluğu yada yanlışlığı sorgulanabilir.

Bu şekildeki öğrenme yöntemine destekleyici öğrenme denir ve kohonenin ağlarındaki LVQ(linear vektör quantitation) (doğrusal vektör parçalama yöntemi) bunun en yaygın kullanım şeklidir. (Öztemel,2012 ; Elmas 2003; Ocakoğlu 2006)

4.1.5.2.Danışmanlı öğrenme

Danışmanlı öğrenme yöntemi yapay sinir ağının kullanılmadan önce eğitilmesi temeline dayanır. Yani ağa giriş ve çıkış bilgi çifti ve istenilen değer sunulur. Ağın hata değeri istenilen değerden daha düşük bir değere ulaşıncaya kadar ağırlıklar sabitlenir ve öğrenme tamamlanmış olur. Öğrenmenin tamamlanması ağın kullanıma hazır olması demektir. (Öztemel,2012 ; Elmas 2003; Ocakoğlu 2006)

4.1.5.3.Danışmansız öğrenme

Bu öğrenme yönteminde girişte sadece ham veriler verilir ve ağ online öğrenme dediğimiz iterasyonlar sırasında bir taraftan yeni duruma adapte olurken diğer taraftan bilgileri unutulur. F1 gösterimi girdiyi ve F2 kategori yada çıktı katmanı şeklinde iki katmanı vardır., F2 katmanında en yüksek proses alan kategori kazanan olur ve 1 değerini alırken diğer proseslere 0 değeri yüklenir. Ağ bu değişken temelinde verileri tek tek ayrıştırarak değişkenin hangi farklı özelliklerden yararlandığını tanımlar. (Öztemel,2012 ; Elmas 2003; Ocakoğlu 2006)

Örneğin; yetenek sınavında kazanan ve kazanmayan şeklinde iki sınıfın var olduğunu düşünelim ve 100 kişilik öğrenci grubunun verilerinin ağa 1000 kez tekrar ettirildiğini düşünelim. Sonuç olarak öğrencilerin hangi kriterlere istinaden kazanmış olduğunu veya kazanmayanların özelliklerini ağ tanımlayarak sınıflama yapabilir ve daha sonraki araştırmalarda ağa girilen rastgele bir kişinin kazanma olasılığını kolayca hesaplayabilir. Bu şekilde ağın kendi kendine öğrenmesine ‘danışmansız öğrenme’ adı verilmektedir.

Bu öğrenme yönteminde ART1 bilinen ilk ağ olup sadece binary değerlerinden oluşan girdi vektörlerini tanımlarken ART2, ART3, FuzzyART, ARTMAP gibi modeller geliştirilmiştir. Grup teknolojisinde makine hücrelerinin belirlenmesi, endüstriyel problemlerde özellikle etiket ayrıştırarak sınıflama problemlerinin çözümünde geleneksel yöntemlere göre çok daha başarılı sonuçlar vermektedir. (Öztemel 2012)

4.1.6. YSA Aktivasyon fonksiyonları (Etkinlik işlevi)

Aktivasyon fonksiyonu, toplama işlevinin sonucunda gelen değerleri işleyerek çıkışa yönlendirir. Amaç toplama işlevinin çıkışının değiştirilmesine izin vermektir. Sinir aktivasyon fonksiyonunda belirlenen eşik değere ulaşmadığı takdirde çıkış vermez.

Bu özellik yapay sinir ağlarının doğrusal olmama özelliğinden kaynaklanmaktadır. Aktivasyon fonksiyonu belirlenirken dikkat edilmesi gereken en önemli özellik fonksiyonun türevinin kolay hesaplanır olmasıdır. Çünkü geri beslemeli ağlarda hata küçültmek için fonksiyonun türevi alınacaktır. Günümüzde en yaygın kullanılan fonksiyon; Sigmoid fonksiyondur. (Çayıroğlu 2017)

Aktivasyon fonksiyonları aşağıda verilmiştir.

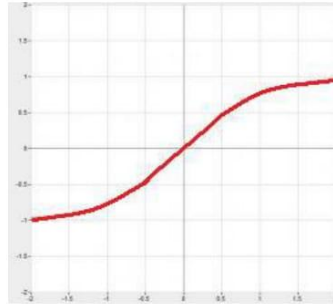
1-Tangent-sigmoid: tansig

2-Logaritmik-sigmoid:Logsig

3-Purelin (Altun v.d. 2012)

1- Tansig: Çift kutuplu fonksiyonun değişim aralığı $[1 \ -1]$ dir. Tanjant Hiperbolik Fonksiyon olarak da ifade edilir. Lineer bir fonksiyon değildir.

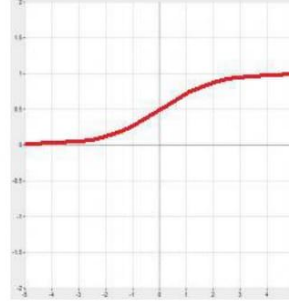
$$y = \frac{1 - e^{-2v}}{1 + e^{2v}} = \tanh(\beta v)$$



Şekil 5. Aktivasyon Fonksiyonları

(The Institute Of Electrical And Electronic Engineers)

2- Logsig: Tek kutuplu fonksiyonun deęişim aralıęı $[0 \ 1]$ dır. Sigmoid fonksiyon olarak da ifade edilmektedir.



$$y = f(v) = \frac{1}{1 + e^{-v}} = \frac{1}{2} \left[\tanh\left(\frac{v}{2}\right) + 1 \right]$$

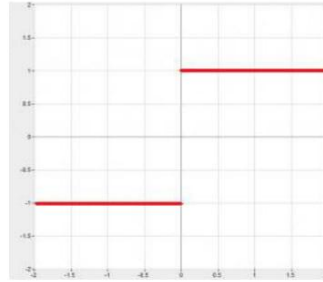
Şekil 5. Aktivasyon Fonksiyonları

(The Institute Of Electrical And Electronic Engineers)

3-Purelin: Fonksiyonun deęişim aralıęı $[-1 \ 1]$ dır. Lineer bir fonksiyondur. Girdi deęeri ilgili aęırlıkla çarpılır ve eşik deęerle toplanarak sinirin aktivasyonu saęlanır.

$$v_k = \sum_{i=1}^n w_{ki} * x_i + \theta_j$$

Perseptron Basit Algılayıcı Model: Çift kutuplu basamak fonksiyonuna da burada yer vermek gerekir.



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

Şekil 5. Aktivasyon Fonksiyonları

(The Institute Of Electrical And Electronic Engineers)

Aktivasyon deęeri ' θ '= $[v \geq 0 \ v < 0]$ aştığı zaman nöron aktif olur. Sonuçlarına göre sınıflama yapılır. Fonksiyonun deęişim aralıęı $[-1 \ 1]$ dir. (Altun v.d. 2012)

4.1.7. YSA kullanılan kavramaları

4.1.7.1.Momentum (Öğrenme kat sayısı)

YSA'da öğrenme ilk adımlarda büyük olur ancak ağın sonuna geldikçe öğrenme düzeyi küçülmeye başlar. YSA son döngülerinde öğrenmeye bir sabit sayının yön vermesi ağın hata düzeyinin sıfıra yaklaşmasında olumlu katkı sağlar ve ağın eğitilmesine hız kazandırır. Uygulanacak sabit sayıya 'Momentum' denir. Ağın hatasının sıfıra yaklaşması ile hassas sonuçlara ulaşılması sağlanır. Bir önceki ağırlık değişim katsayısı ve yönü yardımı ile bir sonraki ağırlık değişim miktar ve yönü momentum uygulanarak kontrol altında tutulabilir.

$$\Delta w_{ij}(t+1) = \eta \delta_j y_i + \alpha \Delta w_{ij}(t)$$

α =Momentum (Elmas 2003)

(Fonksiyonun detaylı anlatımı '2.1.8.1. İleri Beslemeli Ve Çok Katmanlı Sinir Ağ Modelleri' bölümünde mevcuttur.)

4.1.7.2.Öğrenmede iterasyon

Öğrenme Esnasında Verilerin ağa tanıtılması gerekir. Örneğin Türkiye'deki yağış karakterini analiz etmek isteyen bir kişinin 30 günlük yağış kaydının ağa yüklenmesi, ağın tahminlerinin kaba ve güvenilir olmasını engeller. Ancak ağa 10 yıllık yağışın tanıtılması da aynı şekilde ağın cevabının kaba ve yuvarlamalı olmasına neden olur. O nedenle iyi tahmin veya sınıflama için ağın yeter düzeyde eğitilmesi gerekir. Ağın döngüsel bir şekilde tekrarlı eğitilmesine İTERASYON denir. Yapay sinir ağlarında veri karakterini analiz ederken iterasyon oluşumu deneme yoluyla saptanabilir. (Yılmaz 2009)

4.1.7.3. Sorgulama

Sorgulama; YSA' da probleme yanıtın verilmesi yani sonuç değerlerine ulaşılmasıdır. YSA sorgulamayı genelleme yada olasılık şeklinde hesaplar.

YSA daha önce görmüş olduğu örnekleri ezberler ve ağa benzer özellikte veri girildiği zaman eksik alanları daha önce öğrendiği örüntülere benzeterek tamamlamaya çalışır. Buna hata toleransı 'Fault tolerant' denir. Örneğin yüz tanıma

programında yüzün bir kısmının tanımlanamaması durumunda ağ daha önceki örneklerden yola çıkarak yüzü bir bütün şeklinde tanımlamaya çalışır. Bu tekniklerden bazıları; Hamming Distance (HD), Eucladian Distance(ED) , Partitioned Generalized Eucladian Distance şeklindeki uzaklık ölçüm teknikleridir. Sınıflamada ise kendi aktiflik fonksiyonu haricindeki bir HD veya ED gibi fonksiyon kullanmadan, sadece benzerliklerden yararlanarak ayrıştırma yaptığı için ağ çok daha hızlı sonuçlar üretir. Optimizasyon problemlerinde ise her iki yöntem birlikte kullanılarak 'hibrit' uygulama yapılarak sorgu tamamlanır. Ağın aktivasyon seçimi uygulayıcının tercihinine bağlıdır. (Yalçın, 2017)

4.1.8. YSA temel model yapısı

4.1.8.1. İleri beslemeli ve çok katmanlı sinir ağ modelleri

Çok katmanlı ileri beslemeli yapay sinir ağları XOR dediğimiz doğrusal olmayan problemleri çözmek için geliştirilmiş YSA modelleridir. En az girdi, ara katman ve çıkış olmak üzere 3 katmandan oluşmaktadırlar. Ağın eğitilmesi ' DELTA KURALI' ile sağlanmaktadır. Delta kuralı ağ hatasının karesini minimize etmek için bağlantı boyutlarını değiştirir. Hata bir önceki katmana geri çoğalır. Her bir zaman dilimi için hata çoğaltma işlemi ilk katmana ulaşıncaya kadar devam eder. Hata geriye doğru düzenleniyor olsa da bu işlem sadece hatayı en küçükleyecek ağırlık kümesine ulaşmak için değer atama işlemidir. Delta ağ yapısında gizli katman ağırlığını düzenleme şansı yoktur. Oysa geri yayılım algoritmasında ilk katmana ulaşıncaya kadar tüm gizli katmanlarda ağırlığın nasıl ayarlanacağı tanımlanır. J katmanındaki sinir, x_i girdi değerlerine ait w_{ij} ağırlıklarını alır, ($i=1,2,3,...n$ ve $j=1,2,..$) hesaplamaları yaptıktan sonra bir sonraki katmanın tüm sinirleri ile bağlantı kurar, ilerleme 'y' çıktı değeri olarak sonuçlanır. İleri beslemeli ağlardaki en önemli kural, x_i değerlerinin katmandaki tüm sinirlerle bağlantılı olması kuralıdır. İleri beslemeli ağlar $g : R^n \rightarrow R^m$ ($g(x) = z$) yapılarını gerçeklememizi sağlar.

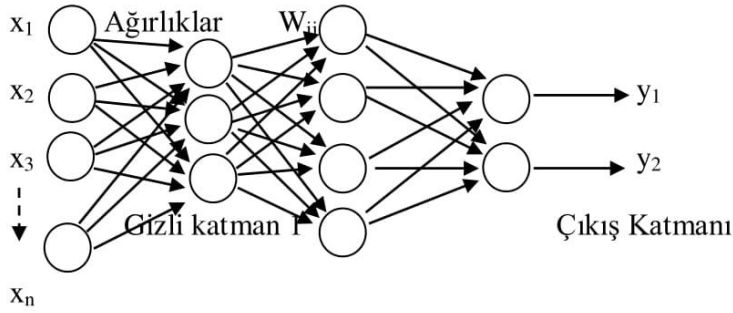
(Eftekhari v.d. 2005)

İleri beslemeli ağlarda,

$$\Delta w_{ij} = \eta \delta_i x_{ij}$$

(η =Momentum(Öğrenme kat sayısı δ_i =i değerlerine ait hata oranı (düzeltme oranı) Δw_{ij} = bir sonraki katmanın ağırlıklarına eklenecek, j katmanındaki, i değerlerine ait ağırlık düzeltilme oranı)

Yukarıdaki işlem yardımı ile Eğimli İniş Öğrenme Yöntemi uygulanır. Bu yöntemde kastedilen hataların kareleri toplamına türev uygulanarak hata düzeyini azaltmaktır, böylece beklenen ve çıkan sonuç değerleri arasındaki fark en aza indirilmiş olur.



Giriş katmanı Gizli Katman 2

Şekil 6: İleri Beslemeli Çok Katmanlı Yapay Sınır Ağı

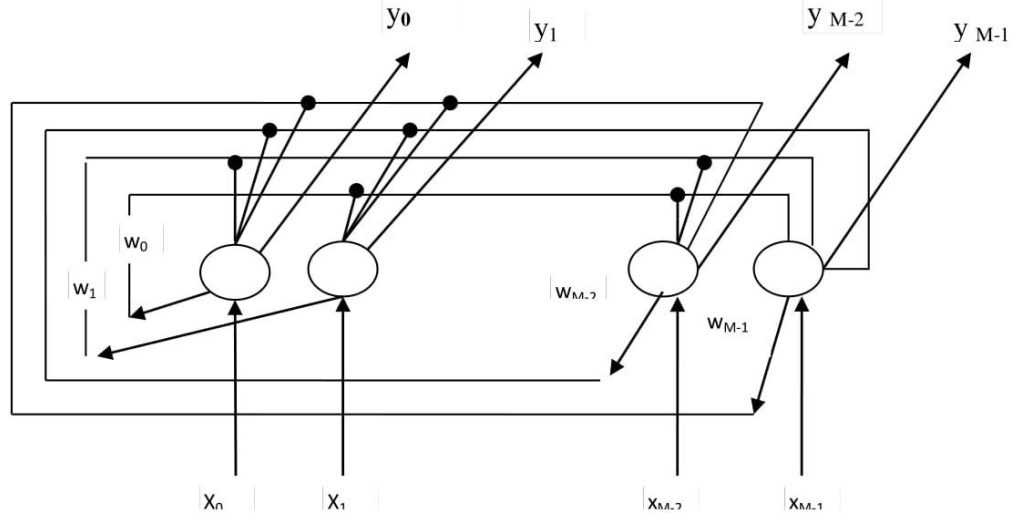
(Elmas 2003)

Bu yöntemin dezavantajı çıkış katmanındaki hataların hangi gizli katmanda ne oranda meydana gelebileceğini saptayamamaktır. Geri yayılım yöntemi bu problemi çözmeyi başarmıştır.

4.1.8.2. Geri beslemeli ağlar

Rumelhard ve arkadaşları geri yayılım modelini (Back propagation) geliştirerek 'Delta Öğrenme Kuralı 'nı geliştirmiş ve problemlerde çözüm aracı olarak kullanmışlardır. Temel amaçları yine ağıın hatasını en aza indirmektedir. Çok katmanlı ağlar danışmanlı öğrenme modeli kullanmışlardır. Ağ kendisine verilen

örneklerden genellemeler yaparak çözüm kümesi üretir ve benzer örnekler ürettiği çözüm kümesinden yararlanır.



Şekil 7: Geri Beslemeli Çok Katmanlı Hopfield Ağı

(Elmas 2003)

Girdi elemanı ve çıktı elemanı arasındaki ilişki karmaşık ise gizli katman sayısının artırılması gerekir. Ancak ağ aşamalara ayrılmalıdır. Aksi takdirde ezberleme meydana gelir.

Geri yayılım ağlarında R_n 'den R_m 'e karmaşık 'g' yapılarını hesaplamayı sağlar. Hopfield ağları bu tip ağlardır. Bu tür ağlarda bir sinirin çıkış değeri bir sonraki sinirin giriş değeri ile bağlıdır.

$$\frac{du_i}{dt} = -u + \sum_{i=1}^n w_{ij} y_i + x_i$$

Burada $y_j = g(u_j)$ ve

$g(u_i) = u_i$, giriş katmanının aktivasyon fonksiyonu

$g(x) = \frac{1}{2} \left(\tanh \left(\frac{x}{\alpha_0} \right) + 1 \right)$ ile verilir. $\alpha_0 = 0$ (momentumun durumu) olması katı

sınırlayıcı transfer işleme karşılık gelir. α_0 değeri yeterince küçük ise ağırlıkların simetrik olması, yani; $W_{ij} = w_{ji}$ olması durumunda, Hopfield ağı

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} y_i y_j - \sum_{i=1}^n y_i x_i$$

şeklinde bir sistem enerji işlevini en küçükleyecek şekilde davranır ve bu enerji işlevi bir yerel minimuma karşılık gelen kararlı duruma ulaşır. (Kuyucu 2012)

Geri yayılım algoritmasında her katmanın ağırlık değişimi kontrol edilerek hata azalma sağlanabilir.

Örnek bir ağdaki aktivasyon fonksiyonunun sigmoid fonksiyon olduğunu düşünelim. Çıkış katmanındaki her bir nöronun aktivasyon değeri: $o_k = \frac{1}{1 + e^{-net_k}}$

K katmanının çıkış değeri: $net_k = \sum_j w_{kj} o_j$

Saklı katmanın aktivasyon değeri: $o_j = \frac{1}{1 + e^{-net_j}}$

Saklı katmanın çıkış değeri: $net_j = \sum_i w_{ij} o_i$

Ortalama Kare Hatası: $E = \frac{1}{2} \sum_p \sum_k (t_{pk} - o_{pk})^2$

$t_k =$ Beklenen çıkış değeri $o_k =$ Çıkış değeri

hatayı minimum düzeye geriletmek amacı ile hatanın ağırlıklara olan bağımlılığı hesaplanır ve gradyana bağlı (eğim, dereceli iniş) olarak ağırlıklar hatayı düşürecek şekilde yenilenir.

$$\Delta w_{jk} = -\eta \left(\frac{\partial E}{\partial w_{jk}} \right) \quad \frac{\partial E}{\partial w_{jk}} = \delta_k o_j \text{ yani} \quad \Delta w_{jk} = -\eta \delta_k o_j$$

$$\delta_j = f'(net_j) \sum_k \delta_k w_{kj} \quad \delta_k = (t_k - o_k) f'(net_k) \text{ yani} \quad \Delta w_{ij} = -\eta \delta_j o_i$$

Çıkış katmanı ağırlık yenilenmesi: $w_{jk}^{yeni} = w_{jk}^{eski} + \Delta w_{jk}$ şeklinde gerçekleşir.

Giriş ve saklı katman arasındaki ağırlık yenilemesi: $w_{ij}^{yeni} = w_{ij}^{eski} + \Delta w_{ij}$ şeklinde gerçekleşir.

δ_k = çıkış katmanı hata terimi

$$\delta_k = (t_k - O_k) f'(net_k)$$

δ_j = Saklı katman hata terimi

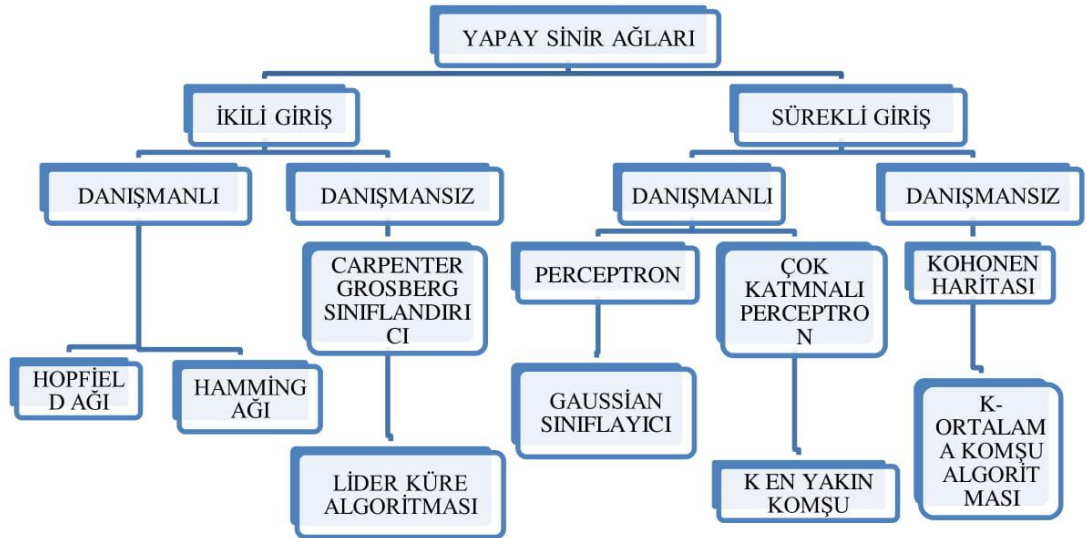
$$\delta_j = f'(net_j) \sum_k \delta_k w_{kj}$$

(Altun v.d.2012)

(The Institute Of Electrical And Electronic Engineers)

İleri beslemeli sinir ağlarında sınıflama yapılır. Hatta 'hata küçültme yöntemi' ile ağ geliştirilip iyi sonuçlar alınması da sağlanabilir. Ancak; ağların esas öğrenmesi ve kendi kendini geliştirmesi geri yayılım ağlarının üstünlüğüdür.

4.1.9. Çok Bilinen ağ yapıları ve genel özellikleri

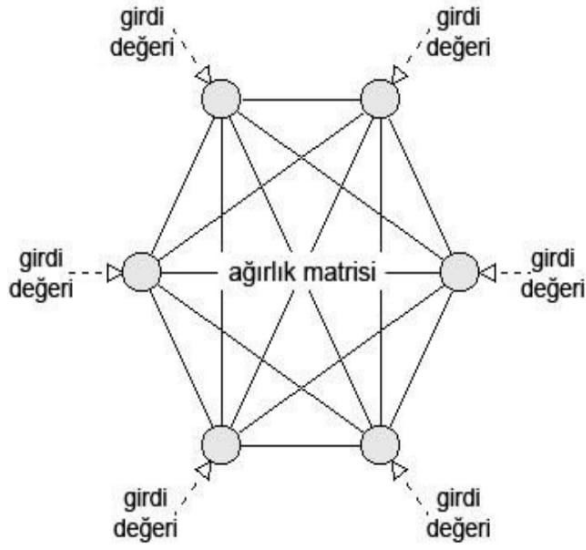


Şekil 8:Yapay Sinir Ağları Şeması

(Elmas 2003; Öztemel 2012; Saridereli 2010)

4.1.9.1. Hopfield ađı

1982'de John Hopfield tarafından geliřtirilmiřtir. Klasik geri yayılım ađındaki gibi katmanlardan oluřmaz, Probalistik Neural Network of Specht'e benzer. rnek desen grupları seer. ıkıř katmanındaki veriler aynı zamanda giriř katmanı verileri gibi alınır. Birimler arasındaki etkileřim ađrılıklar tarafından ynlendirilir. Bazı proseslerin aktif veya bazılarının pasif olması bir eřit yarıř ortamı oluřturur. Vektr uzunluđu baz alınarak sınıflama yapar. Geleneksel yntemlerle zlemeyen optimizasyon problemlerinde bařarılıdırlar. Hebb kuralından farkı; đrenmenin kuvvetlendirme ve zayıflama řeklinde genliđini belirleyebilmesidir (đrenme katsayısı).



řekil 9: Hopfield Ađ Yapısı

www.tukarpengetahuan.com

4.1.9.2. Hamming ađı

Bu ađ hopfield ađına benzer ancak daha abuk ve dođru sonular elde edilmesine olanak sađlar. ıkıř katmanında sıfır ve artı deđer sađlayan ađ desenleri uygun kabul edilir. Ađ en az hata ile ve en uygun durumda alıřır.

4.1.9.3. Grossberg sınıflayıcı

1970 te Stephen Grossberg tarafından geliştirilmiş. Uyarlanırlı Rezonans Ağı modeli olarak bilinir. Giriş ve çıkış katmanı arasında yer alan işlemci katmanını karşılıklı birleştirilmiş iki katman oluşturur. Düşük ve yüksek rezonans aktarımı mantığı ile çalışan bu işlem elemanı ağı gerçek mantığıdır. Biyoloji ve mühendislik çalışmalarında yaygın kullanılır. Ağı dezavantajı giriş vektöründe en küçük miktarda gürültü bile iyi bir ağı model eşleme yeteneğini karıştırabilir.

Robert Hecht Nielsen bu ağ yapısını radyo sinyalleri uygulamaları için kullanmıştır. Bu ağ Yığın Ağı (Spatio) veya Geçici Model ağı olarak da bilinir. Kohonen öğrenme yöntemine benzer. Yığın Ağı sinyalleri ezberleyebilme ve sinyallerin değişim zamanını ayarlayabilme yeteneğine sahiptir.

4.1.9.4. Hebbian ağ (teorisi) yapısı

İki sinir birbiri ile bağlantılı ise ve aynı anda hareket etmeye başlarsa iki sinir arasındaki bağ güçlenir. Hebb öğrenme türünde öğrenilen bilgi için bir hafıza gerekli değildir çünkü öğrenme aktiftir ve bilgi ağlarda mevcuttur. Bu öğrenmenin bir başka öngörüsü öğrenilecek bilginin farklı olmasıdır. 1962 yılında Bernard Widrow Hebb öğrenme kuramını geliştirerek Delta öğrenme kuramı denilen ve Hebb kuramından farklı olarak ağırlık toplama fonksiyonunu giriş katmanına taşıyan kuramını geliştirmiştir. Delta kuramı ileri beslemeli sinir ağlarında kullanılırken 'Delta Bar Delta' yada 'Genelleştirilmiş Delta' kuramı geri beslemeli sinir ağlarında kullanılmıştır.

4.1.9.5. Perseptron teoremi ve madaline

Perseptron öğrenme tek katmanlı ve ileri beslenmeli bir yapay sinir ağ yapısında eşik değeri kullanarak çıkış atayan bir öğrenme yöntemidir. Ağı en bilinen örneği ADALINE ağ yapısı; giriş değerlerini çıkışa iletebilen algoritmaya sahiptir. 'AND' 'OR' yönergeleri ile birçok ADALINE birleştirilerek MADALINE denilen çok katmanlı yapıya ulaşılmıştır. Buna rağmen Perceptronlar doğrusal ayırmayı başarmış ancak doğrusal olmayan verilerin sınıflandırılmasında başarısız olmuşlardır.

Perseptron ađ yapısı geliştirilerek ileri beslemeli ve çok katmanlı yapıya dönüştürülmüştür. Çok katmanlı ađlar danışmanlı öğrenme yöntemi kullanır ve giriş değerine göre genelleme yaparak sonuca ulaşır.

4.1.9.6. Kohonen ađları

1890' de, Kohonen tarafından kurulan bu ađ modeli benzer objeleri uygun kategorilere ayırma özelliđi şeklinde yapılandırılmıştır. Bu yönüyle sınıflama ve görüntü işleme problemleri için uygun yapıdadır. **Öğrenme Vektör Nicelme Ađı** bu mantıkla geliştirilmiş m-boyutlu bir uzayda n-boyutsal boşluđun haritasını çizer ve kalıtsal topolojisini koruyup girdileri sınıflandırmak üzere kendini eğitir. Önceden öğrenilmemiş olan girdi kalıpları en yakın komşu tarafından kategorize edilir. Eğitim vektörünün her bir işlem elemanına uzaklıđını hesaplar, en yakın işleme elemanını 'kazanan' olarak ilan eder ve sadece bir çıktı elemanının harekete geçmesini sağlar. Eğer kazanan eleman beklenen sınıfa ait değilse bağlantı bađlı değerleri eğitim vektöründen uzaklaşır. Tüm işlem elemanları kendi belirli sınıflarıyla bir arada bulunan bölgeye depolanır.

Kohonen Öz Örgütlemeli Sinir Ađında işlem elemanları kendilerine gelen giriş değerlerinin öklit hesabını yaparak çıkışa dođru aktarılmasını sağlar. Çıkış verileri içinde en düşük mesafeye sahip olan veri kazandır. Düşük mesafeye sahip veri hafızaya alınarak geri çağırma sürecinde Kohonen elementi olarak işlemde yer alır. Ađ yarışmacı kazanımdır bu yüzden tüm verilerin giriş değerine indirgenmesi yani öklit mesafesinin sıfırlanması için ađın sürekli eğitilmesi öngörülür.

Kohonenin Karşı Yayma Ađında ise giriş katmanında her bir girdi elemanı için bir işleme elemanı; dengeleme elemanı ekleyerek verileri kohonen katmanına girmeden önce normlaştırmış olur. Bu veriler kohonenin yarışmacı tavrı nedeniyle işleme katmanında; büyük değerli vektör küçük değerli vektörü geçer ve bađlı değeri ayarlanmış şekilde kazanç elde eden elemene bađlanarak gerçek çıktıya dönüşür.

Bu ađların dışında Donalt Spech'in ađları giriş katmanına norm uygular ve hafızasını öğrenilmiş bir kategori veya sınıfın parçası olan bir girdi vektörünün

benzerliğini tahmin etmek için kullanır. Parzen tahmini olasılık değerini güçlendirerek sınıflama yapar.

4.1.9.7. Yönlendirilmiş rastgele atama

Yukarıda anlatılan tüm ağ paradigmaları bağıl değer ayarlama ve eğimli iniş tekniği kullanma üzerine imar edilmiştir. Oysa bu ağ yapısında; rastgele atanan değerler için bir yön unsuru eklenerek bağıl değerlerin daha önceden başarılı olmuş bir arama yönüne doğru yönelmesi garanti altına alınır. Problemin iyi tanımlanmış olması ilk bağıl değer ilk tahminin en iyi bağıl değer tahmini olma ihtimalini gerçekler. Bu sınıflama çok hızlıdır çünkü; temel hesap tekniklerinden (örneğin Delta ve türevler) yararlanmaz, işleme elemanı hata terimi hesaplamaz, yalnızca çıktı hatası hesaplanır. Bağıl değeri, algoritmanın en iyi çalıştığı yer olan yoğun bölge içerisinde tutabilmeyi kolaylaştırmak için bir üst değer atanır ancak algoritmanın gerçek global optimum değeri aramasına izin verilir. Bu sınırlamanın amacı ağın hantal ve yavaş olmasını engellemektir. Eğer ağın 200 den fazla bağlantı bağılı varsa arama uzun eğitim zamanı gerektirir ama kabul edilir sonuçlar verir.

SOM komşuluk ilişkisi, COGNITRON- NEOCOGNITRON uyarıcı ve men edici şeklinde iki process-iki sinyal şeklinde, COUNTER PROPAGATION ise Kohonen ve Grosberg öğrenme kurallarını birleştirerek şekil tanıma amacıyla geliştirilmiş Geri Yayılımlı Yapay Sinir Ağ örnekleridir. Yukarıda anlatılan Genelleştirilmiş Delta ve Hopfield ağ yapıları da geri yayılım ağ yapısına sahiptir.

4.2. Destek Vektör Makinası

Destek Vektör Makineleri çok boyutlu ve doğrusal olmayan, desen tanıma problemlerinde sınıflama yapmak için geliştirilmiş ve yaygın olarak kullanılmış bir bilgisayar öğrenme yöntemidir. Ancak; büyük ölçekli eğitim setleriyle çalışmalarda Destek Vektör Makinaları verim ve pratiklik açısından uygulanabilir bulunmamıştır. Bunun en büyük nedeni modelin kompleks yapısından kaynaklanan hesaplama zorluğudur.

DVM sağlam bir teorik temele sahiptir. Daha çok basit sınıflama ve regresyon problemlerinde doğru tahminleme konusunda başarılı olmuştur. DVM çalıştığı veri setinin model yapısının karmaşıklığını dikkate alır ve modeli öğrenir. Böylece destek vektörlerinin sınırladığı optimal düzlemi saptayarak küçük örnekli çalışmalarda robuts test özelliği gösterdiği gibi genelleştirme yeteneğinde de başarılı olur. Ancak aynı başarıyı ikinci derece (quadratic) problem çözümü gerektiren büyük veri setlerinde eğitim işleminin çok yavaşlaması nedeniyle gösteremez. (Liu. v.d. 2017)

Vladimir Vapnik; 1963 yıllarında Destek Vektör Makinaları üzerine çalışmalarına başlamıştır. Corinna Cortes ve Vapnik; DVM ile ilgili çalışmalarını 1995 yılında yayınlamışlardır. Yüz tanıma, el yazısı tanıma, zaman serisi tahmin testleri ve biyoinformatik en yaygın kullanım alanlarıdır. (Platt,1998)

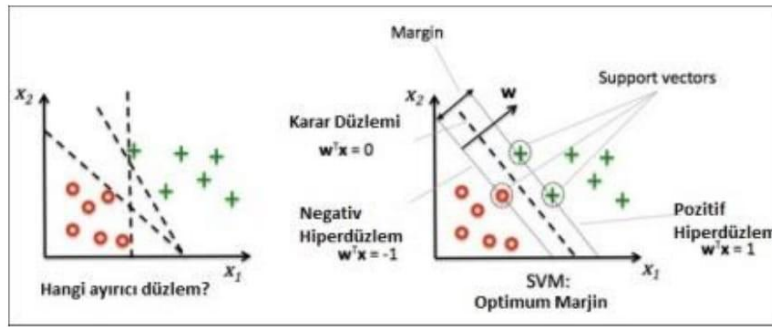
DVM yüz tanıma, el yazısı tanıma, zaman serisi tahmin testleri, tıpta hastalıkların teşhis edilmesi, biyolojide canlı türlerinin sınıflandırması, kimyada etkin ilaç maddelerinin ayrıştırılması, bankacılık sektöründe risk gruplarının saptanması gibi birçok alanda sınıflama amaçlı kullanılmıştır. (Erdoğan ve Ayhan 2016)

DVM; doğru modelin bilindiği ve sadece parametre değerlerinin araştırılması mantığına dayalı klasik istatistik yöntemlerinden farklı olarak model formunun bilinmediği ve doğru olabilecek modeller arasında en iyi tahminleyici modelin belirlenmesi esasına dayanan bir yöntemdir. Doğrusal olarak ayrılabilen verilerde sınıflar arasında sonsuz sayıda ayırıcı doğru yer alır ve DVM marjini en yüksek yapacak doğruyu seçmeyi hedefler. Doğrusal olarak ayıramadığı veri setlerinde ise orijinal çalışma verisini yüksek boyutta uzay düzlemine dönüştürerek doğrusal olmayan haritalama kullanarak veriyi taşıdığı yeni boyutta marjini en büyük (optimal) ayırıcı düzlemi araştırır. (Taylor, 2000)

DVM'nin temel varsayımlarından biri eğitim kümesindeki tüm örneklerin bağımsız ve benzer olarak dağılmış olmasıdır. Destek Vektör Makinaları; doğrusal veri setlerinde tahminleyici, doğrusal olmayan veri setlerinde ise regresyon mantığı ile çalışır. Algoritmasına bakıldığında iki katmanlı ve ileri beslemeli yapay sinir ağlarına benzemektedir. (Yakut v.d. 2014)

4.2.1. Doğrusal destek vektör makinası

Destek vektör makinası verileri optimal olarak iki kategoriye ayıran 'n' boyutlu bir hiper düzlem oluşturur. Eğitim verileri ile elde edilen $g(x) = \text{Sign}(f(x))$ fonksiyonu karar sınırları ile sınıflama yapılır. Sınıf çıktıları $y_i = \{+1 -1\}$ şeklindedir.



Şekil 10: Destek Vektör Makineleri ve Marjin Değerleri

www.picshype.com

'Negatif' ve 'Pozitif' verileri birbirinden ayıran DVM yönteminin 'Karar Ağacı' (Decision Tree) Sınıflama yönteminden farklılığı; DVM sınıflamayı ayırıcı düzlem ile Karar Ağacı ise sadece sınıflardaki homojenite ve sınıf ağırlıkları ile ayırma yapmasıdır. DVM'da ayırıcı düzlemin sınırlarını 'Pozitif Hiper Düzlem' ve 'Negatif Hiper Düzlem' belirler ve bu 'Karar Sınırları' yöntemin 'Destek Vektörlerini' (Support Machine) oluşturur. Destek Vektörlerinin ortasından geçen doğru ise 'Optimal Hiper Düzlem' olarak ifade edilir.

\vec{w} = Grafik Orijininden geçen ve Optimal Düzleme (ayırıcı doğru) dik inen (90°) ağırlık vektörüdür.

\vec{u} = Bilinmeyen vektör olarak ifade edilir. Bu vektör, \vec{w} vektörüne dik düşen görüntüye sahiptir. Görüntünün ‘+’ yada ‘-’ tarafa düşmesi Optimal Düzlemin eğimine bağlıdır.

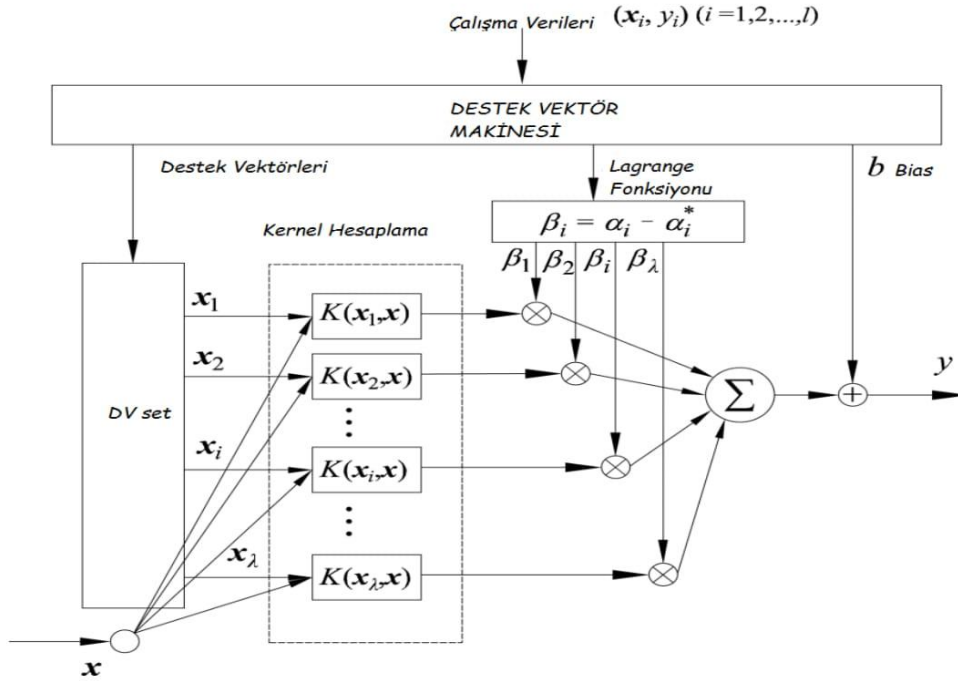
$\vec{w} \cdot \vec{u} \geq c$ eşitliği, ‘C’ sayısına ulaşmamızı sağlar. C her ‘x’ noktasını +1 veya -1 olduğunu gösteren değerdir.

$$\vec{w} \cdot \vec{u} \geq c \quad \rightarrow \quad c = -b \quad \text{öyle ise} \quad \vec{w} \cdot \vec{u} + b \geq 0$$

Bu gösterimin bir düzlem (hiperplane) üzerinde olduğunu düşünürsek; bu düzlem üzerindeki her noktanın ‘wx-b=0’ denklemi ile ifade edilmesi mümkündür. ‘x’ noktanın değişen parametresi, ‘b’ ise kayma oranı olarak kabul edilebilir.

DVM Kural 1: Eğer $\vec{w} \cdot \vec{u} + b \geq 0$ ise değer ‘+’ olur.

b = Yönelim değeri (Bias).



Şekil 11: Destek Vektör Makineleri ve Bias Değerleri

(Ayhan 2014)

DVM' nin genel yapısı incelendiğinde; ‘K(x1, x2)’ çekirdek fonksiyonlarını ve ‘α’ Lagrange çarpanlarını göstermektedir. Çekirdek fonksiyonları yardımıyla

girdilerin iç çarpımları hesaplanmaktadır. Langrange çarpanları ise ağırlıkları göstermektedir. DVM'de bir örneğe ilişkin çıktı değeri girdilerin iç çarpımları ile Langrange çarpanlarının bağımsız kombinasyonlarının toplamına eşittir.

$$\bar{w} \cdot \bar{u} \geq c \quad \rightarrow c = -b \quad \text{öyle ise} \quad \bar{w} \cdot \bar{u} + b \geq 0$$

\bar{u} bilinmeyen vektörü yerine \bar{x}_+ (pozitif 'X' örneklerinden oluşmuş vektör) ve \bar{x}_- (negatif 'X' örneklerinden oluşmuş vektör) kullandığımızda;

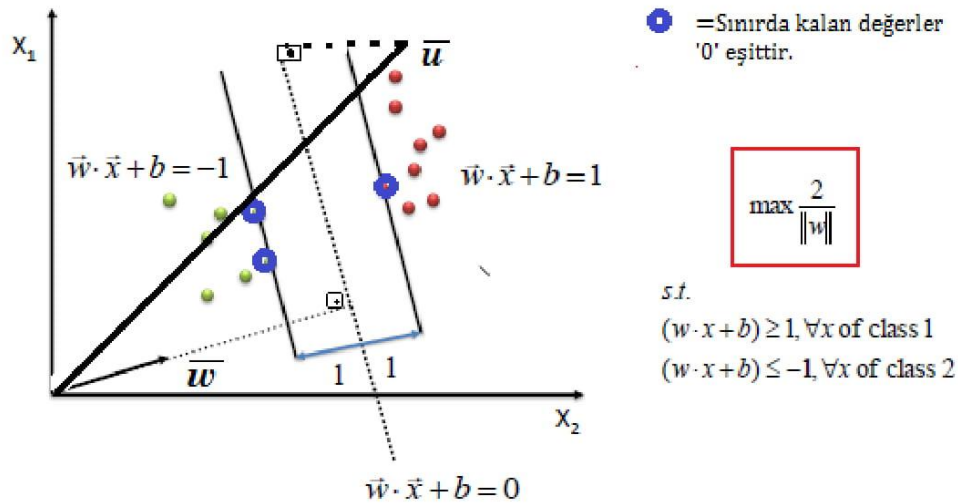
DVM kural 2:

'-' ve '+' Destek Vektörleri (Hiper düzlemin sınırları) üzerinde kalan x_i değerleri '0' sıfır değerine eşittir.

$$\bar{w} \cdot \bar{x}_i + b \geq 1 \quad y_i = +1 \quad y_i(x_i \bar{w} + 1) \geq 1 \quad y_i(x_i \bar{w} + b) - 1 \geq 0 \quad \text{POZİTİF ÖRNEKLER İÇİN}$$

$$\bar{w} \cdot \bar{x}_- + b \leq -1 \quad y_i = -1 \quad y_i(x_i \bar{w} + b) \geq 1 \quad y_i(x_i \bar{w} + b) - 1 = 0 \quad \text{NEGATİF ÖRNEKLER İÇİN}$$

Burada vektörel hesaplamada mutlak değer alındığı için negatif örnekler '-1' ile çarpım yapılarak eşitliğin yönü ve değeri değişir.



Şekil 12: Destek Vektör Makineleri Sınır Değerleri ve Sınıflama

www.blog.sina.com

DVM kural 3:

Marjin değerleri birbirine en yakın '+' ve '-' örneklere ait \bar{x}_+ ve \bar{x}_- vektörler arasındaki mesafeyi ölçer.

$$Marjin = (\bar{x}_+ - \bar{x}_-) \cdot \frac{\bar{w}}{\|w\|} \Rightarrow \frac{(1-b) + (1+b)}{\|w\|} = \frac{2}{\|w\|} \text{ aşağıdaki hesaplama göre}$$

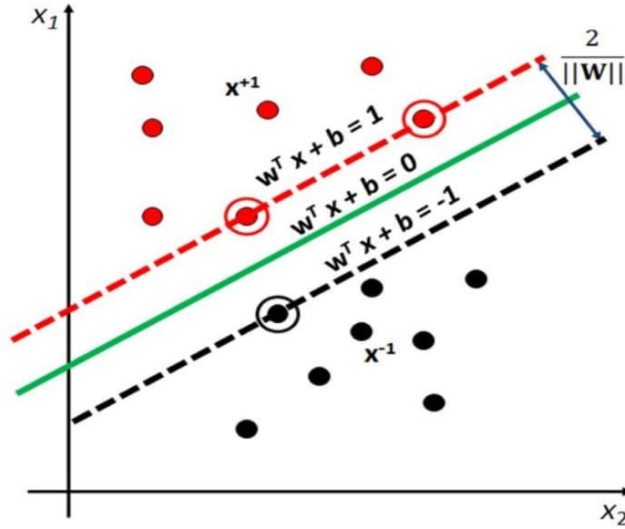
$$y_i \text{ pozitif ise } y_i(x_i \cdot \bar{w} + b) - 1 = 0 \rightarrow (x_i \cdot \bar{w}) = 1 - b \text{ Pozitif } \bar{x}_+ \text{ değerleri için.}$$

$$y_i \text{ negatif ise } y_i(x_i \cdot \bar{w} + b) - 1 = 0 \quad (-x_i \cdot \bar{w}) = 1 + b \text{ Negatif değerde parantezin içi '-' ile çarpılır.}$$

Marjinin maximum olduğu değer minimum $\|w\|$ değere eşittir. (Bamakan,2016)

$$Max \frac{2}{\|w\|} = Min \|w\| \Rightarrow Max = Min \frac{1}{2} \|w\|^2 \text{ Marjinin Maksimum genişliğini ölçer.}$$

(Erdoğan ve Ayhan 2016)



Şekil 13: Destek Vektör Makineleri Maksimum Marjin Genişliği

www.picshype.com

Total $\|w\|$ değerini minimize etmek ayrılabilirliği maksimize etmek demektir. Bunun için Lagrange Çarpanı ' α ' kullanılır. Lagrange çarpanı bir $g(x, y) = c$ denklemi ile sınırlandırılmış $f(x, y)$ işlevinin fonksiyonunun en büyük değeridir. Başka bir ifade ile bu çarpan; $f(x, y)$ fonksiyonu ile $g(x, y) = c$ fonksiyonunun teğet noktasıdır.

$$L(x, y, \alpha) = f(x, y) - \alpha(g(x, y) - c) \quad (\text{Tong ve Kaller 2001})$$

DVM' ni optimize etmek için fonksiyonun Maximum ve Minimum noktalarını bulmak gerekir. Optimizasyon için Lagrange fonksiyonunun \bar{w} ve 'b' ye göre türevlerini almamız gerekir.

$L = \frac{1}{2} \|w\|^2 - \sum \alpha_i [y_i(\bar{w} - \bar{x}_i + b) - 1]$ fonksiyonunun 1.dereceden türevinin alınması ile Krush-Kuhn-Tucher koşulları gerçekleşir.

KKT koşulları; eşitsizlik olan durumlara belirli kısıtlar eklenerek fonksiyonun genelleşmesini sağlar. Bu genelleştirme için ' $\alpha_i \geq 0, \forall_j$ kısıtı' optimizasyon problemine eklenmelidir. Bu kısıta KKT tamamlayıcı (complementary) koşulu adı verilir.

$$\frac{\partial L}{\partial w} = \bar{w} - \sum \alpha_i y_i x_i = 0 \quad \Rightarrow \quad (\bar{w} = \sum \alpha_i y_i x_i)$$

$$\frac{\partial L}{\partial b} = -\sum \alpha_i y_i = 0 \quad \Rightarrow \quad \sum \alpha_i y_i = 0 \quad (\text{Suykens, v.d. 2002})$$

Lagrange çarpanları \bar{w} ve b ye göre minimize edilmeli iken α_i değerleri için maximize edilmelidir. Bu problem primal uzayda (w ve b göre) ya da dual uzayda (α) çözülebilir. Primal çözümün kısıtları dual probleme göre daha karmaşık olduğundan pratikte dual quadratik (ikinci derece) problem çözülmektedir ve verdikleri sonuç aynıdır. Problem çözümü 'Wolfe dual' olarak adlandırılmaktadır ve bir kuadratik programlama problemi olarak elde edilmektedir. (Bayraktar ve Cebi 2013)

DVM 4. Kural

$$L = \frac{1}{2} \left(\sum \alpha_i y_i \bar{x}_i \right) \left(\sum \alpha_j y_j \bar{x}_j \right) - \left(\sum \alpha_i y_i \bar{x}_i \right) \cdot \left(\sum \alpha_j y_j \bar{x}_j \right) - \sum \alpha_i y_i b + \sum \alpha_i$$

$$L = \sum \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j \quad \text{wolfe dual teorisi}$$

Alternatif tüm vektörler ' x_i, x_j ' örneklerine bağlı olarak değişir. Belirleyici ' x_i, x_j ' dir.

Dual Lagrange $L_D(\alpha)$ 'yı maksimum yapan (α_i) değerleri elde edilir. α_i Lagrange çarpanlarından sıfırdan büyük değer alan eğitim örnekleri 'destek vektör'

olarak ifade edilir. Optimal ayırma hiper düzlemi, sıfırdan büyük olan bu Lagrange çarpanları ile belirlenir. (Erdoğan ve Ayhan 2006 ; Liu v.d., 2017)

4.2.2. Doğrusal olmayan destek vektör makinası

Doğrusal destek vektöründe kullanılan 'x' yerine ' $\Phi(x)$ ' kullanılır.

$$\text{Karar Fonksiyonu} = y_i \left((\Phi(x) \cdot w) + b \right) - 1 \geq 0$$

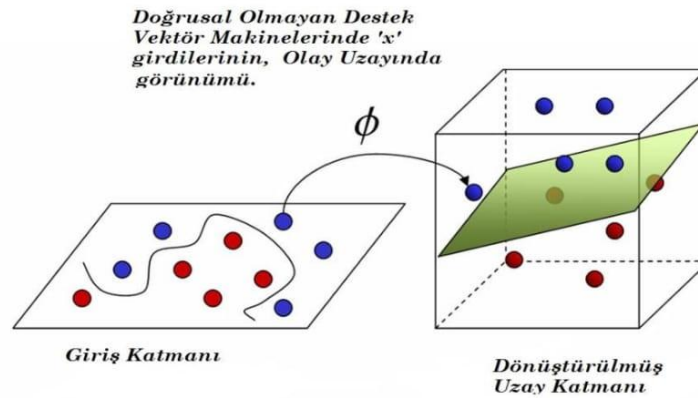
$$\text{Marjin Kısıtı} = \min_{w,b} r(w) = \frac{1}{2} \|w\|^2$$

Doğrusal olmayan DVM'de 'w' ve 'b' parametresinin hesabı

$$w = \sum \alpha_i y_i \Phi(x_i) \quad (\text{Cortes ve Vapnik 1995})$$

$$f(x) = \left(\sum \alpha_i y_i \Phi(x_i) \cdot \Phi(x) \right) + b = 0$$

Pratikte verilerin doğrusal olarak ayrılabilirdiği durumlarla pek karşılaşmamaktadır ve bu durumda pratik uygulamaların çoğu, doğrusal DVM ile çözülememektedir. DVM'ler doğrusal olmayan verilerle karşılaştığında, orijinal verilerden sınıflandırma özelliklerini çıkartmak için, doğrusal olmayan haritalama (mapping) yaparak, verileri n boyutlu orijinal girdi uzayından daha yüksek boyuta sahip belirleyici nitelik (feature)uzayına taşır. (Yılmaz 2013)



Şekil 14: Doğrusal Olmayan Destek Vektör Makineleri Gösterimi

www.cs.toronto.edu

Çekirdek fonksiyonu olan Kernel Fonksiyonu; dönüştürülmüş uzayda iki vektörün iç çarpımı şeklindedir.

$$K(x_i, x_j) = \Phi(\bar{x}_i) \cdot \Phi(\bar{x}_j) \quad \text{Kernel Fonksiyonu}$$

Kernel Fonksiyonu transfer fonksiyonunu bilmediği diğer evrenin (uzayın) açıklanmasını sağlar. Kernel fonksiyonu aşağıdaki şekillerde de formüle edilebilir.

$$K(u, v) = \Phi(u) \cdot \Phi(v) = (u \cdot v + 1)^2 \quad \text{Kernel Fonksiyonu} \quad (\text{Cortes ve Vapnik 1995})$$

4.2.3. DVM’de kullanılan kernel çekirdek fonksiyonları

I. Radyal Tabanlı Fonksiyon Kerneli:

$$K(x_i, x_j) = e^{-\gamma \|x - x_i\|^2} \quad \text{Radyal Tabanlı Çekirdek Fonksiyon}$$

Kernel Boyutu (γ)

II. Polinom Kerneli

$$K(x_i, x_j) = ((x \cdot y) + 1)^d \quad \text{Polinom Çekirdek Fonksiyon}$$

d = Polinom Derecesi

III. Normalleştirilmiş Polinom Kerneli

$$K(x, y) = \frac{((x \cdot y) + 1)^d}{\sqrt{((x \cdot x) + 1)^d ((y \cdot y) + 1)^d}} \quad \text{Normalleştirilmiş Polinom Kerneli}$$

Polinom Derecesi d

IV. Pearson VII (PUK) Kerneli

$$PUK = \frac{1}{\left[1 + \left(\frac{2 \cdot \sqrt{\|x - y\|^2 \sqrt{2^{(1/\omega)} - 1}}}{\sigma} \right)^2 \right]^\omega} \quad \text{Pearson VII (PUK) Kerneli}$$

(Kavzođlu 2010)

' σ ' yeterince küçükse doğrular kırılabilir ve farklı uzaylar yaratabilir. (Datong 2012)

V. Sigmoid Kernel

$K(x_i, x_j) = \tanh(\beta_0 x_i^T x_j + \beta_i)$ Çok Katmanlı Algılayıcı Çekirdek Fonksiyonu

(Ercan 2011)

4.3. Lojistik Regresyon Analizi

Bir arařtırmacı bir olayı incelerken bu olayı oluřturan birden ok etkenle karřılařabilir. Bu etkenlerin her birinin tek tek veya hepsinin birlikte bağımsız deęiřken zerindeki etkisini aıklamak iin istatistik yntemlerinden yararlanır.

Birlikte etkinin incelenmesinde kullanılan deęiřik istatistik yntemler bulunmaktadır. rneęin; X bağımlı deęiřkenin lm biiminin sayısal (numerik), Y cevap deęiřkenlerin kategorik zellikte olması durumunda varyans analizi, her iki deęiřkenin kategorik olması durumunda “log-linear model”ler, yine her ikisinin sayısal lmle elde edildięi durumda regresyon analizi yntemleri tercih edilebilir. (Kuyucu 2012)

Korelasyon katsayısı "r" bağımlı ve bağımsız deęiřken arasındaki iliřkinin yn ve kuvveti hakkında bilgi verirken Regresyon Analizi bağımlı ve bağımsız deęiřkene ait noktaların grafik zerinde gsterimi ile iki deęiřkenin aynı anda artıyor veya azalıyor olması veya ikisi arasında hibir iliřkinin olmaması durumunun doęrusal olarak gsterimini ve iliřkinin doęru denklemi ile aıklanmasını saęlar. Bu iliřkinin doęru denklemi ile aıklanması ise "X" bağımsız deęiřkeninin her bir birimlik artıřında "Y" bağımlı deęiřkeninde oluřacak deęiřimi yani ‘Y’nin olasılıklarını tahminlemeye yarar. Bu nedenle Regresyon analizi hem tanımlayıcı hem de ıkarımsal istatistik zellięi gsterir.

Normal daęılıma sahip aralıklı/oranlı verilerle oluřturulmuř iki deęiřken arasındaki doęrusal iliřkiyi test etme **Basit Doğrusal Regresyon Analizi** ile saęlanır. Birden fazla tahmin deęiřkeninin bağımsız deęiřken zerindeki etkilerini aıklama **oklu Regresyon Analizi** yani oklu Regresyon Denklemi oluřturularak saęlanır.

Doęrusal regresyon analizinde cevap deęiřkeni sayısal (numerik) veya oransal (interval) yapıdadır. Cevap deęiřkeninin ikili (dikhotomous) deęiřken zellięi gsterdięi regresyon analizleri Lojistik Regresyon modeli ile saęlanır.

Logistik Regresyon ve sınıflama aęaları bağımsız deęiřken zerinden her hangi bir ıkarım olmaksızın, kategorik bağımlı deęiřkenlerin sınıf yelięini tahminlemede kullanılan modellerdir. (Camdeviren v.d. 2007)

Doęrusal regresyon analizinde bağımlı deęiřkenin alabileceęi deęerler tahmin edilirken Lojistik Regresyon analizinde bağımlı deęiřkeninin alacaęı deęerlerden

birinin gerekleşme olasılıđı tahmin edilerek risk faktörlerinin belirlenmesi sağlanır. (Ocakoglu 2006)

Dođrusal Regresyon analizlerinde bađımsız deđişkenlerin normal dađılım özelliđinde olması gerekirken Lojistik Regresyon analizinde normal dađılım ön koşulu aranmaz. (Ürük 2007)

Dođrusal Regresyon Analizinde ilişkinin dođrusal olması beklenir ancak Lojistik Regresyonda bađımlı deđişkenin kategorik özellik göstermesi nedeniyle bu durum göz ardı edilir. Bunu çözenin yolu verileri logaritmik dönüşüme uyarlamaktır. Sonuç olarak bađımlı deđişken sıfıra yakınsa deđişkenin olma olasılıđı düşük, bire yakınsa olma olasılıđı yüksek çıkacaktır.

Lojistik Regresyon analizi, kümeleme analizi veya ayırma analizinden farklı olarak en çok olabirlik yöntemi kullanır ve beklenen deđerleri üstel fonksiyon olarak tanımlayarak dođrusal gösterimini elde eder.

Lojistik fonksiyon ve Lojistik eđri 1844 veya 1845 yıllarında Pierre François Verhulst tarafından popülasyon büyümesinde sigmoid eđri olarak kullanıldı. Bazı popülasyon büyümelerinde 'S' şekli modeli olarak kullanılmıştır. Büyüme eđrisi yaklaşık olarak ekponansiyel bir eđridir. (Çelik 2015)

Lojistik Regresyon modelleri ilk olarak Berkson (1944) tarafından epidemiyolojik denemelerde kullanılmıştır. Berkson tarafından kullanılan Lojistik Regresyon modeli Cox (1970) tarafından yeniden modifiye edilerek farklı uygulamalar üzerinde denenmiştir. 1979-1983 yılları arasında Anderson tarafından da kullanılarak yaygınlaşmıştır. (Brown v.d. 2002)

Lojistik Regresyon analizinin sađlık bilimlerinde kullanım amacı:, cevap deđişkeninin var-yok, evet-hayır gibi ikili kategorik deđişken özelliđi gösterdiđi durumlarda veya cevap deđişkeninin ikiden fazla kategorik düzey içerdiđi ve bađımsız deđişkenlerin ise hem kategorik hem de sayısal olduđu durumlarda en az deđişkeni kullanarak en iyi uyuma sahip olacak şekilde bađımlı ile bađımsız deđişkenler kümesi arasındaki ilişkiyi tanımlayıp olasılıklarına göre verilerin ait oldukları gruplara en dođru şekilde atanmasını ve hastalıklara ilişkin risk faktörlerinin belirlenmesini sađlamak ve sonuçları istatistiksel olarak kabul edilebilir bir modelle sunmaktır.

Lojistik Regresyon analizinde açıklayıcı değişkenin sürekli veya kesikli olabilmesi, birçok paket programda yer alması, yarı parametrik olması nedeniyle parametrik test zorunluluklarını gerektirmemesi, anlaşılır bir matematiksel işlem yeteneği ve sonuçların kolay yorumlanabilir olması, özellikle epidemiyoloji biliminin kullandığı terimlerle (risk-olasılık gibi...) kendini ifade ediyor olması sağlık alanında Lojistik Regresyon yönteminin tercih edilmesinin diğer gerekçeleridir.

Lojistik Regresyon hayvan ve insan hastalıklarının hemen her türünde risk faktörlerini belirleme ve tedavi etkinlikleri saptama çalışmalarında kullanılmıştır. Özellikle kanser çalışmalarında yöntem kullanımının çok yoğunlaştığı görülmektedir. Yerleşim yerine veya mevsimlere göre hastalık dağılımı şeklindeki epidemiyolojik çalışmalarda, ilaç etken maddesinin seçiminde, mesleklere yönelik memnuniyetin belirlenmesinde, hizmet kalitesi geliştirme araştırmalarında ve her türlü sonucun (başarı gibi) ve sonucun oluşmasındaki etkenin araştırıldığı sınıflama çalışmaları gibi binlerce çalışmada kullanılmıştır.

4.3.1. Lojistik regresyon yapısal özellikleri

Regresyon analizinde amaç bağımlı değişkenin bir birimlik artışından bağımsız değişkenin alacağı ortalama değeri tahmin etmektir. Bu değer koşullu ortalama olarak ifade edilir ve $E(Y \setminus x)$ şeklinde gösterilir. **Doğrusal Regresyon Koşullu Ortalaması;**

$$E(Y \setminus x) = \beta_0 + \beta_1 x \quad -\infty < x < +\infty \text{ şeklinde gösterilir.}$$

Lojistik Regresyon yönteminde ise koşullu ortalama 0 ve 1 arasında değer almak zorundadır.

$$[0 \leq E(Y \setminus x) \leq 1]$$

Koşullu ortalama $E(Y \setminus x) = \beta_0 + \beta_1 x$ eşitliğinin sol tarafı sonsuz sayıda bağımsız değişken ile ilişkilendirilebilmektedir. Bu durumda sonuç değişkeninin 0 ve 1 arasında olasılık değeri alması için bir takım dönüşümler uygulanmalıdır. Bu dönüşümlerden en yaygın kullanılanı lojit ve probit dönüşümlerdir. (Arı ve Önder 2012)

$$f(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}} \quad \text{Logistik Regresyon Fonksiyonu}$$

Eşitlikte $z = -\infty$ iken $f(z) = 0$ ve $z = +\infty$ iken $f(z) = 1$ değerine sahip olur. Sonuç olarak $0 < f(z) < 1$ aralığında yer alır. Böylece olasılık grafik üzerinde 'S' şeklinde bir gösterime sahip olur.

X bilindiğinde Y'nin **Koşullu Ortalama Olasılık Değeri** $E(Y|x) = \pi(x)$ şeklinde ifade edilebilir;

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

Bu olasılık değerine Lojistik Regresyon yönteminde **Logit Dönüşüm** uygulanır ve aşağıdaki eşitlik elde edilir.

$$g(x) = \ln[\pi(x)/(1-\pi(x))] = \beta_0 + \beta_1 x$$

Eğer $\pi(x)$ olma olasılığı olarak ifade edilecekse $1 - \pi(x)$ olmama olasılığıdır ve Lojistik modele uygulanan Logit dönüşüm; $\pi(x)$ olma olasılığının $1 - \pi(x)$ olmama olasılığına oranıdır ve **ODDS** oranı olarak ifade edilir. (Çelik, 2015)

Logit $\pi(x) = \ln\left[\frac{\pi(x)}{1-\pi(x)}\right]$ dönüşümünün özellikleri;

- $\pi(x)$ arttıkça $\text{Logit}(\pi(x))$ değeri de artar.
- $\pi(x)$ değeri 0 ve 1 arasında değer alırken, $\text{Logit}(\pi(x))$ değeri bütün gerçel doğru üzerinde değer alabilir.
- $\pi(x) < 0.5$ ise $\text{Logit}(\pi(x)) < 0$ ve $\pi(x) > 0.5$ ise $\text{Logit}(\pi(x)) > 0$ şeklinde değer alır ve sınıflara atama bu değerler üzerinden gerçekleşir. (Ünal 1996)

Lojistik Regresyon eğrisini açıklamakta yaşanabilecek zorluklar Hosmer ve Lemeshov olarak ifade edilen logit dönüşüm uygulaması ile kolayca tanımlanabilir. (Göksülük 2011)

Dönüşümdeki esas amaç $g(x)$ değerinin Doğrusal Regresyon modelinin tüm gerekliliklerini yerine getirmesini sağlamaktır.

Yanıt değişkeninin ikili düzeyde olduğu durumlarda kullanılan bir başka model **Probit Model**dir. Logit modelde hesaplama kolaylığı olması ve özel tablolara ihtiyaç duyulmamasına rağmen Probit modelin bazı koşullarda özellikle tercih edildiği gözlenmektedir. Gerek teoride gerekse uygulamada, örneğin; biyolojik

verilerin bazı özelliklerine ilişkin dağılımlarının normal olasılık yoğunluk fonksiyonuna uyum göstermesi, hesaplamada normal kümülatif dağılım fonksiyonunu kullanan Probit modelin kullanılmasını özendirir. Probit regresyon analizi ile Lojistik Regresyon birbirlerine oldukça benzer ve elde edilen olasılık tahminleri birbirlerine yakın değerdedir. Lojistik Regresyon analizinde log-odds (olabilirlik oranları) kullanılırken, Probitte kümülatif normal dağılım kullanılmaktadır. Temel olarak Probit birikimli standart normal dağılımın tersidir. Aşağıdaki şekilde ifade edilir

$$Y_i = \alpha + \beta X_i + U_i$$

$Y_i > 0$ ise $Y_i = 1$ ve $Y_i < 0$ ise $Y_i = 0$ olur. Burada eşik değer sıfırdır. Normal standart değişken Z için $\Phi(z) = P(Z \leq z)$ olarak tanımlanırsa

$$P(Y_i = 1) = P(u_i > -\alpha - \beta X_i) = 1 - \Phi\left(\frac{-\alpha - \beta X_i}{\sigma}\right) \quad (\text{Arı ve Önder 2012})$$

$$P(Y_i = 0) = P(u_i \leq -\alpha - \beta X_i) = \Phi\left(\frac{-\alpha - \beta X_i}{\sigma}\right)$$

Yukarıdaki eşitlikte; βX probit değerini, Φ standart normal dağılım olasılığını ifade eder. β probit kat sayısı bağımsız değişkenin bağımlı değişkene ait standart Z değerinde yapacağı etkiyi ölçer.

Doğrusal Regresyon yönteminde sonuç değişkeni $y = E(Y \setminus x) + \varepsilon$ şeklinde ifade edilir. Burada ε hata terimini doğrusallıktan ayrılışı simgeler ve 0 ortalama ve σ^2 sabit varyans değerine sahiptir. Oysa Lojistik Regresyon yönteminde ε hata terimi mümkün olan iki değerden başka değer alamaz. $Y=1$ ve $\pi(x)$ olasılıkta $\varepsilon = 1 - \pi(x)$ ve $Y=0$ ve $1 - \pi(x)$ olasılık için $\varepsilon = \pi(x)$ değerini alabilecektir. ε sıfır ortama ve $\pi(x)(1 - \pi(x))$ varyanslı binom dağılım özelliği gösterir. (Akçay 2009)

4.3.2. Logisitik regresyon teknikleri

Lojistik Regresyon teknikleri aşağıdaki gibi sınıflayabiliriz.

- İkili (Binary) Lojistik Regresyon Tekniği (Simple Logistic Regression)
- Kategorik Lojistik Regresyon Tekniği (Multinomial Logistic Regression)
- Sıralı (Ordinal) Lojistik Regresyon Tekniği
- Kernel Logisitik Regresyon

4.3.2.1. İkili (Binary) lojistik regresyon tekniği

İkili Lojistik Regresyon tekniğinde bağımlı değişkenin ikili düzeyde sınıf değişkenine sahip olduğu veya ikili değere indirgendiği durumlarda kullanılır.

$$P(Y_j = 1) = \frac{e^{\beta_0 + \beta_1 X_{j1} + \dots + \beta_k X_{jk}}}{1 + e^{\beta_0 + \beta_1 X_{j1} + \dots + \beta_k X_{jk}}}$$

Burada $P(Y_j = 1)$, j biriminin 1. kategoride olma veya birinci kategoriye seçme olasılığıdır. (Alic ve Emilsson 2016)

4.3.2.2. Multinomial (kategorik) lojistik regresyon tekniği

Bağımlı değişkenin medeni durum, mezun olunan okullar, göz rengi gibi kategorik değişken özelliğinde ve en az üç düzey içerdiği Lojistik Regresyon tekniğidir. İkili Lojistik Regresyon Tekniğinin genelleştirilmiş halidir.

Bu modelde ikiden fazla kategori olduğu için ilk ve son kategoriler referans kategori olarak belirtilmelidir.

$$P(Y_j = h) = \frac{e^{\beta_{h0} + \beta_{h1} X_{j1} + \dots + \beta_{hk} X_{jk}}}{1 + \sum_{h=1}^{M-1} e^{\beta_0 + \beta_1 X_{j1} + \dots + \beta_k X_{jk}}}$$

Burada $P(Y_j = h)$, j biriminin h kategoride olma olasılığıdır. (Dolgun 2012)

Eğer bağımsız değişkenler kesikli ve nominal ölçekli ise bu değişkenleri sürekli değişkenlermiş gibi modele dahil etmek yanlış olacaktır. Çeşitli düzeyleri göstermek için kullanılan sayılar sadece tanımlayıcıdır 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 (Brown v.d. 2002)

Lojistik Regresyon Analizinde sayı ile ifade edilmiş sınıflayıcı değişkenin kategorik olduğu Define Categorical Variable seçeneğinde belirtilmelidir.

4.3.2.3. Sıralı (ordinal) lojistik regresyon tekniği

Sıralı Lojistik Regresyon tekniğinde bağımlı değişken en az üç düzey içerdiği gibi bu düzeyler arasında küçükten büyüğe doğru bir sıralanış söz konusudur. Örneğin kanser evreleri (birinci evre, ikinci evre, üçüncü evre), gelir düzeyi (düşük,

orta ve üst düzey gelir), deprem şiddeti (düşük, orta, şiddetli deprem düzeyi) gibi...
(Bender 1997)

$$\text{link}(Y_j) = \frac{r_j [\beta_0 + \beta_1 X_1 + \dots + \beta_k X_k]}{\exp(\theta_0 + \theta_1 Z_1 + \dots + \theta_l Z_l)}$$

Burada Y_j , j kategori için olasılık değerini, r_j ; j kategori için eşik değerini, $\beta_0 + \beta_1 + \dots + \beta_k$; regresyon katsayılarını, $X_1 + X_2 + \dots + X_k$; yer parametrelerini, β ve θ ; bilinmeyen yer ve ölçek parametre vektörlerini, Z_l ; ölçek parametreleri için açıklayıcı değişkenleri temsil eder. (Ayhan 2006)

Sıralı Lojistik Regresyon analizinin uygulandığı beş düzeyli bir memnuniyet anketi çalışması varsayıldığında; bu çalışmada yaz aylarındaki memnuniyetin kış aylarına kıyasla çok yüksek olduğu saptanırsa mevsimsel farklılığın ölçek bileşeni olarak (θZ_l) paydada belirtilmesi gerekmektedir.

Sıralı Lojistik Regresyon modelinden elde edilen bilginin doğruluğu ve güvenilirliği için paralel eğriler varsayımının kesinlikle sağlanması gerekir. Eğer bu varsayım karşılanmazsa elde edilen tüm sonuçlar anlamsız ve yanlış olacaktır. Paralel eğriler varsayımının geçerliliğini kontrol etmek için Wald ki-kare testi (olabilirlik oran testi) gibi testler kullanılmaktadır. (Çolak, 2002)

Çoklu Regresyon analizlerinde sürekli açıklayıcı değişkenler ortak değişken olarak, isimsel açıklayıcı değişkenler ise faktör değişken olarak ifade edilir.

4.3.2.4. Kernel lojistik regresyon

Kernel Logisitk Regresyon yöntemi yarı parametrik bir modeldir. Açıklayıcı değişkenler arasındaki fonksiyonel yapının doğrusal olduğu varsayılmaktadır. Büyük veri setlerinde tahmin ve yorumlamada zorluklar yaşandığı için modelin kullanımı çok yaygın değildir. (Zhu ve Hastie, 2004)

Kernel Lojistik Regresyon $K: R \rightarrow R$ tanımlı fonksiyonu üç şekilde karşımıza çıkar:

$$\text{Yaygın Kullanımı } K(x) = \begin{cases} 1/2 & \text{if } |x| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Gaussian Kernel } K(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2 / 2)$$

$$\text{Epanechnikov Kernel } K(x) = \begin{cases} 3/4(1-x^2) & \text{if } |x| \leq 1 \\ 0 & \text{else} \end{cases}$$

Kernel seçeneklerden birini tercih ettikten ve h genişlik bandını belirledikten sonra,

Kernel Regresyon aşağıdaki şekilde tanımlanır:

$$w(x, x_i) = \frac{K\left(\frac{x_i - x}{h}\right)}{\sum_{j=1}^n K\left(\frac{x_j - x}{h}\right)}$$

Kernel Regresyon Tahminleyicisi ise:

$$\hat{r}(x) = \frac{\sum_{i=1}^n K\left(\frac{x_i - x}{h}\right) \cdot y_i}{\sum_{i=1}^n K\left(\frac{x_i - x}{h}\right)} \quad (\text{Tibshirani, 2014})$$

şeklinde tanımlanır. Burada Gauss Kernel yöntemini x_i yüksekliklerini kullanarak y_i ile ilişkili ağırlıklı ortalamayı bulmayı amaçlar. $\hat{r}(x)$ tahmincisi x_i değerleri ile doğrusal değişim gösterir. Kernel farklı tahmin yöntemleri kullansa da en yaygın kullanımları Gaussian ve Epanechnikov Kernel tahminleyici değerleri arasında büyük farklı sonuçlar gözlenmez.

h genişlik bandı aşağıdaki şekilde tanımlanır:

$$h = \frac{C_2}{2C_1 n^{1/3}}$$

Kernel fonksiyon Destek Vektör Makinelerinde kullanımı güvenilir düzeyde tüm örnekleri doğru sınıflara atayarak genelleme yapmaya izin verir ve Binary

değişkenlerin daha iyi ölçeklenebilmesini sağlar. Logisitik Regresyon yönteminde Kernel kullanımı sınırlandırılmamış ve doğrusal yapıdaki nesnelere olasılığının hesaplanmasında başarılı sonuçlar verir. (Liala 2015)

Multinomial Logisitik Regresyon yönteminde ikiden fazla kesikli veri varlığında Ridge nominal tahminleyici kullanarak kestirim varyansının küçültülmesi amaçlanır. Kernel Lojistik Regresyonda ise Lambda değeri Ridge değeri gibi kullanılır. Böylece tüm değişkenleri modele alınmasını sağlar ve çoklu doğrusal bağlantının varlığında EKK yöntemi kestirimlerinden daha küçük varyanslı parametre kestirimlerinin elde edilmesini ve modelden gereksiz değişkenlerin çıkarılması sağlanır. (Akkuş v.d. 2008)

4.3.3. Lojistik regresyon tahmin yöntemleri

İki gruplu bir lojistik modelin katsayılarının tahmin edilmesinde kullanılan yöntemler;

4.3.3.1. En küçük kareler yöntemi (RILS)

Doğrusal regresyon analizinde en çok kullanılan tahmin yöntemidir. En küçük kareler tekniğinde amaç hata kareler toplamını minimum yapacak regresyon denkleminin elde edilmesidir.

$$\sum e^2 = \sum \left(Y - \hat{\beta}_0 - \hat{\beta}_1 X_1 - \hat{\beta}_2 X_2 \dots \hat{\beta}_k X_k \right)^2$$

fonksiyonun her bir parametresine ait türevin alınması ile elde edilen k tane denklemin birlikte çözümü ile hata terimleri kareler toplamını minimum yapacak denklem elde edilmiş olur.

Doğrusal Regresyon yönteminde u_i hata terimleri normal dağılıma sahiptir ve aritmetik ortalaması 0'dır, Y bağımlı değişkenin aldığı değerler hata terimlerine bağlıdır ve dağılımları benzer özelliktedir. Fakat hata terimleri ve X bağımsız değişken değerleri birbirinden etkilenmez. u_i hata terimleri arasında otokorelasyon yoktur ve eşit varyans değerlerine sahiptir. Açıklayıcı değişken ölçüm değerinde hata mevcut değildir. Hatalar u_i değerinde toplanır. Tahmin yönteminde tüm

değişkenlerin modele alındığı, doğru matematiksel modelin kurulduğu ve ölçümde hata olmadığı varsayılır.

4.3.3.2. En çok olabilirlik yöntemi (ML)

Doğrusal regresyon analizlerinde En Küçük Kareler Yöntemi ve En Çok Olabilirlik Yöntemi aynı olasılık değerlerini veren farklı metotlardır. Doğrusal olmayan modellerde bilinmeyen parametre tahminlerinin elde edilmesinde en çok kullanılan yöntem En Çok Olabilirlik Yöntemidir. Koşulsuz (unconditional) ve Koşullu (conditional) En Çok Olabilirlik Yöntemleri şeklinde iki metot mevcuttur. Parametre değerleri denek sayısına göre az ise koşulsuz metot parametre değerlerinin denek sayısından fazla olduğu durumlarda ise koşullu metot uygulanmalıdır. Y_i gözlem değerleri Bernouli dağılım özelliği gösteriyor ise fonksiyon aşağıdaki şekilde kurulur;

$$P(Y = y_i) = f_i(y_i) = \pi \binom{n}{x_i}^{y_i} \left(1 - \binom{n}{x_i}\right)^{1-y_i}$$

Y_i gözlem değerlerinin bağımsız olduğu durumlarda ise fonksiyon aşağıdaki şekilde kurulur;

$$L(\beta) = \prod_{i=1}^n \pi \binom{n}{x_i}^{y_i} \left(1 - \binom{n}{x_i}\right)^{1-y_i}$$

Bu fonksiyonu maksimize etmek için tüm parametre değerlerinin türevlerinin alınması gerekir. Tüm sonuç eşitlikleri sıfıra eşitlenir ve aynı anda çözümlenir ve bu denkleme olasılık denklemi denir. Olasılık denklemi aşağıdaki şekilde kurulur;

$$\sum_{i=1}^n \left[y_i \pi \binom{n}{x_i} \right] = 0$$

4.3.3.3. Minimum lojit ki-kare yöntemi (MLCS)

2x1 çapraz tablolardan yararlanılarak beklenen ve gözlenen logit değerleri arasındaki farktan elde edilir.

Gözlenen veri kümesinin olasılığını maksimize eden bilinmeyen parametre değerlerini elde etmeyi amaçlar. Bu yöntem; varsayılan Y dağılımını ve Lojistik fonksiyonu örnek verilerle en tutarlı olasılık katsayılarını elde etmek için kullanır.

Gözlenen Y_i değerlerinin toplamı, beklenen değerlerin toplamına eşittir.

$$\sum_{i=1}^n Y_i = \sum_{i=1}^n \pi(x_i)$$

$Y_i = 1$ ve $X = x_i$ durumunda, $\hat{\pi}(x_i)$ değeri en çok olabirlik tahmin değeridir ve bir başka ifade ile aşağıdaki şekilde gösterilebilir;

$$\hat{\pi}(x_i) = \frac{\exp\left(\hat{g}(x_i)\right)}{1 + \exp\left(\hat{g}(x_i)\right)} \quad (\text{Vupa, 2009})$$

4.3.4. Lojistik regresyon modelinde kat sayıların önemlilik testleri

Lojistik Regresyon yönteminde tahmin edilen β katsayıların önemliliği $H_0 : \beta = 0$ hipotezinin test edilmesi ile sağlanır. Bu amaçla kullanılan yöntemler aşağıdaki gibidir:

1. Olabirlik oran (Likelihood Ratio) testi,
2. Wald test istatistiği,
3. Skor test istatistiği.

4.3.4.1. Olabirlik oran (Likelihood ratio) testi

P tane bağımsız değişken içeren regresyon modelinde; oluşturulan iki farklı modelden birinde v tane bağımsız değişken varken diğerinde $p = v + m$ gibi tüm bağımsız değişkenler olsun. İki regresyon modelinde de bağımsız değişkenlerin katsayıları gösteriminde, birinci model için β_v vektörü, ikinci model içinse β_{v+m} vektörü kullanılsın. $X_{v+1}, X_{v+2}, \dots, X_{v+m}$ bağımsız değişkenlerin katsayılarının sıfıra eşit olup olmadığı eşanlı (simültane) olarak test eden benzerlik oran test istatistiği aşağıdaki şekilde ifade edilir.;

$$LR = -2 \left(\ln \left[\frac{l(\beta_v)}{l(\beta_{v+m})} \right] \right) = -2 \left[\ln[l(\beta_v)] - \ln[l(\beta_{v+m})] \right] \quad (\text{Bilgin, 2012})$$

Burada $l(\beta_v)$: v tane bağımsız değişken içeren birinci modelin en çok olabilirlik fonksiyonunu, $l(\beta_{v+m})$: p tane bağımsız değişken içeren ikinci modelin en çok olabilirlik fonksiyonunu gösterir. Olabilirlik oran istatistiği $(v + m) - v = m$ serbestlik dereceli ki-kare dağılım özelliği gösterir.

Olabilirlik Oran Testi Forward Logistik Regresyon model geliştirme yöntemidir, mevcut model ve bağımsız değişkenin dışarıda bırakıldığı modeli karşılaştırmayı sağlar.

4.3.4.2. Wald test istatistiği

Modeldeki logit kat sayıların anlamlılığının test edilmesi için geliştirilmiş bir yöntemdir.

$$Wald = \left(\frac{K}{SE_K} \right)^2$$

K : Regresyon Katsayısı

SE_K : Katsayıların Standart Hatası

Formülde bulunan değer SD=1 olan X^2 (Khi-Kare) dağılışı ile karşılaştırılarak karar verilir. Wald İstatistiği kolay olmasına rağmen güvenilirliği konusunda şüpheler mevcuttur. Küçük örneklem için bu durum daha çok ortaya çıkmaktadır. Katsayıları büyük tahmin eden veriler standart hatayı genelde büyütür. Bu durum daha küçük Wald istatistiği demektir. Bu nedenle açıklayıcı değişkenin yanlışlıkla modelde kalması durumu ortaya çıkar. (Çelik 2015)

Model İleri Doğru (Forward) Lojistik Regresyon model geliştirmeyi sağlar.

4.3.4.3. Skor test istatistiği

İlk model ve geliştirilmiş modelin açıklayıcı değişken ve cevap değişkenleri arasındaki ilişkilerinin modeller açısından farklılıklarının sorgulanmasını sağlayan test istatistiğidir. İlişki değil modeller arasındaki farkın istatistiki anlamlılığını sorgular.

Skor testi log olabilirlik fonksiyonunun türevlerinin dağılım teorisine dayanmaktadır. Log olabilirlik fonksiyonun birinci ve ikinci türevlerinden oluşan matrisden elde edilir. Skor testi p serbestlik dereceli ki-kare dağılımı gösterir. Omnibus test olarak bilinir.

4.3.5. Lojistik regresyonda model uyumu

Karışımli modellerde uygun model seçiminde kullanılan uyum ölçütleri Akaiki Bilgi Ölçütü (AIC), Bayesian bilgi ölçütü (BIC) ve Entropy'dır. AIC ve BIC uygun model seçimi için kullanılırken entropy ölçütü ise bireylerin doğru sınıflandırma oranının belirlemek için kullanılır. Uyum ölçütleri genel olarak;

$$AIC = -\text{Log}L + 2p$$

$$BIC = -\text{Log}L + p\ln(n)$$

şeklinde ifade edilir.

$\text{Log}L$ karışımli modellerde iterasyon bittikten sonra elde edilen Log olabilirlik değeridir. 'p' serbest parametre sayısıdır.

Değişik sınıflar için en küçük AIC ve BIC değerini veren model; uygun modeli (ideal sınıf sayısını) ifade eder. Örnek büyüklüğün yeterli olmaması durumunda BIC uyum ölçütü aşağıdaki gibi hesaplanır.

$$BIC = -2\text{Log}L + p\ln\{(n+2)/24\}$$

Sınıf sayısı belirlendikten sonra bireylerin tüm sınıflarda bulunma olasılığını entropy ölçütü belirler. Entropy ölçütü şu denklem ile hesaplanır.

$$E_c = 1 - \frac{\sum_i \sum_k \left(-\hat{\pi}_{ik} \ln \hat{\pi}_{ik} \right)}{n \ln K}$$

Bu değer $0 \leq E_c \leq 1$ aralığında değişir. Bu değer 1'e yaklaştıkça sınıflandırma olasılığın yüksek olduğu anlaşılır. (Kaya 2007)

5. GEREÇ VE YÖNTEMLER

Araştırma sonuçları WEKA paket programı ile yapılmıştır.

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 günümüz veri analizi uygulamalarının tüm formlarında yaygın olarak kullanılmaktadır.

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.

Weka açılan penceresinde aşağıdaki seçenekleri görmek mümkündür.

Simple CLI: WEKA komutlarının direkt olarak işlenmesine olanak sağlayan basit bir komut satırı ara yüzü sağlar.

Explorer: Verinin WEKA ile keşfi için bir ara yüzüdür. Bu ara yüzde; veriler 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 önışleme, 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 ilişkilerin iki boyutlu grafiklerle izlenebildiği paneldir.

Experimenter: Deneyleerin gerçekleştirilmesi ve öğrenme planları arasındaki istatistiksel testleri yürüten bir ara yüzüdür. 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 sürükleyip bırak ara yüzü ile yerine getirmektedir. Experimenter tarafından desteklenmeyen ek özellikleri ve experimenterde bulunan bazı eksik özellikleri ile gelişmekte olan bir bölümdür. (Kuyucu, 2012)

5.1. Veri seçimi

Weka paketinde yer alan 'explorer' ara yüzü içerisinde veri üretme olanakları mevcuttur. Oluşturulabilen veriler;

Agrawal: Bir sınıf değişkeni ve araç satışına yönelik 10 değişkenin, karışık özellikteki verilerini içeren hazır veriyi içermektedir. Agrawal tarafından mühendislikte kullanılmak üzere hazırlanmıştır.

Bayesnet: 10 nominal değişken içeren ve olasılık hesaplamalarında kullanılacak, rastgele atanmış 'false-true' şeklindeki verileri içermektedir. Sınıf değişkeni bulunmamaktadır.

LED24: 25 binary değişkeni içeren ve bir sınıf değişkeni bulunan hazır veri setidir. Burada oluşturulan sınıflarda 'false-true' verilerinin literal anlamları yerine, 'boolean' olarak ifade edilen 7 matematiksel mantık ilişkilendirilmiştir. Döngüsel programlamalarda kullanılan veri tipidir.

RandomRBF: Her değişken için atanmış rastgele sayılar; bir ağırlık değeri, standart sapma ve merkezi değer ile ilişkilendirilmiştir. Tüm vektör uzunluklarının Gauss dağılımından rastgele eşit ağırlıkla ölçeklenmesi ile oluşturulmuştur. Merkezi değer sınıf aralıkları dikkate alınarak belirlenmiştir. Bir sınıflayıcı ve 10 sayısal özellikte değişkenden oluşmaktadır.

RDG1:Rastgele atanmış tanımlayıcı veri setidir. Ancak bu rastgele atama tüm sınıfların bire bir kendi aralarında ilişkilendirilip, tüm sınıf kurallarını içerecek türden oylama ile belirlenen yeniden sınıflama işlemine tabi olarak oluşturulmuştur. Bir sınıflayıcı ve 10 nominal değerden oluşmaktadır.

Araştırmaya konu olan Yapay Sinir Ağları, Destek Vektör Makineleri ve Lojistik Regresyon analizi yöntemleri ile doğru sınıflama başarısı ölçüldüğü için, sınıf değişkeni olmayan Bayesian ve LED24 veri seti kullanılmamıştır. Aslında; yöntemler, veri kümesinde sınıf değişkeni olmadan da, yeni sınıf değişkeni oluşturarak verileri atamada oldukça başarılıdır. Ancak böyle bir çalışma için benzer bir çalışmanın verileri sisteme önceden tanıtılır, sonrasında esas çalışmanın gerçek verileri kullanıldığında verilerin korelasyonu saptanarak veya olasılıkları hesaplanarak yeni sınıflar oluşturulur. Çalışmamızda; yöntemlerin üstünlüğü belirlenirken, gerçek bir veri kümesi içeren bilimsel sonuçlara ulaşmak ya da sınıflama yapmak (örneğin; Myokard Enfarktüsünün etkilerini araştırmak) gibi bir amaç hedeflenmediğinden dolayı yöntemlerin bu özelliği üzerinde durulmamıştır.

Agrawal ve RDG veri setinde mevcut nominal değişkenler arasındaki ilişki, rastgele üretilen yeni verilerde kurulamayacağı için kullanılamamıştır.

RandomRBF algoritması, sınırlayıcılara bağlı kalarak en iyi veri üretme başarısını gösterdiği için çalışmada bu hazır veri seti ile çalışılmıştır. Cevap değişkenin normal dağıldığı ve normal dağılmadığı, $n=75$, $n=150$ şeklinde veri içeren ve yanlılık (bias) değeri eklenerek üretilen veri setleri ile çalışıldı.

5.2. Veri çoğaltma yöntemleri

Araştırma yöntemlerinin doğru sonuçlar üretebilmesi yöntem gerekliliklerini yerine getirmesine bağlıdır. İstatistik araştırma yöntemleri parametrik ve parametrik olmayan araştırma yöntemleri olarak ayrılırlar. Parametrik testler daha doğru sonuçlar vermesi nedeniyle daha çok tercih edilir. Parametrik test varsayımlarından ilki araştırma setinde $n \geq 30$ olmasıdır. Daha sonrasında ise verilerin normal dağılıma sahip olması verilerin evrenden rastgele seçilmiş olması koşulunun gerçekleşmesi araştırılır.

Bu bağlamda veri sayısının artırılması ile doğruluk değerlerindeki değişim ve verilere yanlılık eklenerek, verilerin homojenitesi bozulduğunda yöntemlerin nasıl

etkilendiğini araştırmak için Filtreleme yapılmıştır. Weka programında denetimli veri üretme (supervised instance) menüsünde, Yeniden Örnekleme (Resample) başlığı altında bu işlemi gerçekleştirmek mümkündür. Şayet bir sınıf değişkeni yoksa çalışmada, veri üretimi denetimsiz veri üretme menüsünden yapılmalıdır(unsupervised).

Resample ile yerine koyarak, rasgele alt veri kümesi oluşturulabilir. Bu üretimin sağlıklı çalışması için sınıf değişkeninin nominal olması gereklidir. Buradaki amaç sınıf verilerinin numerik yada sıralayıcı olarak algılanmasını engellemektir. Sınıf verisinin eşit dağılımını koruyarak verinin üretilmesi gereklidir. Sınıf değişkenine filtreleme seçeneklerinden ayrıklaştırma (discretize) uygulanmış, sınıflara etiket atanmıştır. Sınıflar arası dengesizlik ‘balance’ seçeneği ile düzenlenerek, sınıflar dengelenmiştir.

Yerine koyarak örnekleme; Örneklenen verilerin, verinin seçildiği havuza tekrar konularak, takip eden seçimler de yeniden örneğe alınabilmesi durumudur. Bu yöntemde, “farklı” veya “seçilmeyen” birim tanımı önemini kaybetmektedir. Örneklenen birimlere birden fazla sayıda seçilme şansının verilmesidir. Bu örnekleme yöntemi permütasyon olarak da adlandırılmaktadır. n birimin geliş sırası dikkate alınarak birbirinden farklı r kadar düzenlemenin elde edilmiş sayısı aşağıdaki gibidir.

$$nP_r = \frac{n!}{(n-r)!}$$

Yerine koymaksızın örnekleme; “farklı” ve “seçilmeyen” birim tanımı ile daha önce örneğe alınmış olan birimlerin seçim aşamasında dikkate alınmaması ve daha sonraki seçimlere bu birimlere seçim olasılığının verilmemesidir. Bu örnekleme yöntemi kombinasyon olarak da adlandırılmaktadır. n birimin geliş sırası dikkate alınmadan birbirinden r farklı düzenlemenin elde edilmiş sayısı:

$$nC_r = \frac{n!}{(n-r)!r!}$$

(Kökçü, 2014)

Bu çalışmada yerine koyarak örnekleme yöntemi kullanılmıştır.

5.3. Araştırma yöntemlerini karşılaştırmada kullanılacak ölçütler

Weka programında var olan sınıflayıcı (classify) ara yüzü ile fonksiyon kullanarak sınıflama yapan 'function' seçeneği; Yapay Sinir Ağları (Multilayer Perceptron), Destek Vektör Makinesi (SMO, Lib SVM), Lojistik Regresyon (Simple Logistic Regression, Kernel Logistic Regression) yöntemlerine ulaşmamızı sağlar.

Yöntemlerin sınıflama başarısını değerlendirmek için aşağıdaki ölçütlerden yararlanılacaktır.

- Doğru sınıflama değeri
- Kappa değeri
- Ortalama mutlak hata
- Ortalama karesel hata
- Göreceli mutlak hata
- Göreceli karesel mutlak hata
- Düzensizlik matrisi

Yöntemlerin karşılaştırılmasında baz alınacak ölçütler:

- Kesinlik
- Duyarlılık
- F ölçütü
- AUC (ROC)
- Kappa değerleridir.

Kesinlik(Accuracy): Sınıfı 1 olarak tahminlenmiş doğru pozitif örnek sayısının, sınıfı 1 olarak tahminlenmiş tüm örnek sayısına oranıdır.

$$Kesinlik = \frac{TP}{TP + FP} \quad (\text{Ferrari, 2014})$$

Duyarlılık (Sensitivity): Doğru sınıflanmış pozitif örnek sayısının, gerçek pozitif örnek sayısına oranıdır.

$$Duyarlılık = \frac{TP}{TP + FN} \quad (\text{Dorrius, 2014})$$

F ölçütü: Uygulanan yöntemle elde edilen kesinlik ve duyarlılık oranlarını birlikte değerlendirilmesine olanak veren, her iki ölçütün harmonik ortalama değeridir.

$$F \text{ Ölçütü} = \frac{2x \text{ Duyarlılık } x \text{ Kesinlik}}{\text{Duyarlılık} + \text{Kesinlik}} \text{ (Coşkun ve Baykal, 2011)}$$

AUC (ROC):Tanı testinin duyarlılık ve özgüllük yeteneğini belirlemek için, çalışmaya ait optimum kesim noktalarını kullanarak oluşturulan grafiksel gösterim yöntemidir. Bu yöntemde, eğrinin altında kalan alan yöntemin sınıflamadaki doğruluk oranını belirler. ROC eğrisinin altında kalan alan 0 ve 1 arasında değer alır. (Mickey, 2014)

Kappa: İki araştırmacının aynı çalışmaya ait görüşlerinin birbiri ile uyumunun ölçüldüğü bir yöntemdir. Amaç tahmin doğruluğunu değerlendirmektir. Kappa kat sayısı 0 ve 1 arasında değer alır. Buna göre:

0.93-1 arası: mükemmel

0.81-0.92: çok iyi

0.61-0.80: iyi

0.41-0.60: orta düzeyde

0.21-0.40: ortanın altında

0.01-0.20: zayıf uyumu tanımlar. (Keskin, 2004)

Araştırmaya konu olan veriler Ek 1’de belirtilmiştir.

5.4. Veriler ve Özellikleri

Tablo 1’de birinci sütunda “Bağımsız Değişkenler ; a₁, a₂... şeklinde gösterilmiştir. “Sınıf Ağırlığı Dengelenmemiş” sütununda gerçek veriler yer almaktadır. Ve sınıf üye sayıları yani “n” sayısının eşit değildir. “Sınıf Ağırlıkları Dengelenmiş” sütununda sınıflara ait “n” sayıları eşitlenmiştir. Amaç sınıf sayılarında dengesizlik olduğu durumda sınıflama yeteneği bakımından modellerdeki değişimi gözlenmektir. Son sütunda ise sınıf üye sayılarına 0.10 ve 0.20 düzeyinde dengesizlik uygulanmış ve değişken bazında değişiklikler tabloda yansıtılmıştır. Standart sapmadaki değişimlerin modellerdeki sınıflama yeteneğine etkisi gözlenmek istenmiştir.

Tablo 1: Sınıf Değişkenlerinin Sınıf Sayısı ve Bias'a (Yanlılık) Bağlı, Standart Sapma Ve Ortalamaları Değişim Tablosu

		Sınıf Ağırlığı Dengelememi	Sınıf Ağırlıkları Dengelemiş		Sınıf Yanlılığı Uygulanmış	
		Bias=0	Bias=0	Bias=0	Bias=0.10	Bias=0.20
Bağımsız değişken	Sınıf Sayısı	n=75.00 C ₀ =27.00 C ₁ =48.00	n=75.00 C ₀ =27.00 C ₁ =48.00	n=150.0 C ₀ =54.00 C ₁ =96.00	n=150.0 C ₀ =56.00 C ₁ =93.00	n=150.0 C ₀ =58.00 C ₁ =91.00
	Sınıf Ağırlıkları	C₀=27.00 C₁=48.00	C₀=37.50 C₁=37.50	C₀=75.00 C₂=75.00	C₀=77.70 C₁=72.65	C₀=80.56 C₁=71.09
a ₀ Değişkeni	\bar{X}	0.391	0.361	0.343	0.344	0.344
	S	0.355	0.362	0.372	0.372	0.370
a ₁ Değişkeni	\bar{X}	0.490	0.464	0.471	0.464	0.456
	S	0.427	0.421	0.439	0.433	0.433
a ₂ Değişkeni	\bar{X}	0.464	0.467	0.412	0.415	0.419
	S	0.311	0.315	0.310	0.311	0.312
a ₃ Değişkeni	\bar{X}	0.494	0.493	0.496	0.491	0.485
	S	0.351	0.359	0.353	0.351	0.354
a ₄ Değişkeni	\bar{X}	0.563	0.589	0.553	0.562	0.569
	S	0.301	0.303	0.351	0.345	0.344
a ₅ Değişkeni	\bar{X}	0.549	0.566	0.602	0.595	0.594
	S	0.402	0.396	0.495	0.481	0.478
a ₆ Değişkeni	\bar{X}	0.535	0.538	0.570	0.566	0.566
	S	0.333	0.348	0.327	0.328	0.328
a ₇ Değişkeni	\bar{X}	0.437	0.425	0.485	0.479	0.473
	S	0.350	0.351	0.363	0.361	0.359
a ₈ Değişkeni	\bar{X}	0.483	0.472	0.536	0.534	0.527
	S	0.347	0.357	0.350	0.355	0.358
a ₉ Değişkeni	\bar{X}	0.501	0.482	0.542	0.531	0.530
	S	0.373	0.370	0.419	0.411	0.409

Veriler normal dağılıma sahiptir.

Tablo 2:Normal Dağılım Tablosu

Sınıf		Shapiro-Wilk		
		Statistic	Df	Sig.
a0	c0	0.972	27.00	0.655
	c1	0.963	48.00	0.128
a1	c0	0.982	27.00	0.914
	c1	0.938	48.00	0.014
a2	c0	0.988	27.00	0.986
	c1	0.983	48.00	0.694
a3	c0	0.960	27.00	0.360
	c1	0.969	48.00	0.228
a4	c0	0.976	27.00	0.753
	c1	0.976	48.00	0.415
a5	c0	0.980	27.00	0.864
	c1	0.934	48.00	0.010
a6	c0	0.944	27.00	0.155
	c1	0.954	48.00	0.060
a7	c0	0.956	27.00	0.301
	c1	0.972	48.00	0.303
a8	c0	0.961	27.00	0.383
	c1	0.948	48.00	0.033
a9	c0	0.983	27.00	0.926
	c1	0.961	48.00	0.108

Shapiro-Wilk Test istatistiği değerlerinin '0' ve '1' aralığında olması verinin normal dağılıma sahip olduğunu gösterir. (Rozali, 2011)

Veriler arasında oto-korelasyon olmadığı test edilmiştir.

Tablo 3: Oto-korelasyon Tablosu

Değişken	Durbin Watson 'd' Test Değeri
a0	1.796
a1	2.045
a2	1.812
a3	1.882
a4	1.943
a5	1.966
a6	1.878
a7	2.203
a8	2.245
a9	2.306

Durbin Watson 'd' test değerinin $1.5 < d < 2.5$ aralığında değer aldığı otokorelasyon yoktur denilir. (Çelik, 2015)

6.BULGULAR

6.1.Yapay Sinir Ağları Sınıflama Sonuçları

WEKA veri programında, Çok Katmanlı Geri Yayımlı Sinir Ağ modelinde (Multilayer Perceptron) bazı fonksiyonlar çalışma verimini artırmak için kullanılmıştır. Bu seçeneklerin özellikle büyük veri kümelerinde kullanılması çalışmada yarar sağlamaktadır.

GUI: Ağın görsel sunumuna izin verir.

DECAY: Ağ çalışırken, öğrenme oranının zaman içerisinde azalmasını sağlar ve uzun veri çalışmalarında avantaj oluşturur. Bu işleme ‘Çürüme Seçeneği’ denir. Çalışmamızda veri kümesinin az olması nedeniyle ‘Çürüme’ kullanılmamıştır.

Learning Rate(Öğrenme Oranı): 0.3 (Overestimate oluşmaması için önerilen değer.)

Momentum (Öğrenme Hızı): 0.2 (Ağ eğitimini “sıfır” öğrenme seviyesine kadar sürdürür. Bu değer ağın “sıfır”a yaklaşırken kendini tamamlamadaki süresini ciddi anlamda kısaltır.)

Training Time: EPOCH’u yani öğrenme adımını (döngü sayısını) belirlemeyi sağlar.. Döngü sayısını belirleyerek, nöron ve sinapsislerin değer güncellemesini sağlar. Ağın sınıflama belirleyiciliğini artırmak için ‘Öğrenme Tekrarı’ (Epoch)’u 500 olarak seçtik. Sınıf değişkenine sahip olmayan, yeni ve gerçek veri çalışıldığında, sınıflama yapabilmek için, ağın önden eğitilmesi gerekmektedir. Böylece yeni veri setinde sınıflama, tanımlana sınıfların özelliklerine bağlı kalarak oluşturulur.

Hidden Layer: Gizli katman sayısıdır.

Modelde ‘Gizli Katman Sayısı’ belirlemek için araştırmacıya üç seçenek sunmaktadır. Bu seçenekler aşağıdaki gibidir:

Gizli Katman Sayısı ‘i’, ‘o’ ve ‘a’ olarak seçilebilir:

$i = \text{Özellik(Değişken)Sayısı}$

$o = \text{SınıfSayısı}$

$a = \frac{\text{ÖzellikSayısı} + \text{SınıfSayısı}}{2}$

Çalışmamızda her üç gizli katman sayısının, sınıflama doğruluk oranı üzerindeki performansları denenmiştir.

WEKA veri kümesini parçalara bölerek veya yüzde oranlar şeklinde ayırarak kontrol verisi olarak kullanır ve sınıflama doğruluk düzeyini gözlemler. Bu bölünme şeklini araştırmacının belirlemesini bekler. Verilerin onda biri kontrol verisi olarak kullanıldığında, doğruluk oranında daha yüksek değerler elde ettik.

Tablo 4: Çok Katmanlı Sinir Ağı Doğru Sınıflama Tablosu (Multilayer Perseptron)

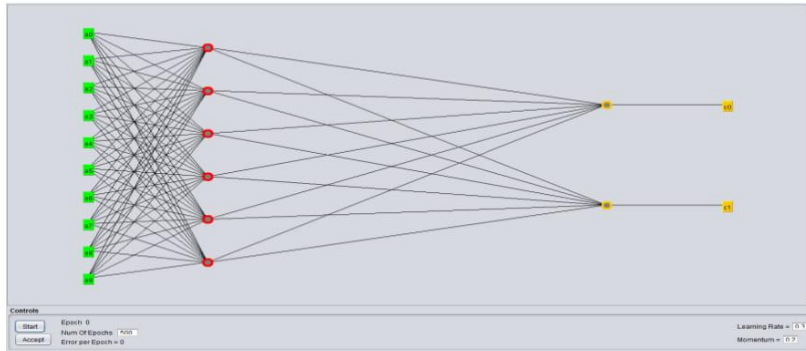
VERİ	Katma Sayısı=2 HiddenLayer (o)	Katman Sayısı=6 Hidden Layer (a)	Katman Sayısı=10 Hidden Layer(i)
Dengesiz Veri n=75.00 C ₁ = 48.00 Co=27.00 Ağırlık: C ₁ = 48.00 Co=27.00	0.707	0.720	0.707
Dengelenmiş Veri n=75.00 C ₁ = 48.00 Co=27.00 Ağırlık: C ₁ =37.50 Co=37.50	0.642	0.731	0.777
Çoğaltılmış veri n=150.0 C ₁ = 96.00 Co=54.00 Ağırlık: C ₁ = 75.00 Co=75.00 BiAS=0.000	0.921	0.966	0.958
Çoğaltılmış Veri n=149.0 C ₁ = 93.00 Co=56.00 Ağırlık: C ₁ = 72.65 Co=77.77 BiAS=0.100	0.877	0.967	0.967
Çoğaltılmış Veri n=149.0 C ₁ = 91.00 Co=58.00 Ağırlık: C ₁ = 71.09 Co=80.56 BiAS=0.200	0.943	0.970	0.977
% Oranı			

Sınıfların ağırlıkları eşitlendiğin de başarının arttığı görülmektedir. Bu başarı yöntemin sınıflamada vektör uzunluğunu baz almasından kaynaklanmaktadır. Eşit sınıf ağırlığı durumunda vektör uzunluk farklılıkları daha kolay ayrıştırılır. Ancak bu başarı, gizli katman sayısının iki olarak sınırlandırıldığı dizaynda ve sınıflar eşit ağırlandırıldığında negatif etkilenmektedir. Buna bağlı olarak, bu veri kümesi için, çalışmada iki gizli katmanın bulunması yeterli değildir. Gizli katman sayısı arttıkça doğruluk oranında artış gözlenmektedir. Veri sayısının artması ile başarı yüzdesi,

tüm dizaynlarda %70'den %90'lara yükselmiştir. Diğer yöntemlerin (SVO ve Lojistik Regresyon) aksine, standart hata yükseldikçe başarı oranının düştüğü gözlenmiştir. Yine bu üç yöntemi kendi aralarında kıyasladığımızda, sınıf sayısının (n) artmasından pozitif yönde en çok etkilenen YSA yöntemidir. Yöntemin en başarılı dizaynı olan. Dengelenmiş sınıf ağırlığı, n=150 ve bias=0 olduğu durumda en başarılı dizaynı gizli katman sayısının altı seçildiği çalışmadır. Dizayna ait sınıflama değerleri aşağıdadır.

Tablo 5.Yapay Sinir Ağları En İyi Modelin Sınıflama Tablosu

Çok Katmanlı Yapay Sinir Ağı	Değerler
Gizli Katman Sayısı =6	
Doğru Sınıflama	0.966
Ortalama Mutlak Hata	0.052
Ortalama Karesel Hata	0.176
Göreceli Mutlak Hata	10.34
Göreceli Karesel Hata	35.08
Kesinlik	0.966
Duyarlılık	0.966
F Ölçütü	0.966
ROC	0.972
Kappa	0.932
Genel Doğruluk	0.966



Şekil 15:YSA modelinin görsel sunumu

Tablo 6: YSA Modeli Karışıklık (Confusion) Matrisi

a=C ₀	b=C ₁
72.22	2.780
2.340	72.66
Toplam Doğru	144.9

6.2. Destek Vektör Makineleri Sınıflama Sonuçları

Destek Vektör Makineleri (SVM) doğrusal denklem sistemi (Conjugate Gradient) kullanan ve veriyi bir bütün olarak inceleyen algoritmaya sahiptir. SVM veriyi doğrusal olarak düşünür ve bundan ötürü iki parça şeklinde ayırarak, her bir parça için optimal Lagrange Çarpanını bulmaya çalışır. Ancak bu işlem uzun sürer ve kullanılan Dörtlü Programlama Algoritması (QP), karar fonksiyonu gibi değil, kendi matematiksel performansını ölçmeye çalışan bir içsel döngüye döner. (Doreswamy, 2013)

SVM'nin, Ardışık Minimal Optimizasyon (SMO) algoritmasında, veriyi 'sparse' denilen parçalara bölerek, bu parçaları adım adım inceler, QP'yi bir karar fonksiyonu gibi kullanarak her adım için ayrı ayrı Lagrange çarpanları bulur ve son olarak bulunduğu Lagrange çarpanını SVM için en uygun Lagrange çarpanı şeklinde günceller. (Platt, 1998)

WEKA SVM sınıflama için **SMO algoritması** kullanır. SMO sınıflayıcı penceresinde başka bir sınıflayıcının arka planda çalışmasını sağlayarak, en uygun olasılık tahminine erişebilmek için, SMO'nun ulaştığı değerleri kalibre etmesine olanak sağlar. Bu ikinci sınıflayıcının hangisi olacağına araştırıcı karar verir. SMO yönteminde 'C' parametresi gerekli ise kullanılır. 'C' parametresi bir çeşit 'eşik' değer gibi kullanılır.

String Kernel: Tüm değişkenleri SSK (Subsequence Kernel) denen alt dizinler olarak kabul eder ve uzunluklarına bağlı olasılık hesabı yapar. Son olarak sınıflara Lambda uygulayarak budama gerçekleştirir. Fonksiyonlar aşağıdaki şekillerde kullanılmıştır.

- RBF: Kernel (Radial Basis Function).

$$K(x, y) = e^{-\gamma \langle x, y \rangle} \quad (\text{Platt, 1998- Keerthi, 2001})$$

- PolyKernel= Polinom derecesi arttıkça algoritmanın karışık bir hal alması nedeniyle, büyük veri setlerinde tercih edilmez.

$$K(x, y) = (\langle x, y \rangle + 1)^p$$

- Normalized Polikernel: Polykernel fonksiyon değerlerinin karesel hesaplamasıdır.

- Precomputed Kernel: Matriks Kernel olarak da ifade edilebilir. Nominal sınıf deęişkenine nominal verilerin atamasını yapar. (Lodhi v.d. 2002)
- PUK (Pearson VII function): Kernel fonksiyonundaki gauss olasılık deęerine Lorent dönüşüm uyarlayarak 'v' zaman birimini '-v' yönde olumsuzlar. Sınıflamada üstüste çakışık verileri ayırıştırarak sınıf ataması yapar.

WEKA'nın bir başka SVM algoritması ise **LibSVM algoritmasıdır**. SVM sınıflayıcının en önemli kütüphanesidir. (WEKA içerisindeki her bir yöntemi kütüphane diye adlandırır.) LibSVM sarmalayıcı sınıf modelidir. WEKA'daki diğer SVM modelinden farklı olarak hedef deęişkeni normalize ederek analizi yapar. Kernel fonksiyonları kütüphanede aşağıdaki şekilde yer alır.

- $RB F = \exp.(-\text{gamma} * |u - v|^2)$
- $Linear = (u * v)$
- $Polinomial = (\text{gamma} * u * v + \text{coeff})^{\text{deg ree}}$
- $Sigmoid : \tanh(\text{gamma} * u * v + \text{coeff}0)$

6.2.1. Ardışık minimal optimizasyon (SMO) fonksiyonlarına göre DVM sınıflama başarı dağılımı

Tablo 7: Ardışık Minimal Optimizasyon (SMO) Fonksiyonlarına Göre DVM Sınıflama Tablosu

Veri	PUK	Polikernel	RBF	Normalize Polikernel
Dengesiz Veri n=75.00 C ₁ = 48.00 Co=27.00 Ağırlık: C ₁ = 48.00 Co=27.00	0.800	0.627	0.640	0.640
Dengelenmiş Veri n=75.00 C ₁ = 48.00 Co=27.00 Ağırlık: C ₁ =37.50 Co=37.50	0.826	0.604	0.455	0.646
Çoğaltılmış veri n=150.0 C ₁ = 96.00 Co=54.00 Ağırlık: C ₁ = 75.00 Co=75.00 BiAS=0.000	0.976	0.689	0.481	0.687
Çoğaltılmış Veri n=149.0 C ₁ = 93.00 Co=56.00 Ağırlık: C ₁ = 72.65 Co=77.77 BiAS=0.100	0.976	0.713	0.517	0.711
Çoğaltılmış Veri n=149.0 C ₁ = 91.00 Co=58.00 Ağırlık: C ₁ = 71.09 Co=80.56 BiAS=0.200	1.000	0.730	0.531	0.707

SMO algoritması ile yapılan sınıflamada, PUK fonksiyonu diğer fonksiyonlara kıyasla çok büyük başarı (%97) göstermiştir. PUK fonksiyon hesaplamada ‘ σ ’ ve ‘ ω ’ kullandığı için SVM’de başarı oranı yüksektir. (Kavzoğlu, 2010)

6.2.2.Libsvm fonksiyonlara göre sınıflama başarı oranı

Tablo 8: Libsvm Fonksiyonlara Göre Sınıflama Tablosu

Veri	RBF	Linear (u*v)	Polinomial	Sigmoid
Dengesiz Veri n=75.00 C ₁ = 48.00 Co=27.00 Ağırlık: C ₁ = 48.00 Co=27.00	0.640	0.680	0.640	0.640
Dengelenmiş Veri n=75.00 C ₁ = 48.00 Co=27.00 Ağırlık: C ₁ =37.50 Co=37.50	0.500	0.620	0.500	0.500
Çoğaltılmış veri n=150.0 C ₁ = 96.00 Co=54.00 Ağırlık: C ₁ = 75.00 Co=75.00 BiAS=0.000	0.550	0.685	0.500	0.500
Çoğaltılmış Veri n=149.0 C ₁ = 93.00 Co=56.00 Ağırlık: C ₁ = 72.65 Co=77.77 BiAS=0.100	0.589	0.707	0.483	0.483
Çoğaltılmış Veri n=149.0 C ₁ = 91.00 Co=58.00 Ağırlık: C ₁ = 71.09 Co=80.56 BiAS=0.200	0.640	0.724	0.469	0.469
Toplam				

LibSVM algoritması ile yapılan sınıflamada Linear (u*v) fonksiyonu başarılı olmuş (% 72) ancak genel olarak algoritmanın sınıflama başarısı oldukça düşüktür. Bu durum seçtiğimiz veri seti ile ilgilidir. LibSVM sınıflayıcı tüm verilerin nominal özellikte olduğu veri setlerinde çok başarılı (%93) sınıflama yeteneğine sahiptir.

SVM sınıf ağırlıkları dengelenmemiş veri setini sınıflamada, dengelenmiş veri setinde çalıştığından daha başarılı olmuştur. SVM'de standart sapma arttıkça başarı oranı yükselmiştir. Bunun sebebi şöyle açıklanabilir; standart hatanın büyümesi marjinin büyümesi demektir ve sınıflar arasındaki fark daha kolay saptanır.

Tablo 9: Destek Vektör Makinesi En İyi Modelin Sınıflama Başarı Değerleri

Ardışık Minimal Optimizasyon (SMO) (PUK fonksiyon)	Değerler %
Doğru Sınıflama	0.976
Ortalama Mutlak Hata	0.023
Ortalama Karesel Hata	0.154
Göreceli Mutlak Hata	4.742
Göreceli Karesel Hata	30.79
Kesinlik	0.977
Duyarlılık	0.976
F Ölçütü	0.976
ROC	0.953
Kappa	0.976
Genel Doğruluk	0.976

Tablo 10: Destek Vektör Makinesi Karışıklık (Confusion) Matrisi

a=C ₀	b=C ₁
72.22	2.780
0.780	74.22
Toplam Doğru	146.4

6.3. Logistic Regresyon Modeli Sınıflama Sonuçları

WEKA programı Logistik Regresyon konusunda pek çok alternatif sunar bu nedenle veri türüne göre doğru sınıflamaya karar vermek önemlidir. Multinomial Regresyon Modeli tüm verilerin nominal olduğu ve sınıf değişkenin çoklu olduğu veri setlerinde kullanılmalıdır. Verilerin doğrusal olmaması nedeniyle Linear Lojistik Regresyon modeli kullanılmamıştır. Araştırmaya Lojistik Regresyon, Basit Lojistik Regresyon (Simple Logistic Regression) ve Kernel Lojistik Regresyon modelleri eklenmiştir. Üç yöntemin seçilmesinin nedeni, sınıf değişkeninin binary olmasıdır.

6.3.1. Lojistik regresyon sınıflama başarı oranı

Tablo 11: Lojistik Regresyon Sınıflama Başarı Değerleri

Veri	Başarı %
Dengesiz Veri n=75.00 C ₁ = 48.00 Co=27.00 Ağırlık: C ₁ = 48.00 Co=27.00	0.680
Dengelenmiş Veri n=75.00 C ₁ = 48.00 Co=27.00 Ağırlık: C ₁ =37.50 Co=37.50	0.569
Çoğaltılmış veri n=150.0 C ₁ = 96.00 Co=54.00 Ağırlık: C ₁ = 75.00 Co=75.00 BiAS=0.000	0.717
Çoğaltılmış Veri n=149.0 C ₁ = 93.00 Co=56.00 Ağırlık: C ₁ = 72.65 Co=77.77 BiAS=0.100	0.698
Çoğaltılmış Veri n=149.0 C ₁ = 91.00 Co=58.00 Ağırlık: C ₁ = 71.09 Co=80.56 BiAS=0.200	0.693
Toplam	

Genel başarı oranı düşüktür. WEKA'da Lojistik Regresyon yöntemi Ridge regresyon modeli şeklinde kurulmuştur. Eğer 'k' sınıf, 'n' örnek ve 'm' değişken varsa β matris parametresi oluşturur ve $m*(k-1)$ matris şeklinde hesaplama yapar. Regresyon değerleri direk alınmayıp, olasılık değerleri üzerinden hesaplama

yapılır. 'sed Ridge' seçeneği Ridge değerini belirlememizi sağlar. Ridge değeri matrisin köşegenlerine bir küçük pozitif sayı eklenerek, kestirim varyansı küçültür. Ancak bu değer görecelidir. (Le Cessie, 1992)

Tablo 12: Lojistik Regresyon Kat Sayıları

Değişken	Beta	Ods oranları	Walt Test	Sig.
a0	-1.985	0.137	3.475	0.062
a1	-0.723	0.485	0.555	0.456
a2	0.182	1.199	0.923	0.337
a3	0.772	2.163	1.347	0.246
a4	4.091	59.77	23.54	0.000
a5	0.511	1.667	2.779	0.096
a6	1.707	5.510	6999	0.008
a7	0.542	1.719	0.029	0.866
a8	0.376	1.457	0.896	0.344
a9	-1.185	0.306	1.003	0.317
Toplam	-2.699		9.969	0.002

Değişkenlerin sınıflama üzerindeki etkisi ve anlamlılıkları bu tabloda yer almıştır.

a0, a4, a6 değişkenlerinin sınıflama üzerindeki belirleyicilikleri diğer değişkenlere kıyasla daha fazladır ve istatistiki olarak odds oranları arasındaki fark anlamlıdır.

6.3.2. Simple lojistik regresyon sınıflama başarı oranı

Tablo 13: Simple Lojistik Regresyon Sınıflama Başarı Değerleri

Veri	Başarı %
Dengesiz Veri n=75.00 C ₁ = 48.00 Co=27.00 Ağırlık: C ₁ = 48.00 Co=27.00	0.733
Dengelenmiş Veri n=75.00 C ₁ = 48.00 Co=27.00 Ağırlık: C ₁ =37.50 Co=37.50	0.635
Çoğaltılmış veri n=150.0 C ₁ = 96.00 Co=54.00 Ağırlık: C ₁ = 75.00 Co=75.00 BiAS=0.000	0.720
Çoğaltılmış Veri n=149.0 C ₁ = 93.00 Co=56.00 Ağırlık: C ₁ = 72.65 Co=77.77 BiAS=0.100	0.684
Çoğaltılmış Veri n=149.0 C ₁ = 91.00 Co=58.00 Ağırlık: C ₁ = 71.09 Co=80.56 BiAS=0.200	0.683
Toplam	

Genel başarı düşüktür. Bu yöntem bir açıklayıcının tek bir yanıt değişkeni üzerindeki etkisini açıklarken daha başarılı sonuçlar verir.

Doğrusal Lojistik Regresyon modeli şeklinde kurulmuştur. Basit regresyon fonksiyonu olan LogitBoost kullanır. Gerçekleşen optimum LogitBoost yinelemelerinin sayısı, otomatik öznelik seçimini sağlar. (Landwehr 2005)

Yöntem çalıştırılırken açılan pencereden, maksimum iterasyon sayısı 500 olarak sunulur. Küçük veri kümelerinde daha düşük iterasyonlar seçilmelidir. Bu pencerede 'heuristik stop' seçeneği, çapraz doğrulama ile minimum bir değer elde edildiğinde çalışmanın durmasını sağlar aksi takdirde çalışma uzun zaman alır. Önemli bir diğer seçenek olan 'W.Betadrim' ile bir önceki sınıfın Beta (β) değeri, $1 - \beta$ şeklinde sonraki sınıfa aktarılacak düzeltme yapılmasını sağlar.

6.3.3. Kernel lojistik regresyon sınıflama başarı oranı

Tablo 14: Kernel Lojistik Regresyon Sınıflama Başarısının Fonksiyon Bazında Değerlendirilmesi

Veri	Normalized polikernel F	PUK Funtion	RadialBasis Function	PoliKernel Function
Dengesiz Veri n=75.00 C ₁ = 48.00 Co=27.00 Ağırlık: C ₁ = 48.00 Co=27.00	0.787	0.720	0.787	0.680
Dengelenmiş Veri n=75.00 C ₁ = 48.00 Co=27.00 Ağırlık: C ₁ =37.50 Co=37.50	0.736	0.611	0.744	0.628
Çoğaltılmış veri n=150.0 C ₁ = 96.00 Co=54.00 Ağırlık: C ₁ = 75.00 Co=75.00 BiAS=0.000	0.980	0.935	0.859	0.704
Çoğaltılmış Veri n=149.0 C ₁ = 93.00 Co=56.00 Ağırlık: C ₁ = 72.65 Co=77.77 BiAS=0.100	0.966	0.945	0.868	0.679
Çoğaltılmış Veri n=149.0 C ₁ = 91.00 Co=58.00 Ağırlık: C ₁ = 71.09 Co=80.56 BiAS=0.200	0.995	0.963	0.860	0.695
Toplam				

Kernel Logisitik Regresyon’da kullanılan String ve Precompute kernel fonksiyonları, nominal açıklayıcı değişkenler için geliştirildiğinden burada kullanılmamıştır.

Bu yöntemde, Negatif log-likelihood olasılık değerleri ikinci derece ceza çarpanı uygulanarak, indirgenmeye çalışılır. Bunun için BFGS optimizasyonu kullanır. Bu yöntemde açıklayıcı x değerlerinin, değişken bazında değil, her birinin bağımsız etkilerinin ölçülmesi sağlanır. WEKA’da araştırmacının Lambda (hata) değerini ataması beklenir (0.01 ve 0.05 aralığında).

Lojistik Regresyon yöntemi standart sapmadan en çok etkilenen yöntem olmuştur. Standart sapma arttıkça başarı oranında artma gözlenmiştir. Ancak Normalized polikernel fonksiyonda n sayısının artmasıyla başarı oranında artış

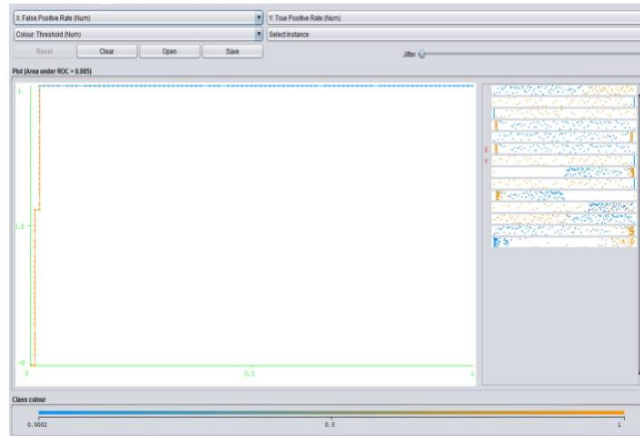
olduğu gözlenmiştir. Standart sapmanın yüksek olması ve 'n' sayısının yüksek oluşu sınıflar arasındaki farklılığın ve dolayısı ile oddsun netleşmesini sağlar.

Tablo 15: Kernel Lojistik Regresyon Yönteminde En İyi Modelin (Fonksiyon türüne bağlı) Sınıflama Başarı Değerleri

Kernel Lojistik Regresyon (Normalized Polikernel Fonksiyonu)	Değerler %
Doğru Sınıflama	0.980
Ortalama Mutlak Hata	0.081
Ortalama Karesel Hata	0.154
Göreceli Mutlak Hata	16.26
Göreceli Karesel Hata	30.78
Kesinlik	0.980
Duyarlılık	0.980
F Ölçütü	0.980
ROC	0.985
Kappa	0.960
Genel Doğruluk	0.980

Tablo 16:Kernel Lojistik Regresyon (Normalized Polikernel) Karışıklık (Confusion) Matriksi

a=C ₀	b=C ₁
73.61	1.390
1.560	73.44
Toplam Doğru	147.1



Şekil 16:Kernel Lojistik Regresyon (Normalized Polikernel) ROC analizinin gösterimi

6.4. Yapay Sinir Ağları, Destek Vektör Makinesi Ve Kernel Lojistik Regresyon Sınıflama Başarılarının Karşılaştırılması

Her üç yöntemin en başarılı algoritmalarını aşağıdaki tabloda karşılaştırabiliriz.

Tablo 17: Yapay Sinir Ağları, Destek Vektör Makinesi Ve Lojistik Regresyon Sınıflama Başarılarının Karşılaştırılması

	Çok Katmanlı Yapay Sinir Ağı Gizli Katman Sayısı =6	Destek Vektör Makineleri (SMO-PUK fonksiyonu)	Kernel Lojistik Regresyon (Normalized Polikernel Fonksiyonu)
Doğru Sınıflama	0.966	0.976	0.980
Ortalama Mutlak Hata	0.052	0.023	0.081
Ortalama Karesel Hata	0.176	0.154	0.154
Göreceli Mutlak Hata	10.34	4.742	16.26
Göreceli Karesel Hata	35.08	30.79	30.78
Kesinlik	0.966	0.977	0.980
Duyarlılık	0.966	0.976	0.980
F Ölçütü	0.966	0.976	0.980
ROC	0.972	0.953	0.985
Kappa	0.932	0.976	0.960
Genel Doğruluk	0.966	0.976	0.980

Yukarıdaki tabloda her üç yöntem için doğru model oluşturulduğunda, üç yöntemin de çok iyi sınıflayıcı olma özelliğini taşıdıklarını görmekteyiz.

Tablo 18: Yöntemlere İlişkin Karışıklık Matrisi

Yapay Sinir Ağları		Destek Vektör Makineleri		Kernel Lojistik Regresyon	
C ₀	C ₁	C ₀	C ₁	C ₀	C ₁
72.22	2.780	72.22	2.780	73.61	1.390
2.340	72.66	0.780	74.22	1.560	73.44
Toplam	144.9	Toplam	146.4	Toplam	147.0
Doğru		Doğru		Doğru	

Doğru sınıflama yüzdesi her üç yöntemde de benzer başarı göstermiştir.

7.TARTIŞMA

Yapay Sinir Ağları ile çalışırken en önemli etken doğru katman sayısının belirlenmesidir (Tablo 4). Simple Lojistik Regresyon tek bir bağımsız değişken ile bağımlı değişkenin tanımlanacağı doğrusal veri kümelerinde başarı gösterirken, Lojistik Regresyon yöntemi kategorik veri ile çalışıldığında başarı göstermektedir. Diğer taraftan verilerin numerik ve doğrusal olmadığı veri setlerinde ise Kernel Lojistik Regresyon modeli en başarılı sınıflayıcı olmuştur. Destek Vektör Makineleri ile sınıflama yaparken seçilen fonksiyonun türü başarıyı etkileyen en önemli faktör olmuştur (Tablo 7).

Sınıf ağırlıklarını eşitlendiğinde Yapay Sinir Ağları (Tablo 4) ve Kernel Lojistik Regresyon (Tablo 14) bu durumdan olumlu etkilenmiş ve sınıflama başarı oranları artmıştır. Fakat diğer taraftan Destek Vektör Makineleri dengelenmemiş sınıf ağırlığı (n=75) ile çalıştığımızda başarı oranı 64 % iken veriler dengelendiğinde başarı oranı % 50'ye gerilemiştir (Tablo7). Destek Vektör Makinelerinde verinin dengelenmemiş olması marjinin daha kolay hesaplanacağı şeklinde yorumlanabilir.

Cergy Pontoise üniversitesi ekonomi bölümünün bir araştırmasında banka verileri ile kredi puanlaması yapılmak istenmiş. Kullanılan Lojistik Regresyon ve YSA yöntem sonuçları karşılaştırıldığında üretilmiş veri ve üretilmemiş veriden elde ettikleri sonuçlarının istatistik olarak farklı anlamlar ifade ettiğine değinmişlerdir (üretilmiş veri doğruluk oranı LR:0.68, YSA:0.69 ve üretilmemiş veri doğruluk oranı LR:0.74 YSA:0.76). Doğruluk oranı düşmesine rağmen istatistik olarak; duyarlılık ve geçerlilik dikkate alındığında üretilmiş verinin verdiği oranlar anlamlı çıkarken üretilmemiş veriden elde edilen sonuçlar anlamlı bulunmamıştır. Bu durum; özellikle sınıf dengesizliğinin oluşturduğu bir sonuç olarak açıklanmıştır. (Sayeh, 2016).

WEKA'nın kendi verileri ile yaptığımız çalışmada her üç yöntemde % 90 üstünde başarı göstermiştir (Tablo 17). DVM yanlış sınıflama' değeri diğer yöntemlere göre daha düşük gözlenmiş ve bunu 'kesinlik' ölçütünde desteklemiştir (Tablo 18). Tablo 17 değerlendirilirken Sayeh'in de savunduğu gibi sonuçlar değil bu sonuçları destekleyen 'doğruluk' ve 'kesinlik' değerleri dikkate alınmalıdır.

Destek Vektör Makinelerinde standart sapma büyüdükçe ve buna bağlı olarak verilerin farklılığı artıkça sınıflamada daha başarılı sonuçlar elde edilmiştir (Tablo 7).

Yapay Sinir Ağlarında standart sapma arttıkça başarı oranı düşmüştür (Tablo 5). Standart sapmanın artışından Kernel Lojistik Regresyon ve DVM modeli olumlu etkilenmiştir (Tablo 14). (Üretilmiş veya dengelenmiş verilere ait standart sapma değerleri Tablo 1'de gösterilmiştir.)

YSA'nın aktif bir öğrenme sisteminin olması, simültane olarak çıktılara bakıp model yapılandırması, nonlinear olabilmesi, eksik verileri tamamlayabilmesi (oysa geleneksel yöntemler verinin bütününe bakarak, genel parametreler ile sonuçlara ulaşmaya çalışır) ve tüm veri setini ilgilendirmeyen değişkenleri sistemden çıkarabilme yeteneğine sahip olması modelin üstünlüğüdür. YSA'nda örnek veri seti küçük, ya da değişkenler veya kayıtlar arasındaki oranın dengesiz olduğu durumlarda da çalışır. YSA bu problemi boyutsallıktan kaçınarak kurmaya çalışır. (Shi, 2012) Tablo 5'in aksine, YSA'nın standart sapma değerinden en az etkilenen model olması beklenirdi.

YSA'nın performansını, multilineer regresyon yöntemi ile karşılaştıran Oxaline ve oxazolon ilaç dozlarının antitüberküloz tedavi çalışmasında, nonlinear veri içeren moleküler incelemelerde YSA'nın daha başarılı olduğu ispatlanmıştır. Bu başarı YSA'nın desen tanıma yeteneğine dayandırılmıştır. (Drosowamy, 2013)

Çalışmamızda Simple Logisitik Regression yöntemi denenmiş ve linear olmayan veri setimiz nedeniyle sınıflama başarı oranı düşük bulunmuştur (Tablo 13).

YSA karışık veri kümelerinde başarılı sonuçlar vermektedir. Şiraz Tıp Bilimleri Üniversitesi Biyoistatistik Bölümü; veri setinin sayısal, sıralayıcı ve kategorik olduğu (yaş, çalışma saati, kursa katılım, medeni durum v.s.) öğrenci başarısının ölçüldüğü bir araştırma yapmış ve Lojistik Regresyonun 0.77, YSA'nın 0.84 doğru sınıflama yeteneğine sahip olduğunu ve sonuç olarak YSA'nın daha başarılı bir sınıflayıcı olduğunu savunmuştur. (Teshinzi, 2015) Bu durum Tablo 13'ü desteklemektedir.

Kernel Logisitik Regresyon bağımsız değişkenlerin sayısal (numerik) olduğu veri kümelerinin analizinde daha fazla başarı gösteren bir sınıflayıcı modeldir. Bu nedenle nominal veriler ile denendiğinde Kernel Logistik Regresyon sınıflayıcı başarı oranını düşürmüştür, ancak YSA ve DVM modelleri de beklenenin aksine sayısal verilerde yakaladığı başarıyı gösterememiştir. Tablo 17'deki sonuçlarımız bu sonucumuzu ve aynı zamanda Teshinzi'nin görüşünü desteklemekte ve veri setinin

özelliği değıştikçe yöntemlerin başarı oranlarının ciddi anlamda değıştiğini göstermektedir.

WEKA programından hazır olarak kullandığımız açıklayıcı değışkenlerimiz sayısal özellikte olduğu için her üç yöntem yüksek düzeyde başarı sunmuştur (Tablo 17).

Tahran Üniversitesi Biyoistatistik ve Epidemiyoloji Bölümü, 1271'i kafa travması tanılı toplam 8452 travmalı hastaların verilerine dayanarak, ölümcül travma hazırlayıcı etkenlerin model yapılandırılması çalışması yapmıştır. Bu çalışmada kalibrasyon ve discriminant analiz sonuçlarına göre (YSA-ROC:0.96, LR-ROC:0.95, YSA-Hosmer-Lemeshow:0.41, LR-Hosmer-Lemeshow:0.53) YSA üstün bulunurken, doğru sınıflama başarısında LR yöntemi üstün bulunmuştur (YSA accurary:0.95, LR accurary: 0.96) Çalışma sonucunda büyük veri kümeleri ile çalışılırken Lojistik Regresyon yönteminin yetersiz olduğu savunulmuştur.(Eftekhari 2005)

Verileri türeterek çoğalttığımızda ve 'n' değeri arttığında, Yapay Sinir Ağları (Tablo 4), Kernel Lojistik Regresyon (Tablo 14) ve SMO (Tablo 7) yöntemlerinin başarı oranı ciddi oranda yükselmiş, Simple Logistic Regression (Tablo 13) ve Limsvm Destek Vektör Makineleri (Tablo 8) ise olumsuz etkilenmiş ve başarı oranı düşmüştür.

Sınıflayıcı üzerinde değışken bazında etki ölçülecekse veya öznitelik seçimi yapılacaksa Lojistik Regresyon yöntemi tercih edilmelidir. Çünkü Lojistik Regresyon değışkenlerin odds ve β değerlerini görmemizi sağlar. Eğer veri bazında odds oranları çalışmayı açıklayacaksa Kernel Lojistik Regresyon her bir verinin ayrı ayrı logit değerlerini görmemizi sağlar. Destek Vektör Makineleri her bir verinin vektör uzunluğunu ve eksi ya da pozitif sınırdaki kaldığını görüntülememizi sağlar. Yapay Sinir Ağları ise her bir düğümün eşik değerini ve bu düğümlerde her bir değışkenin ağırlığını görmemizi sağlar.

Cumhuriyet Üniversitesi Jeoloji Fakültesi, Tokat'ın Kat köyünde yaptıkları bir araştırmada, toprak kayması haritası geliştirmek üzere, frekans değerlerini ölçmüş ve çalışmalarında YSA'nın 0.852, Lojistik Regresyonun 0.842 oranında aynı duyarlılığa sahip olduğuna karar vermişlerdir. (Yılmaz, 2009)

Araştırmamızda Yapay Sinir Ağları 0.97, Destek Vektör Makineleri 0.98 ve Kernel Lojistik Regresyon 0.98 oranında sınıflama başarısı göstermiştir (Tablo 17).

YSA yöntemi ile çalışılırken, sonuçlar 0.52 saniye gibi kısa bir sürede oluştu. Ancak tekrarlayan ölçüm özellikleri gösteren verilerde, örneğin su içinde artan organik maddeler gibi, çalışma uzun zamanlar alabilir. Buna örnek olarak, Bejing Teknoloji Üniversitesi; Çin'in güneybatısında yer alan Fuxian Gölünün su kalitesi göstergelerine dayalı olarak, ötrofikasyon (Besin artışına bağlı olarak alglerin ve plaktonların artması.) düzeyini incelemek için 10 ay boyunca YSA eğitmişlerdir. Bu çalışmanın sonucunda, YSA yöntemi indikatörleri tahminlemede, eğitim verisi 0.90, test verisi 0.76 başarılı olmuştur. (Huo vd.2013)

Bu duruma örnek olarak, Sao Paulo Eyalet Üniversitesi, hareket eden robotlar üzerine yaptığı çalışmada, navigasyon takip sisteminde direksiyon açılarının etkilerini ölçmüş ve araştırma yöntemi olarak YSA ve DVM ile çalıştıklarında iki yöntemin 5⁰C'lik açı farkı gösterdiğini ancak YSA'nın günler süren ölçümüne karşın DVM'nin sadece birkaç dakikada sonuç verdiğini ispatlamışlardır. (Jodas vd. 2013)

Çalışmamızda yöntemlerimizin eşit başarı düzeyine sahip olduğu bilgisine ulaştık fakat, Tayvan Koahsiung Tıp Fakültesinde yapılan 1998-2009 yılı verilerine dayalı retrospectif çalışmada, karaciğer kanser hastalarının ameliyat sonrası ölüm durumları çalışılmış ve YSA ve Log. Regresyon yöntemlerinin doğru tahminleme performansları karşılaştırılmıştır. YSA 0.97 ve LR ise 88.29 doğruluk oranı bilgisine ulaşmıştır. (Shi, vd.2012)

Ondokuz Mayıs Üniversitesi ise bu iki yöntemi bir arada kullanarak, Lojistik Regresyon ve YSA tabanlı yeni bir yöntem uygulanmıştır. Bu yeni hybrid yöntemde Lojistik Regresyondan elde edilen açıklayıcı değişkenler ve tahminler, İleri beslemeli YSA'nın giriş katmanı olarak kullanılmıştır. Çalışmada elde edilen sonuçlar akciğer kanseri sınıflama başarı değerleri; Lojistik Regresyon 0.81, YSA 0.84, LR-YSA hybrid yöntem:0.88 şeklinde kaydedilmiş ve Hybrid LR-YSA yönteminin hem Lojistik Regresyon hem de YSA'dan daha önerilir sonuçlar oluşturduğu savunulmuştur. (Tunç, 2012)

8.SONUÇ ve ÖNERİLER

Yapay Sinir Ağları, Destek Vektör Makineleri ve Lojistik Regresyon yöntemlerinin sınıflamadaki başarı düzeylerinin karşılaştırıldığı bu çalışmada, yöntemlerin sınıflama başarısı aynı ve üç yöntem % 90'nın üstünde başarı göstermiştir. Uygulanan duyarlık ve geçerlilik testleri yöntemlerin ortaya koyduğu sınıflama başarı yüzdelerini desteklemiştir.

Araştırmacı yöntem belirlerken çalıştığı veri setini iyi tanımalı ve uygulayacağı yöntemin veya bu yöntemde hangi fonksiyonun elindeki veri için uygun olabileceği deneyimine sahip olmalıdır.

Küçük 'n' ve düşük standart sapma değerine sahip veri kümeleri ile çalışılacaksa, Lojistik Regresyon yöntemi en iyi sınıflayıcı özelliğindedir. Örnek büyüdükçe, bağımsız değişken sayısı arttıkça, değişkenler arasında çoklu ilişki söz konusu ise ve kayıp veriler mevcutsa yöntemin kullanılmaması gereklidir. Bu duruma alternatif Kernel Lojistik Regresyon yöntemi, özellikle açıklayıcı değişkenlerin sayısal özellikte olduğu durumlarda kullanılabilir.

Yapay Sinir Ağları yarı parametrik varsayıldığından, parametrik test koşullarını sağlamayan veri setlerinde iyi bir sınıflayıcıdır. Büyük veri setlerinde, normal dağılıma sahip olmayan verilerde, bağımlı ve bağımsız değişken arasında kompleks ve nonlinear ilişki varlığında da çalışabilme yeteneğine sahiptir. Kayıp veri etkisini azaltabiliyor olması, yöntemin eksik veri setlerinde de çalışmasını sağlar. 'n' sayısının büyük olduğu veri setlerinde, sınıflama başarısı yüksektir.

Yapay Sinir Ağları eğitim denilen ve dışarıdan müdahale edilebilecek basit bir matematik model ile çalışıyor olması nedeniyle dez avantaj olabilir. Yine katsayıların ve odds oranlarının tespit edilememesi, Yapay Sinir Ağları araştırma koşullarını yazarken, anlatırken araştırmacıya zorluklar yaşatabilir. Yapay Sinir Ağlarının sınıflamayı optimize etmek için büyük eğitim setine ihtiyaç duyması, başlatma ve ayarlama gibi bireysel parametre değerlerini gerektiriyor olması, güven aralığı (confidence) ve katsayılarını (coefficient) kolaylıkla hesaplayamıyor olması, ağırlıkları programın kendisinin üretmesi noktasında açıklanamaz olması karşılaşılabilecek sorunlardır.

Üretilmemiş, gerçek ve düşük 'n' sayıda veri setiyle çalışılacaksa Destek Vektör Makineleri tercih edilmelidir. Yöntem standart sapmanın büyük ve sınıf

sayılarının eşit olmadığı modellerde özellikle tercih edilmelidir. Veri setinde $n=136$, Destek Vektör Makineleri için büyük olabilir.

WEKA paket programının halka açık bir program olması ve diğer istatistik programları kadar geniş ara yüzlere sahip olması nedeniyle bilimsel arařtırmalarda kullanımı desteklenmelidir.

9. KAYNAKLAR

Akçay, A., (2009), Brailer coccidiosis'inde risk faktörlerinin lojistik regresyon analizi ile belirlenmesi. Doktora Tezi, Ankara Üniversitesi Biyoistatistik Anabilim Dalı, Ankara, (Danışman: Prof. Dr. Okan Ertuğrul)

Akkuş, Ö.,Demir, S., Karasoy D., (2008), İki düzeyli bağımlı değişken modelinin yarı parametrik tahmini. *İstatistikçiler Dergisi, 1, 135-143 [Elektronik Dergi]*

Alic, A., Emilsson, C., (2016), Med multinomial respective ordinal logistic regression. *Stokholm Royal Teknik Enstitute [Elektronik Dergi]*

Altun, A., Eminoğlu, B., Tezekçi, C., Neural Networks for Signal Processing. (Erişim Tarihi:18.02.2017), <http://www.ieee.org>

Arı, A., Önder, H., (2012), Farklı veri yapılarında kullanılacak regresyon yöntemleri. *Ondokuz Mayıs Üniversitesi, Tarım bilimleri dergisi, 28, 168-174 [Elektronik Dergi]*

Ayhan, S., (2014), Destek vektör makineleriyle sınıflandırma problemlerinin çözümü için çekirdek fonksiyonu seçimi. *Eskişehir Osmangazi Üniversitesi İibf Dergisi, 9(1), 175- 198 [Elektronik Dergi]*

Ayhan, S., (2006), Sıralı lojistik regresyon analiziyle türkiye'deki hemşirelerin iş bırakma niyetini etkileyen faktörlerin belirlenmesi. Yüksek Lisans Tezi, Osman Gazi Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir, (Danışman: Yrd. Doç.Dr. Zeki Yıldız)

Bamakan, A., Wang, B.,Ravason,C., (2016), Parameters optimization for nonparallel support vector machine by particle swarm optimization. *Procedia Computer Science, 91, 482-491 [Elektronik Dergi]*

Bayata, H.F., Hattatoğlu,F, (2010), Yapay sinir ağları ve çok değişkenli istatistik yöntemlerle trafik kaza modellenmesi. *Fen Bilimleri Enstitüsü Dergisi, 3, 207-219 [Elektronik Dergi]*

Bayraktar, D., Çebi,F., Yöneylem araştırması. (Erişim Tarihi: 27.02.2017), <http://web.itu.edu.tr>

Bender, R., Grouven, U., (1997), Ordinal logistic regression in media research. *Journal of the Royal College of Physicians of London, Vol 31 (5), 546-551 [Elektronik Dergi]*

Bilgin, M., (2012), Türetilmiş ikili heterojen veri yapılarında genel, sağlam ve kesin lojistik regresyon yöntemlerinin karşılaştırılması. Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi Sağlık Bilimleri Enstitüsü Biyoistatistik ve Tıbbi Bilişim Anabilim Dalı, Eskişehir, (Danışman: Yrd. Doç. Dr. Ertuğrul Çolak)

Bishop, C.M., (2005), Neural network for pattern recognition. 2 Baskı, New York: Oxford Press, 196

Bissocat, A.C.G., Salgado, S.A.B., Balestrassi, P.P., v.d. (2016), Comparison of neural networks and logistic regression in assessing the occurrence of failures in steel structures of transmission lines. *The Open Electrical and Electronic Engineering Journal*, 10, 11-26 [*Elektronik Dergi*]

Brown, L.K., Schult, J.R., and Forsberg, A.D., (2002), Predictors of retention among HIV/hemophilia health care professionals. *General Hospital Psychiatry*, 24, 48-54 [*Elektronik Dergi*]

Camdeviren, H.A., Yazıcı, A.C., Akkuş, Z., Buğdaycı, R.,Sungur, M.A.,(2007), Comparison of logistic regression model and classification tree: an application to postpartum depression data. *Science Direct* 32, 987-994 [*Elektronik Dergi*]

Cortes, C., Vapnik, V., (1995), Support vector network. *Machine Learning*, 20, 273-297 [*Elektronik Dergi*]

Coşkun, C., Baykal, A., (2011), Veri madencilğinde sınıflandırma algoritmalarının bir örnek üzerinde karşılaştırılması. Akademik Bilişim Konferansı Bildirileri, İnönü Üniversitesi Malatya

Çayiroğlu, İ., İleri Algoritma Analizi. (Erişim Tarihi: 18.02.2017), <http://ibrahimcayiroglu.com>

Çelik, M.Y., (2015), SPSS ile bilimsel araştırma ve biyoistatistik çözümlene yöntemleri. 2.Baskı, İstanbul: 481-490

Çolak, E., (2002), Koşullu ve sınırlanmış lojistik regresyon yöntemlerinin karşılaştırılması ve bir uygulama. Yüksek Lisans Tezi, Osmangazi Üniversitesi Sağlık Bilimleri Enstitüsü, Eskişehir

Datong, C., (2012), Comparison of support vector machine and neural network for text texture verification. (Erişim Tarihi: 26.02.2017), <http://www.idiap.ch>

- Dolgun, N.,A.,(2012), An adapted wald test statistic to determine the variables which do not satisfy the proportionality assumption in the adjacent category logistic regression model. Doctoral Dissertation, Hacettepe University Institute of Health Sciences, Ankara, (Thesis Supervisor: Prof. Dr. Osman Saraçbaşı)
- Dorrius, M.D., Dijkstra, H., Oudkerk, M., (2014), Effect of ‘b’ value and pre-admission of contrast on diagnostic accuracy of 1.5 tbreast dwi. *European Radioloji*, 24, 2835-2847 [*Elektronik Dergi*]
- Dreiseitl, S., Ohno-Machado, L., (2002), Logistic regression and ANN classification models: A methodology review. *Journal of Biomedical Informatics*, vol 35. Issues 56, 352-359 [*Elektronik Dergi*]
- Droswamy, C.M., (2013), Performance analysis of neural network models for oxazolines and oxazoles derivatives descriptor and techniques. *International Journal of Information Sciences and Techniques*, vol 3, 6 [*Elektronik Dergi*]
- Eftekhari, B., Mohammad, K., Ardebili, H.E., (2005), ‘Comparison of artificial neural network and logistic regression model for prediction of mortality in head trauma based on initial clinical data. *BMC Medical Informatics and Decision Making* Doi:10.1186 [*Elektronik Dergi*]
- Elmas, Ç., (2003), Yapay sinir ağları. 1. Baskı, Ankara: Seçkin Yayın Evi, 92-143
- Ercan, S., (2011), Destek vektör makineleri kullanarak patent değerlendirme. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, (Danışman: Yrd. Doç. Dr. Gülgün Kayakutlu)
- Erdoğan, Ş., Ayhan, S., (2016), Destek vektör makineleri ile sınıflandırma problemlerinin çözümü için çekirdek fonksiyonu seçimi. *ESOGÜ, Nisan 2014*, 181-182 [*Elektronik Dergi*]
- Ferrari, D., Kuriki, H.U., Silva, C.R., Alves, N., (2014), Diagnostic accuracy of the electromyography parameter associated with anterior knee pain in diagnostic of patellofemoral pain syndrome. *Archives of Physical Medicine and Rehabilitation*, 95, 1521-1526 [*Elektronik Dergi*]
- Göksülük, D., (2011), Penalized logistic regression. Master Thesis, University Graduate School of Natural and Applied Sciences, İzmar, (Thesis Supervisor: Prof. Dr. Aylin Alın)

Gönül,Y, Ulu,Ş, Bucak,A, Bilir, A., (2015), Yapay sinir ağları ve klinik araştırmalarda kullanımı. Genel Tıp Dergisi, 25, 104-111

Hsu, C.W., Chang, C.J., Lin, C.J., (2003), A practical guide to support vector classification. *datascienceassn.org. [Elektronik Dergi]*

Huo, S. He, Z., Su, J., Xi, B., (2013), Using artificial neural network models for eutrophication prediction. *Procedia Environmental Sciences, 18, 310-316 [Elektronik Dergi]*

Jodas, D., Marranghella, N., Pereira, A., v.d. (2013), Comparing support vector machines and artificial neural network in the recognition of steering angle for driving of mobile robots through paths in plantations. *Procedia, 18, 240-249 [Elektronik Dergi]*

Kavzoğlu, T., Collesen, İ., (2010), Destek vektör makineleri ile uydu görüntülerinin sınıflandırılmasında kernel fonksiyonlarının etkilerinin incelenmesi. *Harita Dergisi, 144 [Elektronik Dergi]*

Kaya, Y., (2007), Binary karışımli lojistik regresyon. Yüksek Lisans Tezi, Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü, Zooteknik Anabilim Dalı, Van, (Danışman: Yrd. Doç. Dr. Abdullah Yeşilova)

Keerthi, S.S., Shevade, S.K., Bhattacharya,C., Murthy, K.R.K. (2001), İmprovements to platt's smo algorithm for svm classifier design. *Technical Report CD-99, 14 [Elektronik Dergi]*

Keskin, S., (2004), 2x2 Tablolarda bazı örnek genişlikleri ve tip I. hata seviyeleri için kappa istatistiğine ait ampirik olarak gerçekleşen kritik değerler. *Tarım Bilimleri Dergisi, 10, 169-173 [Elektronik Dergi]*

Kökçü, A., Köse, B., Bulut, C.,v.d., Kollektif öğrenme algoritmaları ile çocuklarda obezite hastalığına yakalanma olasılığının hesaplanması. (Erşim Tarihi: 30.04.2014), <https://www.ce.yildiz.edu.tr>

Kuyucu, Y.E., (2012), Lojistik regresyon analizi, yapay sinir ağları ve regresyon ağaçları yöntemlerinin karşılaştırılması ve tıp alanında bir uygulama. Yüksek Lisans Tezi, Gazi Osman Paşa Üniversitesi Sağlık Bilimleri Enstitüsü Biyoistatistik Anabilim Dalı, Tokat, (Danışman: Yrd. Doç. Dr. Ünal Erkorkmaz)

- Landwehr, N., Hall, M., Frank, E.,(2005), Logistic model trees. *Machine Learning*, 59, 161-205 [Elektronik Dergi]
- Le Cessie, S., Van Houwelling, J.C., (1992), Ridge estimators in logistic regression. *Royal Statistical Society*, 41, 191-201 [Elektronik Dergi]
- Liala Renjie, (2015), Support vector machine. *CSC 411,Toronto Üniversitesi* [Elektronik Dergi]
- Lodhi, H., Sounders, C., Shawe, T., Cristianni, N.,Watkins, C., (2002), Text classification using string kernel. *Journal of Machine*, 2, 419-444 [Elektronik Dergi]
- Liu, A., Wang, B., v.d., (2017), An efficient instance selection algorithm to reconstruct training set for support vector machine. *Knowledge-Based-System*, 116, 58-73 [Elektronik Dergi]
- Mickey, L., Moreland, M.B., Clark, S.E., (2014), Missing the information needed to perform roc analysis. *National Science Foundation SES-1155248* [Elektronik Dergi]
- Ocakođlu, G.,(2006), Lojistik regresyon analizi ve yapay sinir ađları tekniklerinin sınıflama özelliklerinin karşılaştırılması ve bir uygulama. Yüksek Lisans Tezi, Uludađ Üniversitesi Sağlık Bilimleri Enstitüsü Biyoistatistik Anabilim Dalı, Bursa, (Danışman: Prof. Dr. İsmet Kan)
- Öztemel, E, (2012), Yapay sinir ađları. 3. Baskı, Ankara: Papatya Yayıncılık, 37
- Platt, J.C.,(1998), A fast algorithm for training support vector machines. *Microsoft Research, MSR-TR-98-14*, 21 <http://microsoft.com> [Elektronik Dergi]
- Sarıdereli, A., (2010), Boyahanelerde laboratuvar ile işletmeler arasındaki renk deđişimlerinden kaynaklanan hataların azaltılması. Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü Tekstil Mühendisliđi Anabilim Dalı, İzmir, (Danışman: Doç. Dr. Merih Saruşıık)
- Sayeh, w., Bellier, A., (2016), Neural network versus logistic regression; the best accuracy in predicting credit rationing decision. Thema, University of Cergy-Pontoise, (Erişim Tarihi:18.11.2017) <https://gdrenice2015sciencesconf./53551>
- Shi, H.Y., Lee, K.T., Lee H.H., Ho W.H., Sun, D.P., (2012), Comparison of artificial neural network and logistic regression model for predicting in hospital mortality after primary liver cancer surgery. *Plasone*, 7(4), 35781 [Elektronik Dergi]

- Solomon, P.E., (2002), İnsan anatomisi ve fizyolojisine giriş. 4. Basım, İstanbul: Birol Yayınevi, 93-118
- Suykens, A., Gestel, B., Brabanter, C., (2002), Least squares support vector machines. *Londra, World Scientific Publishing, 71 [Elektronik Dergi]*
- Şeker, S.E., (2013), İş zekası ve veri madenciliği. Birinci Baskı, İstanbul: Cinius Yayınları, 182-202
- Taylor, J.S, Christianini, N., (2000), An introduction to support vector machines and other kernel-based learning methods. *Cambridge University Press, 18, 687-689 [Elektronik Dergi]*
- Teshinzi, S.H., Ayatollahi, S.M.T., (2015), A comparison of logistic regression model and artificial neural network in predicting of student's academic failure. *ACTAinformMED, doi:10.5455, 296-300 [Elektronik Dergi]*
- Tibshirani, R., (2014), Advanced methods for data analysis. *Statistics 36-402/36-608 [Elektronik Dergi]*
- Tong, S., Kaller, D., (2001), Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research, 01, 45-66 [Elektronik Dergi]*
- Tunç, T., (2012), A new hybrid method logistic regression and feedforward neural network for lung cancer data. Hindawi Publishing Corporation, Vol 2012, Article ID:241690
- Ünal, A., (1996), Lojistik regresyon analizi ve uygulaması. Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Fakültesi
- Ürük, E., (2007), İstatistiksel uygulamalarda lojistik regresyon analizi. İstanbul, Yüksek Lisans Tezi, Marmara Üniversitesi.
- Vupa, Ö., (2009), Assessment of interaction and confounding effects in logistic regression model: an application in a case-control study of stomach cancer. Doctoral Dissertation, Dokuz Eylül University Graduate School of Natural and Applied Sciences, İzmir, (Thesis Supervisor: Prof. Dr. Gül Ergür)

Yakut, E.,Elmas,B.,Yavuz,.S.,(2014), Yapay sinir ađları ve destek vektör makinalari yöntemleriyle borsa endeksi tahmini. *Süleyman Demirel Üniversitesi İktisadi ve idari Bilimler Fakültesi Dergisi*, 19,139-157 [*Elektronik Dergi*]

Yalçın, H., Yapay sinir ađları. (Erişim Tarihi:11.11.2017), <http://web.itu.edu.tr>

Yavuz, S., Deveci, M., (2013), İstatiksel normalizasyon tekniklerinin yapay sinir ađın performansına etkisi. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, Sayı: 40, 167-187 [*Elektronik Dergi*]

Yılmaz, I., (2009), Landslide susceptibility mapping using frequency ratio, logistic regression, artificial neural networks and their comparison. *Computer and Geoscience*, 35, 1125-1138 [*Elektronik Dergi*]

Yılmaz, Ö.A, Ankaralı, H.C., Aydın, D., Sıraçlı, Ö., (2013), Tıbbi tahminde alernatif bir yaklaşım: destek vektör makineleri. *Türkiye Klinikleri, J Biostat*, 5(1), 19-28

Zhu, J., Hastie, T., (2005) Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics*, Volume 14 (1), 185-205

10.EKLER

EK 1

WEKA programından elde edilmiş veri seti.

```
@relation weka.datagenerators.classifiers.classification.RandomRBF-S_1_-n_75_-  
a_10_-c_2_-C_50
```

```
@attribute a0 numeric
```

```
@attribute a1 numeric
```

```
@attribute a2 numeric
```

```
@attribute a3 numeric
```

```
@attribute a4 numeric
```

```
@attribute a5 numeric
```

```
@attribute a6 numeric
```

```
@attribute a7 numeric
```

```
@attribute a8 numeric
```

```
@attribute a9 numeric
```

```
@attribute class {c0,c1}
```

```
@data
```

```
0.814341,0.809854,0.512249,0.419757,0.216842,0.186186,0.626497,0.166968,0.50  
4129,0.856797,c1
```

```
0.192669,0.997833,0.514416,0.655416,0.222015,0.940929,0.845017,0.75514,0.337  
593,-0.070651,c1
```

```
1.009699,0.823296,0.84259,0.156677,0.62608,0.025761,0.616653,0.199607,0.8955  
01,0.716553,c1
```

```
0.394982,0.450572,0.693571,0.775373,0.556944,0.549397,0.588843,0.836024,1.03  
4899,0.555447,c0
```

```
0.461656,-0.218205,0.472861,0.353269,0.942354,0.773539,0.605775,0.226246,-  
0.165556,0.192277,c0
```

0.846376,0.312877,0.527302,0.631353,0.632716,0.599007,0.051462,0.35604,0.611713,-0.054753,c1

0.137762,0.486548,0.536024,0.571573,0.205502,0.611957,0.173217,0.001639,0.163987,0.177649,c1

0.150437,0.545046,0.592044,0.604286,0.242566,0.649714,0.244466,-0.058775,0.209396,0.126088,c1

0.296079,0.581593,0.186106,0.853404,0.711697,0.945089,0.446743,0.324922,0.795422,0.369829,c1

-

0.095461,0.191065,0.605682,0.863042,0.486083,0.768623,0.351786,0.316372,0.483244,0.22598,c0

-0.051118,-0.702076,0.387363,0.729015,-0.349491,2.119873,0.423048,0.919255,1.100751,1.546522,c1

-

0.434441,0.297022,0.521717,0.632549,0.588432,0.261921,0.727004,0.518934,0.346131,0.555559,c0

0.528506,0.941376,0.431293,-0.015291,0.314578,0.529902,0.808705,0.875094,0.537592,0.654815,c1

0.226222,0.475419,0.905204,0.263339,0.655447,0.466635,0.937651,0.514031,0.082338,0.13044,c1

0.460405,0.782746,0.743625,0.44575,0.629042,0.144432,0.369141,0.548804,0.099402,0.640974,c1

0.348131,0.859418,0.660004,-0.055376,0.338907,0.731232,0.847005,0.956115,0.448827,0.927527,c1

0.729241,0.726381,0.264488,0.67915,0.230802,0.54991,0.496762,0.251444,0.927953,0.632458,c1

0.307432,0.321224,0.343779,0.328875,0.946175,0.801333,0.472197,0.281779,-0.045736,0.641495,c0

0.361466,0.796555,0.432643,0.079096,0.390639,0.566418,0.85237,0.984464,0.407959,0.818377,c1

0.827158,0.660197,0.823818,0.214896,0.318475,0.858679,0.092866,0.911283,0.48061,0.835002,c1

0.802748,0.199886,0.130004,0.435436,0.061506,0.633731,0.854827,0.96008,0.882545,0.571654,c1

0.558321,0.378937,0.379105,0.629818,0.892956,0.309035,0.179403,0.353035,0.543335,0.091283,c1

0.085769,0.375201,0.435139,-0.103023,0.165334,-0.448608,0.44402,0.519171,1.096568,0.608521,c1

0.221198,0.453349,0.93125,0.23268,0.636428,0.480392,0.90627,0.508052,0.10389,0.151667,c1

0.735044,0.407453,0.226265,0.315825,0.985386,-0.008262,0.946046,0.957644,0.949841,0.914724,c0

0.802598,0.252885,0.653076,0.672608,0.83653,0.418902,0.092824,0.544651,0.697267,0.053889,c1

-

0.107554,0.411914,0.934658,0.538866,0.550451,0.694942,0.079167,0.42436,0.853371,0.206978,c0

0.571691,0.466542,0.098128,1.016682,0.116121,1.226251,0.599484,1.214504,0.921082,0.742161,c1

0.370673,0.055441,0.74226,-0.137477,0.73008,0.664688,0.772453,0.216477,-0.035877,0.433852,c0

0.204679,0.874944,-0.000563,0.369726,0.514797,0.7422,1.044547,1.027983,-0.035672,0.679357,c1

0.611442,0.658894,0.333786,0.505093,0.445555,0.211464,0.437817,0.263606,0.435888,1.003268,c1

0.795825,0.147134,0.697366,0.639588,0.780062,0.469626,0.263076,0.574719,0.681256,-0.088188,c1

0.209245,0.834763,0.395555,0.959991,0.520105,0.389854,0.26368,0.111387,0.219229,0.9847,c1

0.432385,0.982549,0.494977,0.973237,0.58019,0.164403,1.03193,1.134451,0.4868
6,0.423879,c0

0.488865,0.000624,0.327725,0.279948,0.833029,0.144902,0.539897,0.536876,0.33
7343,0.413334,c1

-0.165887,0.564747,0.093858,-

0.30853,0.507176,0.935193,1.054532,0.28288,0.850683,0.283498,c0

-0.065695,0.636744,-

0.275122,0.932764,0.159108,1.363163,0.977979,0.02399,0.631084,-0.259641,c0

-0.064157,2.12577,-

0.250237,0.660589,0.973706,1.105797,0.097506,0.551934,0.587733,0.097451,c1

-0.278093,-0.129253,0.771819,0.623009,0.133008,0.780532,-

0.007191,0.298572,0.358839,0.049191,c0

0.235731,0.865809,0.367111,0.982741,0.530612,0.433781,0.316156,0.152245,0.24
1488,0.964746,c1

0.174907,0.860927,0.385482,0.610822,0.66191,1.130034,0.920101,0.185473,0.179
926,0.788494,c0

-0.15975,-

0.077423,0.012191,0.982218,0.166194,0.785409,1.0118,1.047866,0.959215,0.9453
75,c1

0.696886,0.348719,0.164808,0.398003,0.966037,-

0.054957,1.019634,0.962247,0.890943,0.891133,c0

0.393448,1.031591,-

0.094682,0.49914,0.343776,0.487009,0.424117,0.633515,0.535531,0.454508,c1

0.637115,0.795431,0.387225,0.759863,0.28432,0.608639,0.509926,0.303472,0.886
991,0.558949,c1

0.190719,0.396679,0.34541,0.597971,1.03758,0.835836,0.25982,-

0.04419,0.907482,0.671289,c1

0.378868,0.156362,0.413075,0.397002,0.897536,0.683851,0.283072,0.175908,0.00
9643,0.624188,c0

0.573239,0.505041,0.197293,0.307619,0.417078,0.905623,0.268911,0.338875,0.485603,0.0085,c0

1.168869,1.232584,-0.089839,-

0.207087,0.092922,0.446851,0.50596,0.052908,0.185389,0.994312,c1

0.134695,0.488191,0.18185,0.183018,1.255614,0.399399,0.582389,0.631793,0.457683,0.503672,c0

0.027381,0.535848,1.149564,0.457263,0.868027,1.21033,0.045715,0.674571,0.642024,-0.175217,c0

0.233437,0.404453,0.988847,0.211692,0.586881,0.520461,0.940996,0.509289,0.072206,0.203568,c1

0.542814,0.032072,0.305126,0.350929,0.764539,0.021969,0.63053,0.413527,0.343434,0.33207,c1

0.161397,-0.601986,-0.094503,0.365556,1.249546,0.324881,1.506824,-0.161079,-0.430861,0.543336,c0

0.05403,0.147593,0.494085,1.406915,0.947877,1.129,0.139277,-

0.164117,0.435276,1.283063,c0

0.623939,-

0.110394,0.233416,0.223144,0.682064,0.032696,0.603625,0.541299,0.343947,0.318804,c1

0.953681,0.172418,0.926591,0.430967,0.828741,0.095667,0.17446,0.38057,0.334333,0.841143,c0

0.419448,0.052731,0.668508,0.056505,0.744118,0.297069,0.584572,0.015445,0.948194,0.396052,c1

1.039859,0.616235,0.784915,0.707698,0.298297,-

0.240254,0.77243,0.524407,0.416464,0.844305,c1

-

0.013998,0.232028,0.71984,0.656917,0.433068,0.649014,0.048951,0.321862,0.968455,0.174502,c0

0.242701,0.806013,0.431998,0.933084,0.440874,0.265721,0.383808,0.332643,0.164119,1.155169,c1

0.77354,0.82082,0.443267,0.278136,0.262919,0.053421,0.610585,0.307221,0.6486
31,0.96308,c1

0.199781,0.404487,0.996115,0.20087,0.538467,0.435166,0.984165,0.462319,0.016
717,0.197043,c1

0.160714,1.008902,0.158893,0.22476,0.837038,0.634953,0.041177,-
0.247447,0.513926,0.75167,c1

0.195731,0.84366,0.424347,0.943593,0.504424,0.369395,0.285071,0.095423,0.244
259,0.975501,c1

0.701355,0.055604,0.597339,0.582889,0.496328,0.84258,0.863939,0.99195,0.1309
72,0.09064,c1

0.703317,0.933176,0.308534,0.252571,0.755788,0.374707,0.668762,0.319874,0.03
7779,0.450995,c0

0.196567,0.034717,0.907598,0.87439,0.591772,0.481229,0.020288,0.051907,0.731
019,0.08915,c0

0.947727,0.805338,0.843321,0.182816,0.63237,0.012724,0.592193,0.217207,0.913
439,0.69131,c1

0.742218,0.264992,0.673886,0.918028,0.306677,0.685166,0.189221,0.908204,0.70
5102,0.492294,c0

0.861899,0.281492,0.715099,0.791399,0.877601,0.422446,0.119054,0.407656,0.79
0608,0.001191,c1

0.077544,0.331654,0.149802,1.065359,0.60633,1.040429,0.948693,0.743575,0.892
564,0.578175,c1

-0.240811,1.163674,0.653488,-0.358918,0.343511,0.917444,0.626193,-
0.270364,0.572531,0.042495,c0

0.768798,0.79042,0.156671,0.376543,0.709855,0.332487,0.662318,0.432088,-
0.052018,0.423209,c0

0.785232,0.262983,0.770786,0.975165,0.972634,0.21769,0.460162,0.125405,0.249
68,0.648131,c1

11.ÖZGEÇMİŞ

Adı Soyadı: DENİZ TEZER

Doğum Tarihi ve Yeri : 11.07.1976

Mail Adresi: deniztezer111@gmail.com

Unvanı: Öğretmen

Öğrenim Durumu: Lisans

Derece	Okul Adı ve Bölümü	Mezuniyet Yılı
Önlisans	Kırıkkale Sağlık Eğitim Enstitüsü Tıbbi İstatistik ve Bilgisayar Bölümü	2005
Lisans	İnönü Üniversitesi Malatya Sağlık Yüksek Okulu Sağlık Eğitimi	2010

Yayımları (Varsa)

Ödülleri (Varsa)

YAPAY SİNİR AĞLARI, DESTEK VEKTÖR MAKİNELERİ VE LOJİSTİK REGRESYON YÖNTEMLERİNİN SINIFLAMA BAŞARILARININ KARŞILAŞTIRILMASI

ORIJINALLIK RAPORU

%9

BENZERLİK ENDEKSİ

%8

İNTERNET
KAYNAKLARI

%3

YAYINLAR

%2

ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1	es.scribd.com İnternet Kaynağı	%3
2	acikarsiv.ankara.edu.tr İnternet Kaynağı	%1
3	www.researchgate.net İnternet Kaynağı	%1
4	pt.slideshare.net İnternet Kaynağı	<%1
5	Submitted to Afyon Kocatepe University Öğrenci Ödevi	<%1
6	readgur.com İnternet Kaynağı	<%1
7	www.scribd.com İnternet Kaynağı	<%1
8	www.tobb.org.tr İnternet Kaynağı	<%1