T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

İÇ HASTALIKLARI ALANINDA MAKİNE ÖĞRENMESİ YÖNTEMLERİNİN UYGULAMALARI VE KARŞILAŞTIRILMALARI

YÜKSEK LİSANS TEZİ

LEYLA TÜLÜ

DENİZLİ, KASIM - 2022

T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

İÇ HASTALIKLARI ALANINDA MAKİNE ÖĞRENMESİ YÖNTEMLERİNİN UYGULAMALARI VE KARŞILAŞTIRILMALARI

YÜKSEK LİSANS TEZİ

LEYLA TÜLÜ

DENİZLİ, KASIM - 2022

Bu tez çalışmasında Pamukkale Üniversitesi Girişimsel Olmayan Klinik Araştırmalar Etik Kurulu tarafından 21.12.2021 tarihli E-60116787-020-144739 sayılı izin ile Pamukkale Üniversitesi İç Hastalıkları Polikliniğine ait 2021 yılı hasta verileri kullanılmıştır.

Bu tezin tasarımı, hazırlanması, yürütülmesi, araştırmalarının yapılması ve bulgularının analizlerinde bilimsel etiğe ve akademik kurallara özenle riayet edildiğini; bu çalışmanın doğrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etiğe uygun olarak kaynak gösterildiğini ve alıntı yapılan çalışmalara atfedildiğine beyan ederim.

LEYLA TÜLÜ

ÖZET

İÇ HASTALIKLARI ALANINDA MAKİNE ÖĞRENMESİ YÖNTEMLERİNİN UYGULAMALARI VE KARŞILAŞTIRILMALARI

YÜKSEK LİSANS TEZİ

LEYLA TÜLÜ

PAMUKKALE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

(TEZ DANIŞMANI: DR. ÖĞR. ÜYESİ BEDRİ BAHTİYAR)

DENİZLİ, KASIM - 2022

İnsan sağlığı, kaliteli yaşamı doğrudan etkileyen bir faktördür. Bu kalitenin bozulmaması için doğru zamanda gerekli testler yapılmalı ve tedbir alınmalıdır. Ancak artan nüfus sayısıyla hekimlere düşen iş yoğunluğu da doğrudan artmaktadır. Bu sebeple hastalara ayrılan süre kısalmaktadır. Bu sürede problemi anlayabilme ve hastalık teşhisi koyma süreci ise sağlık alanının en önemli ve en büyük problemlerden biridir. Bu süreci kolaylaştırmak için yapay zeka ile çeşitli çözümler uygulanmaktadır. Bu tez çalışmasında Pamukkale Üniversitesi Hastanesi İç Hastalıkları Polikliniğine başvurmuş hastaların kan testleri bilgilerini içeren veri seti ele alınarak makine öğrenmesinin çok-sınıflı ve çok etiketli sınıflandırma özelliği üzerinde çalışılmıştır. Ayrıca literatürde fazlaca çalışılmış olan Pima Indian Diabetes veri seti üzerinde de ikili sınıflandırma çalışması yapılmıştır. Veri setlerine yapay sinir ağı, destek vektör makineleri ve hafif gradyan artırma makineleri makine öğrenmesi yöntemleri uygulanmış ve modellerin performansları karşılaştırılmıştır.

ANAHTAR KELİMELER: Endokrinoloji, Veri Bilimi, Makine Öğrenmesi, Çok-Sınıflı Sınıflandırma, Çok Etiketli Sınıflandırma.

ABSTRACT

APPLICATIONS AND COMPARISONS OF MACHINE LEARNING METHODS IN THE FIELD OF INTERNAL DISEASES

MSC THESIS

LEYLA TÜLÜ

PAMUKKALE UNIVERSITY INSTITUTE OF SCIENCE

ELECTRICAL AND ELECTRONICS ENGINEERING

(SUPERVISOR: ASSIST. PROF. BEDRİ BAHTİYAR)

DENİZLİ, NOVEMBER 2022

Human health is a factor that directly affects the quality of life. In order not to deteriorate this quality, necessary tests should be done at the right time and precautions should be taken. However, with the increasing population, the workload of doctors directly increases. For this reason, the time allocated to patients is shortened. In this period, the process of understanding the problem and diagnosing the disease is one of the most important and biggest problems in the field of health. To facilitate this process, various solutions are implemented with artificial intelligence. In this thesis, the multi-class and multi-label classification feature of machine learning was studied by considering the data set containing the blood test information of the patients who applied to the Pamukkale University Hospital Internal Diseases Polyclinic. In addition, a binary classification study was carried out on the Pima Indian Diabetes data set, which has been studied extensively in the literature. Artificial neural network, support vector machines and light gradient boosting machine machine learning methods were applied to the datasets and the performances of the models were compared.

KEYWORDS: Endocrinology, Data Science, Machine Learning, Multi-Class Classification, Multi-Label Classification.

İÇİNDEKİLER

Sayfa

ÖZET ..... **i**
ABSTRACT ..... **ii**
İÇİNDEKİLER ..... **iii**
ŞEKİL LİSTESİ ..... **iv**
TABLO LİSTESİ ..... **vi**
KISALTMALAR LİSTESİ ..... **vii**
ÖNSÖZ ..... **viii**
1. GİRİŞ **1**
2. MAKİNE ÖĞRENMESİ İLE SINIFLANDIRMA **8**
2.1 Yapay Sinir Ağları 8
2.1.1 Aktivasyon Fonksiyonları 10
2.1.2 Yapay Sinir Ağlarında Sınıflandırma 15
2.2 Destek Vektör Makineleri 17
2.2.1 Doğrusal Sınıflandırma 17
2.2.1.1 Esnek Marjinli Sınıflandırma 20
2.2.2 Doğrusal Olmayan Sınıflandırma 22
2.3 LightGBM 24
2.4 Sınıflandırma 26
2.4.1 İkili Sınıflandırma İçin Değerlendirme Metrikleri 28
2.4.2 Çok-Sınıflı Sınıflandırma İçin Değerlendirme Metrikleri 30
2.4.3 Çok-Etiketli Sınıflandırma İçin Değerlendirme Metrikleri 31
2.4.3.1 Örnek Temelli Metrikler 32
2.4.4 Ölçeklendirme 34
2.4.5 Çapraz Doğrulama 35
2.4.6 Bire Karşı Geriye Kalan Sınıflandırması 35
3. DENEYSEL SONUÇLAR **37**
3.1 Kullanılan Donanımlar ve Yazılımlar 37
3.2 Endokrinoloji Veri Seti 38
3.2.1 Veriyi Anlama 38
3.2.2 Çok Sınıflı Sınıflandırma Sonuçları 41
3.2.2.1 Yapay Sinir Ağı ile Sınıflandırma 45
3.2.2.2 Destek Vektör Makineleri ile Sınıflandırma 48
3.2.2.3 Hafif Gradyan Artırma Makineleri ile Sınıflandırma 50
3.2.3 Çok Etiketli Sınıflandırma Sonuçları 52
3.2.3.1 Yapay Sinir Ağı ile Sınıflandırma 55
3.2.3.2 Destek Vektör Makineleri ile Sınıflandırma 58
3.2.3.3 Hafif Gradyan Artırma Makineleri ile Sınıflandırma 60
3.3 Pima Indians Diabetes Veri Seti 63
3.3.1 Yapay Sinir Ağı ile Sınıflandırma 66
3.3.2 Destek Vektör Makineleri ile Sınıflandırma 68
3.3.3 Hafif Gradyan Artırma Makineleri ile Sınıflandırma 70
3.4 Sonuçların Karşılaştırılması 73
4. SONUÇLAR **75**
5. KAYNAKLAR **77**
6. ÖZGEÇMİŞ **81**

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1: Çok katmanlı yapay sinir ağı. 9
Şekil 2.2: İkili adım fonksiyonu. 11
Şekil 2.3: Doğrusal fonksiyon. 12
Şekil 2.4: Sigmoid fonksiyonu ve türevi. 12
Şekil 2.5: Hiperbolik tanjant fonksiyonu ve türevi. 13
Şekil 2.6: ReLU fonksiyonu ve türevi. 13
Şekil 2.7: Sızıntılı ReLU fonksiyonu ve türevi. 14
Şekil 2.8: Softmax fonksiyonu. 14
Şekil 2.9: Yapay sinir ağlarında ikili sınıflandırma. 15
Şekil 2.10: Yapay sinir ağlarında çok sınıflı sınıflandırma. 16
Şekil 2.11: Yapay sinir ağlarında çok etiketli sınıflandırma. 16
Şekil 2.12: Farklı boyutlarda sınıflandırma a) İki boyutlu sınıflandırma, b) Üç boyutlu sınıflandırma. 17
Şekil 2.13: İki sınıfı ayırabilecek olan doğrular. 18
Şekil 2.14: Maksimum marjinli doğrusal sınıflandırıcı. 18
Şekil 2.15: Esnek marjinli sınıflandırıcı. 20
Şekil 2.16: Giriş uzayını öznitelik uzayına taşıma. 23
Şekil 2.17: Seviye odaklı büyüme. 24
Şekil 2.18: Yaprak odaklı büyüme. 25
Şekil 3.1: Hedef değişkendeki sınıfların dağılımı. 42
Şekil 3.2: Veriye ait korelasyon matrisi. 42
Şekil 3.3: Çok sınıflı için kurulan temel modele göre değişkenlerin önem sırası. 44
Şekil 3.4: Çok sınıflı YSA eğitim ve doğrulama setleri için doğruluk grafiği. 46
Şekil 3.5: Çok sınıflı YSA eğitim ve doğrulama setleri için kayıp grafiği. 47
Şekil 3.6: Çok sınıflı YSA için ROC eğrileri. 47
Şekil 3.7: Çok sınıflı SVM için ROC eğrileri. 49
Şekil 3.8: LightGBM için değişkenlerin önem sırası. 51
Şekil 3.9: Çok sınıflı LightGBM için ROC eğrileri. 52
Şekil 3.10: Etiketlerdeki hasta sayısı dağılımı. 53
Şekil 3.11: Çok etiketli için kurulan temel modele göre değişkenlerin önem sırası. 54
Şekil 3.12: Çok etiketli YSA için eğitim ve doğrulama setlerinin doğruluk grafiği. 57
Şekil 3.13: Çok etiketli YSA için eğitim ve doğrulama setleri kayıp grafiği. 57
Şekil 3.14: Çok etiketli YSA için ROC eğrileri. 58
Şekil 3.15: Çok etiketli SVM için ROC eğrileri. 60
Şekil 3.16: LGBM için değişkenlerin önem sıralaması. 62
Şekil 3.17: Çok etiketli LightGBM için ROC eğrileri. 63
Şekil 3.18: Hedef değişkenin sınıf dağılımı. 64
Şekil 3.19: Veri setine ait korelasyon matrisi. 65

Şekil 3.20: Pima Indian Diabetes eğitim ve doğrulama setleri için doğruluk grafiği. 67
Şekil 3.21: Pima Indian Diabetes eğitim ve doğrulama setleri için kayıp grafiği. 67
Şekil 3.22: Yapay sinir ağına ait ROC eğrisi. 68
Şekil 3.23: Destek vektör makineleri için ROC eğrisi. 69
Şekil 3.24: Kategorik değişkenlerle oluşturulan modeldeki değişkenlerin önemi. 71
Şekil 3.25: İkili sınıflandırma LGBM için değişkenlerin önem sıralaması. 72
Şekil 3.26: İkili sınıflandırma LGBM için ROC eğrisi. 73

TABLO LİSTESİ

Sayfa

Tablo 2.1: Yapay sinir ağlarında sınıflandırma. 16
Tablo 2.2: Sınıflandırma tipleri ve özellikleri. 26
Tablo 2.3: İkili sınıflandırma karmaşıklık matrisi. 28
Tablo 2.4: Çok sınıflı sınıflandırma karmaşıklık matrisi. 30
Tablo 2.5: Alt küme doğruluğu örneği. 34
Tablo 3.1: Sonuç verilerine ait veri tipleri. 39
Tablo 3.2: Veri setindeki hastalıklara ait gözlem sayıları. 40
Tablo 3.3: Çok sınıflı sınıflandırma verisinde sınıflara ait gözlem sayıları. 41
Tablo 3.4: Korelasyon matrisinde yer alan değişkenler ve sıra numaraları. 43
Tablo 3.5: Çok sınıflı YSA sonuçları. 45
Tablo 3.6: Çok sınıflı YSA için karmaşıklık matrisi. 46
Tablo 3.7: Çok sınıflı SVM sonuçları. 48
Tablo 3.8: Çok sınıflı SVM için karmaşıklık matrisi. 49
Tablo 3.9: Çok sınıflı LightGBM sonuçları. 50
Tablo 3.10: Çok sınıflı LightGBM için karmaşıklık matrisi. 51
Tablo 3.11: Çok sınıflı sınıflandırma verisinde sınıflara ait gözlem sayıları. 53
Tablo 3.12: Çok etiketli YSA sonuçları. 55
Tablo 3.13: Çok etiketli YSA için karmaşıklık matrisi. 56
Tablo 3.14: Çok etiketli SVM sonuçları. 59
Tablo 3.15: Çok etiketli SVM için karmaşıklık matrisi. 59
Tablo 3.16: Çok etiketli LGBM sonuçları. 61
Tablo 3.17: Çok etiketli LGBM için karmaşıklık matrisi. 61
Tablo 3.18: Veri setine ait değişkenlerin açıklamaları. 64
Tablo 3.19: İkili sınıflandırma için YSA sonuçları. 66
Tablo 3.20: İkili sınıflandırma için YSA karmaşıklık matrisi. 66
Tablo 3.21: İkili sınıflandırma için SVM sonuçları. 68
Tablo 3.22: İkili sınıflandırma için SVM karmaşıklık matrisi. 69
Tablo 3.23: İkili sınıflandırma için LGBM sonuçları. 70
Tablo 3.24: İkili sınıflandırma için LGBM karmaşıklık matrisi. 70
Tablo 3.25: Çok sınıflı sınıflandırma verisi için model sonuçları. 73
Tablo 3.26: Çok etiketli sınıflandırma verisi için model sonuçları. 74
Tablo 3.27: İkili sınıflandırma verisi için model sonuçları. 74

KISALTMALAR LİSTESİ

AUC	**:**	Area Under Curve
EFB	**:**	Exclusive Feature Bundling
GOSS	**:**	Gradient One-Side Sampling
LGBM	**:**	Light Gradient Boosting Machines
ROC	**:**	Receiver Operating Characteristic
SVM	**:**	Support Vector Machine
YSA	**:**	Yapay Sinir Ağı

ÖNSÖZ

Yüksek lisans eğitimim boyunca tecrübelerini ve hayat kurtaran bilgilerini benimle paylaşan değerli danışman hocam Dr. Öğr. Üyesi Bedri BAHTİYAR'a, paylaştığı bilgilerle yolumu aydınlatan değerli hocam Prof. Dr. Serdar İPLİKÇİ'ye, gerek sektör gerek akademik bilgisiyle yardımıma koşan değerli hocam Dr. Öğr. Üyesi Şenay TOPSAKAL'a ve ilerlediğim bu yolda bana destek veren, yardımlarını esirgemeyen Arş. Gör. Sinem CEYLAN KONAK'a sonsuz teşekkür ederim.

Her zaman olduğu gibi üniversite hayatım boyunca da yanımda olan, her koşulda beni destekleyen, asla haklarını ödeyemeyeceğim babama, anneme, babaanneme, kardeşime ve minik kedime teşekkür ederim.

1. GİRİŞ

Hastalık teşhisi yapmak sağlık alanının en önemli ve en büyük problemlerden biridir. Bazı hastalıklarda görülen semptomların benzer olması sebebiyle hekimin hastaya teşhis koyması zorlaşabilmektedir. Bu aynı zamanda yanlış teşhis koyularak tedavinin yanlış yöntemlerle devam edilmesine sebep olabilmektedir. Erken teşhis ile tedavi süresi kısalabilir, kişinin hastalığının ilerlemesi engellenebilir. Artan nüfus sayısı ile sağlık sektöründeki karmaşıklık da artmaktadır. Artan iş yükü ve bunun yanında yetersiz kalan insan gücü, verilen sağlık hizmetinin veriminde düşüşlere yol açmaktadır. Hacim olarak sürekli büyümekte olan hasta verilerinin kullanılması sağlık alanındaki uygulamalarda hızlı bir şekilde aksiyon almayı sağlamaktadır. Bu amaçla yapay zeka ile hekimlerin hastalığı erken ve doğru teşhis edebilmelerine destek olacak çalışmalar yapılmaktadır.

Endokrinoloji alanında hastalık teşhisi diğer birçok hastalıkta olduğu gibi insan sağlığı açısından önemli bir konudur. Endokrinolojik hastalıklar, insan vücudunun işleyiş düzenini ve metabolizma sistemini etkilemektedir. Bu hastalıkların erken tespit edilmesi ve tedaviye başlanması hastanın vücut dengesinin bozulmasını en aza indirgeyecektir.

Sağlık kurum ve kuruluşlarında klinik araştırma, toplum verileri, sağlık kayıtları, çeşitli biyomedikal veriler ve elektronik sağlık kayıtları halinde çok büyük veriler birikmiş durumdadır. Makine öğrenmesi algoritma modellerini doğru bir şekilde eğitebilmek ve sağlık alanında etkili bir şekilde kullanabilmek için büyük miktarda verilerin kullanılması önemlidir. Toplanan verilerin temizlenip işlenmesi sonrasında makine öğrenmesi modeli uygulanmasıyla hastalık teşhisi, ilaç belirleme, tedavi süreci planlama gibi konularda sağlık çalışanlarına yardımcı olabilecek uygulamalar yapılmaktadır. Sağlık kurumları için maliyeti azaltmak, sağlık personelinin üzerindeki sorumlulukları hafifletebilmek ve bunların yanında tanı koymada daha yüksek doğruluk, tedavi planlamasında iyi tahminler yaparak hasta memnuniyetini artırmak için yapılan yapay zeka ve makine öğrenmesi çalışmalarının önemi günden güne artmaktadır.

Chen ve Pan (2018) ikilisinin yaptığı çalışmada lojistik, rastgele orman, yükseltme algoritmalarından olan Adaboost.M1 ve LogitBoost makine öğrenmesi modelleri ve 10 katlı çapraz doğrulama kullanılmıştır. Bu çalışma tek merkezli bir çalışmadır ve veriler Wenzhou Medical University'ye bağlı olan First Affiliated Hospital kurumunda toplanmıştır. Bu makine öğrenmesi modelleri klinik test verilerinden elde edilmiş 35.669 kişinin diyabet verilerini içeren bir veri seti üzerinde uygulanmıştır. Bu veri seti iki kısımdan oluşmaktadır. Birinci bölüm diyabet hastası olan kişilerin koyulmuş tanı ve tedavi verilerini, ikinci bölüm ise diyabet hastası olmayan kişilerin fiziksel muayenelerinden elde edilen verileri içermektedir. Çok iyi sınıflandırma yapabilme yeteneklerinden dolayı Adaboost.M1 ve LogitBoost modelleri kullanılmıştır. Çalışma lojistik 43 dakika 22 saniyede %85.35, rastgele orman 30 saniyede %91.55, AdaBoost.M1 1 saniyede % 92.6, LogitBoost 1 saniyede %93.93 doğrulukla gerçekleştirilmiştir. Bu veriler için lojistik ve rastgele ormanlar algoritmalarının hesaplama süresi ve sınıflandırma başarısı yükseltme algoritmaları kadar başarılı olamamıştır.

Lai ve diğ. (2019) tarafından yapılan çalışmada amaç vücudun glikozu metabolize edememesi üzerine oluşan diyabet hastalığına sahip hastaları belirlemek için yüksek duyarlılık ve yüksek seçiciliğe sahip bir makine öğrenmesi modeli oluşturmaktır. Bunun için Kanada popülasyonundaki hastaların demografik verilerine ve laboratuvar sonuçlarını kullanarak sınıflandırma işlemi yapmışlardır. Yaş, cinsiyet, açlık kan şekeri, vücut kitle indeksi, yüksek yoğunluklu lipoprotein, trigliseritler, kan basıncı ve düşük yoğunluklu lipoprotein gibi laboratuvar bilgilerini kullanarak ve Lojistik Regresyon, Rastgele Orman, Karar Ağacı, GBM (Gradient Boosting Machine) teknikleriyle tahminleme yapmışlardır. Veri setlerini 80:20 oranında ayırıp 10 katlı çapraz doğrulama uygulayıp model üzerinde hiper parametrelere ince ayar yapmışlardır. En iyi performansı GBM ve Lojistik Regresyonda elde etmişlerdir.

Nazhat ve Yağanoğlu (2021) çalışmalarında birçok insanda görülen diyabet hastalığını daha basit ve hızlı bir şekilde teşhis edebilmek için makine öğrenmesi algoritmalarını kullanmışlardır. Pima Indian Diabetes Dataset'ini kullanmışlardır. Veri setlerinde eksik değer olmamasına rağmen bazı özellikler anlamsız bir şekilde sıfır değerine sahip olduğu için özelliklerin kendi içlerinde ortalamasını ve ortanca

değerlerini bularak sıfır değerlerini değiştirmişlerdir. Normalizasyon işlemi sonrası veri setlerini 70:30 oranında ayırarak makine öğrenmesi modellerini uygulamışlardır. K-En Yakın Komşu (KNN), Rastgele Orman (RO), Destek Vektör Makinesi (DVM), Yapay Sinir Ağı (YSA) ve Karar Ağacı (KA) algoritmalarıyla çalışmalarını tamamlamışlardır. %88.31 başarısıyla en iyi sonucu Rastgele Orman algoritması tahminleme yapmıştır. Arkasından %86 başarıyla Yapay Sinir Ağları en iyi tahminlemeyi yapmıştır.

Akgül ve diğ. (2020) tarafından yapılan çalışmada açık kaynak olarak veri setleri yayınlanan UCI veri tabanında bulunan hipotiroidi hastalığına ait veri seti üzerinde çalışmışlardır. Bu veri setinde hedef değişkendeki sınıf dağılımının dengesiz olması sebebiyle birçok farklı veri örnekleme yöntemi uygulanmıştır. Çalışmada veri önişleme adımlarından sonra lojistik regresyon, k-en yakın komşu ve destek vektör makinesi algoritmaları kullanılmıştır. Geliştirilen modeller arasında en yüksek sonucu veren %97.8 oran ile lojistik regresyon olmuştur.

Pandey ve diğ. (2015) yaptıkları bu çalışmalarının sonucunu tiroit hastalıklarının sınıflandırılmasını daha önce yapılan çalışmaların sonuçlarıyla karşılaştırmışlardır. UCI Machine Learning Repository açık kaynak veri setlerinden tiroit veri setiyle çalışmışlardır. Sınıflandırma doğruluğunu ve performansı artırmak için özellik seçimi yöntemini seçmişlerdir. Çalışmada C4.5 ve rastgele orman algoritmalarını kullanarak %95 başarı elde etmişlerdir.

Li ve diğ. (2018) Zhejiang University kurumundaki bir sağlık muayenesine katılan 10508 kişinin anketlerini, laboratuvar testlerini, fiziksel muayenelerini ve karaciğer ultrasonografilerini kullanarak bir çalışma gerçekleştirmişlerdir. Alkolsüz yağlı karaciğer hastalığı tip 2 diyabet, kardiyovasküler hastalık, metabolik sendrom gibi çeşitli hastalıklara sebep olabilecek bir hastalıktır. Doğru bir şekilde ve erkenden teşhis edilmesi kritik derecede önemlidir. Çalışmada açık kaynak yazılımı olan Weka kullanılmıştır. Veri seti üzerinde öznitelik çıkarma işlemi, gereksiz görülen özellikler veri setinden çıkarma işlemi yapılmıştır. 10 katlı çapraz doğrulama ile geleneksel algoritmalar, topluluk algoritmaları ve algoritma uzantıları olmak üzere 3 ayrı algoritma ailesi kullanılmış. En iyi doğruluk değeri %83.41 ile lojistik regresyonda, %72.5 ile en iyi kesinlik değeri destek vektör makinesi algoritmasında elde edilmiştir.

Yıldız (2019) tarafından yapılan çalışmada iki ayrı veri grubu üzerinde tiroit hastalığını tespit etme çalışması yapılmıştır. Bu çalışmada Weka programı ve Matlab programı Yapay Sinir Ağları Toolbox'ı kullanılarak makine öğrenmesi modellerinin eğitim sonuçları elde edilmiştir. Birinci ve ikinci grup verileri üzerinde Weka programı ile Çok Katmanlı Algılayıcı (ÇKA) algoritması ve k-en yakın komşu algoritması, Matlab programı üzerinde Levenberg-Marquardt algoritması kullanılmış ve en düşük doğruluk oranı k-en yakın komşu, en yüksek doğruluk oranı ise çok katmanlı yapay sinir ağları ile elde edilmiştir.

AlKaabi ve diğ. (2020) tarafından çalışmada ilaç ve yaşam tarzı değişiklikleri ile tedavi edilebilen hipertansiyon hastalığı için üç denetimli makine öğrenmesi algoritması kullanılmıştır. 5 katlı çapraz doğrulama ile karar ağacı, rastgele ormanı lojistik regresyon algoritmaları üzerinde Weka ve Stata programını kullanarak tahminleme yapılmıştır. Veri setinde bulunan tüm değişkenler kategoriktir. Bu değişkenlerin ilişkilerini değerlendirebilmek için ki-kare testini uygulamışlardır. Makine öğrenmesi algoritmalarının veri seti üzerindeki başarılarını karşılaştırabilmek için doğruluk, hassasiyet, f-score, pozitif tahmin değeri, alıcı işletim karakteristik eğrisi altında kalan alan gibi metriklere bakılmıştır. En iyi performans doğruluk değeri %82.1, pozitif tahmin değeri %81.4, hassasiyet değeri %82.1 ile rastgele orman algoritması olmuştur.

Venkatesan ve Er (2014) çok-etiketli sınıflandırma için multimedya, metin, biyoloji alanlarında toplam 6 veri setine aşırı öğrenme makinelerini uygulamışlardır. Daha önceki çalışmalarda aşırı öğrenme makineleri yönteminin kullanılmadığı ifade edilmiştir. Çalışmadaki veri setleri etiket sayısı 6 etiketten 374 etikete kadar değişen etiket sayısına sahip veri setleridir. Çok-etiketli sınıflandırma için literatürde mevcut olan Algoritma Uyarlama (Algorithm Adaptation – AA), Problem Dönüşümü (Problem Transformation – PT) ve Topluluk Metodları (Ensemble Methods – EM) yöntemleri temelinde Destek Vektör Makineleri, Karar Ağaçları ve K-En Yakın Komşu makine öğrenmesi algoritmaları kullanılmıştır. Elde edilen sonuçlarla ELM'ye dayalı yöntemden elde edilen sonuçlar karşılaştırılmıştır. Sonuçları değerlendirmek için ise çok etiketli sınıflandırma metriklerini kullanmışlardır.

Zufferey ve diğ. (2015) tarafından kronik hastalıklara sahip hastaların bilgilerini içeren MIMIC-II klinik bir veri setine çok-etiketli sınıflandırma

algoritmaları uygulanmıştır. Kullandıkları veri seti 19773 kişi ve bu kişilerin 10 farklı kronik hastalığa ait bilgilerini içermektedir. Çalışmada temel sınıflandırıcı olarak SVM, DT ve Naive Bayes (NB), çok-etiketli sınıflandırıcı algoritmalarından ML-*k*NN, AdaBoostMH, binary relevance, classifier chains, HOMER ve RA*k*EL Java kütüphanelerinden MULAN ve WEKA kullanılmıştır.

Karakoyun ve Hacıbeyoğlu (2014), UCI veri deposundan ikili ve çok-sınıflı sınıflandırma üzerine çalışmak üzere 9 farklı hastalık için ayrı ayrı veri setleri ele almışlardır. Veri setlerine literatürde sık kullanılan makine öğrenmesi algoritmalarından Rastgele Ormanlar, Yapay Sinir Ağı, CN2, NB, KNN, SVM uygulanmıştır. Yapılan deneysel ve istatistiksel sonuçlara göre YSA daha yüksek başarı, KNN ise daha hızlı çalışma performansı göstermiştir.

Çok-etiketli sınıflandırma konusunda yapılan çalışmalar incelenildiğinde çalışmaların birçoğunda hastalara ait birden fazla hastalığı tespit etme amacı vardır. Zhou ve diğ. (2021) tarafından yapılan çalışmada diyabete sebep olan 4 diyabetik komplikasyonu aynı anda tahmin etmeyi amaçlanmıştır. Çin'deki Changzhou 2 Nolu Halk Hastanesine kabul edilen hastaların modern elektronik sağlık kayıtlarından (modern electronic health records - EHRs) demografik bilgileri ve laboratuvar verileri alınmıştır. Farklı diyabetik komplikasyonlar arasındaki korelasyonları analiz etmek için Pearson Korelasyon Katsayısı, tahminleme uygulaması için temel algoritma RF seçilmiş olup çok-etiketli sınıflandırma için geleneksel algoritmalar kullanılmış ve ikili ilişki (Binary Relevance -BR) sonucuyla karşılaştırılma yapılmıştır. Değerlendirme için hamming kaybı, doğruluk, f1-puanı, kesinlik, geri çarpma, f1-mikro, f1-makro, hassas mikro, hassas makro, geri çağırma mikro, geri çağırma makro ve alıcı işletim karakteristik eğrisi (AUROC) kullanılmıştır.

Li ve diğ. (2017) çalışmalarında hipertansiyon, diyabet, yağlı karaciğer, kolesistit, kalp hastalığı ve obezite olmak üzere 6 kronik hastalığa sahip bir veri seti kullanmışlardır. Veri setinde 110300 kişi ve bu kişilerin fiziksel muayene, rutin kan tetkikleri, karaciğer fonksiyon testleri ve doktorlar tarafından teşhisi konulmuş hastalıklar olmak üzere 62 özellik bulunmaktadır. Çalışmada (Ensemble Label Power-set Pruned datasets Joint Decomposition - ELPPJD) metodu önerilmiştir. İlk olarak çok-etiketli sınıflandırma problemini çok-sınıflı sınıflandırma problemine dönüştürmüşlerdir. Dengesiz öğrenme problemini çözebilmek için budanmış veri

setleri ve ortak ayrıştırma yöntemlerini önermişlerdir. Önerilen bu yöntem ile geleneksel çok-etiketli sınıflandırma algoritmalarından RAkEL ve HOMER ile karşılaştırma yapılmış ve ELPPJD yönteminden çok daha üstün bir performans elde edilmiştir.

Bromouri ve diğ. (2014) yaptıkları bir çalışmada kronik hastalıkları içeren iki gerçek hayat verisi üzerinde çok-etiketli sınıflandırma yapmak için kelime torbası (Bag of Words-BoW) ve denetimli boyut azaltma algoritmalarını birleştirmişlerdir. Kullanılan Portavita veri seti dengeli etiket dağılımı gösterir iken MIMIC II veri seti dengesiz etiket dağılımı gösteren bir veri setidir. Veriler çok değişkenli zaman serilerine dönüştürülmüştür. BoW tekniği öncesinde ve sonrasında standartlaştırma işlemleri yapılmış son olarak boyut azaltma algoritması uygulanmıştır. Portavita veri seti için kullanılan modelleme algoritmalarından Çekirdek Yerel Fisher Diskriminant Analizi (Kernel Local Fisher Discriminant Analysis-KLFDA) en iyi sonucu verirken MIMIC II veri seti için kullanılan boyut azaltma algoritmaları veri setinin dengesiz etiket dağılımından ötürü sonuçları kötü etkilemiştir.

Turhan ve diğ. (2020) yaptıkları çalışmalarında İzmir Bozkaya Eğitim ve Araştırma Hastanesi, Endokrinoloji ve Metabolizma Hastalıkları polikliniğine başvuran hastaların diyabet tanısının sınıflandırılması üzerine çalışmışlardır. Veri setindeki sınıf dengesizliği sebebiyle sentetik azınlık aşırı örnekleme (SMOTE), alt örnekleme (undersampling) ve aşırı örnekleme (oversampling) yöntemlerini kullanmışlardır. Elde ettikleri sonuçlara göre sınıf dengesizliği durumunda verinin dengelenmesi sonrası sınıflandırma algoritmalarının kullanılması gerektiğini öne sürmüşlerdir.

Literatürde yapılan çalışmalar incelenildiğinde endokrinolojik hastalıkları sınıflandırma üzerine birçok çalışma yapılmıştır. Bu çalışmaların bir kısmı ikili sınıflandırma bir kısmı çok-sınıflı sınıflandırma bir kısmı da çok-etiketli sınıflandırma ile yapılmıştır.

Bu tez çalışmasında iki farklı veri seti üzerinde çalışılmıştır. Birincisi Pamukkale Üniversitesi Hastanesi'ne ait veri tabanından endokrinoloji polikliniğine 2021 yılında başvurmuş kişilerin verileri alınarak çok-sınıflı ve çok-etiketli sınıflandırma çalışması gerçekleştirilmiştir. İkincisi literatürde çokça incelenmiş olan

Pima Indian Diabetes veri seti üzerinde ikili sınıflandırma gerçekleştirilmiştir. Tezin ikinci bölümünde yapay sinir ağları, destek vektör makineleri, hafif gradyan artırma makinesi (Light Gradient Boosting Machine-LightGBM-LGBM), makine öğrenmesinde uygulanan sınıflandırma çeşitleri açıklanıp ve bu çeşitlerin değerlendirme metrikleri anlatılmıştır. Üçüncü bölümde gerçekleştirilen çalışmanın detayları açıklanmış, dördüncü bölümde yapılan çalışmaların sonuçları değerlendirilmiştir.

2. MAKİNE ÖĞRENMESİ İLE SINIFLANDIRMA

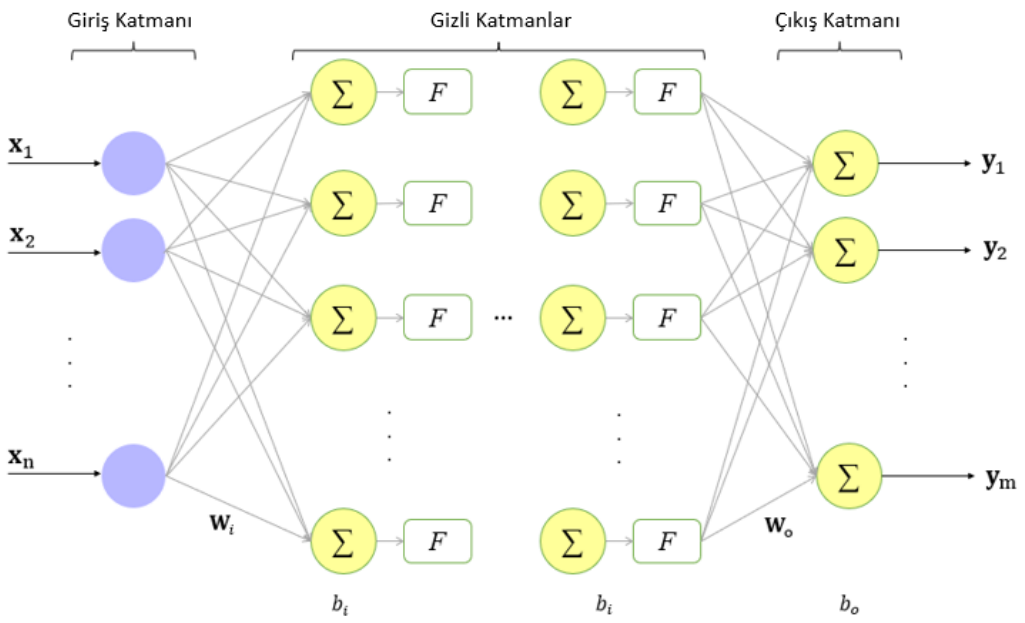
Makine öğrenmesi denetimli ve denetimsiz öğrenme olarak ikiye ayrılmaktadır. Bu bölümde denetimli öğrenme problemlerinden olan sınıflandırma ele alınmıştır. Kullanılan yöntemlerden olan yapay sinir ağları, destek vektör makineleri ve hafif gradyan artırma makineleri açıklanmıştır.

2.1 Yapay Sinir Ağları

Yapay sinir ağları (YSA), insan beyninden esinlenerek geliştirilmiş biyolojik sinir ağlarını taklit ederek veri kümeleri arasındaki ilişkiyi bulmaya çalışan bir yapıdır. Nöronlar, aldıkları girdi bilgisini elektrik sinyallerine dönüştürerek aldığı bilgiyi işlerler. Sonucu bir sonraki nörona iletirler. Sinir hücresindeki dendritler, önceki nöronun terminal düğmesinden veya sinapsından uyarı alır. Dendritler, uyarıyı sinir hücresinin çekirdeğine taşıyarak elektriksel darbe işlenir ve sonrasında aksona iletilir. Akson, uyarıyı sinir hücresi çekirdeğinden sinapsa taşıyan yapıdır. Sinaps gelen uyarıyı bir sonraki nöronun dendritlerine iletir. Bu şekilde bir nöron ağı oluşturmuş olmaktadır (Panchal 2018).

Yapay sinir ağları tek katmanlı ve çok katmanlı olacak şekilde modellenebilir. Tek katmanlı sinir ağı karmaşık problemleri çözmede yetersiz olduğu için giriş katmanı ile çıkış katmanı arasına problemin durumuna göre gizli katmanlar eklenerek Şekil 2.1'deki gibi çok katmanlı sinir ağı oluşturulur.

$\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ ifadeleri nöronun girişlerini, $\mathbf{W}_i$ve $\mathbf{W}_o$ ifadeleri ağırlıkları, b_i ve b_o ifadeleri yanlılıkları, F aktivasyon fonksiyonunu, h indisi gizli katman sayısını, k indisi 1'den m'ye kadar olan değerleri, $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m$ ifadeleri ise çıkış değerlerini temsil etmektedir.



Şekil 2.1: Çok katmanlı yapay sinir ağı.

Denklem (2.1)'de çok katmanlı yapay sinir ağının matematiksel formülü bulunmaktadır (Bagheri 2020).

$$\mathbf{Y}_k = \mathbf{W}_o F(\mathbf{W}_{ih} \dots F(\mathbf{W}_{i1}\mathbf{x}_i + b_{i1}) \dots + b_{ih}) + b_ç \tag{2.1}$$

Yapılan işlem her zaman giriş değerleri ile rastgele başlatılan ağırlıkların çarpılması, yanlılık değerinin eklenmesi ve aktivasyon fonksiyonunun uygulanmasıdır. Bu işlem ileri yayılım olarak adlandırılır. Uygulamanın sonucunda tahminleme yapılır ve gerçek değerler ile tahmin edilen değer arasındaki fark yani hatalar hesaplanır. Bu hatayı minimize etmek için geri yayılım yapılır. Geri yayılımda ağırlık ve yanlılık değerlerine göre kayıp fonksiyonunun gradyanları hesaplanarak bu değerler güncellenir. Bu güncelleme işlemi optimizasyon metotlarıyla gerçekleştirilir.

Sınıflandırma problemlerinde kayıp fonksiyonu olarak Denklem (2.2)'de görülen çapraz entropi formülü kullanılır (Yamini 2021).

$$\mathbf{J}(\theta) = \frac{1}{\mathrm{m}}[\sum_{i=1}^{m} -\mathrm{y}^{(\mathrm{i})}\log(\mathrm{h}_\theta(x^{(i)})) + (1-\mathrm{y}^{(\mathrm{i})})\log(1-\mathrm{h}_\theta(x^{(i)}))] \tag{2.2}$$

Burada m; gözlem sayısını, $y^{(i)}$; gerçek değerleri ve $h_\theta(x^{(i)})$; 1 sınıfına ait olma olasılıklarını temsil etmektedir.

Optimizasyon metotlarından biri gradyan azalan algoritmasıdır. Bu algoritma türevlenebilir bir fonksiyonu minimum yapabilecek parametre değerlerini bulur. Türevlenebilir fonksiyonun ilgili parametreye göre kısmi türevi hesaplanır. Elde edilen türev yani gradyan değeri ilgili fonksiyonun maksimum artış yönünü verir. Dolayısıyla Denklem (2.3)'te gösterildiği gibi ilgili fonksiyonun maksimum artış yönünü veren gradyanın tersine doğru belirli bir şiddet ile giderek parametrenin eski değerinde değişiklik yaparak her iterasyonda hatanın azalmasını sağlar (Garcia 2018).

$$\theta_j = \theta_j - \alpha \frac{\boldsymbol{\partial}}{\boldsymbol{\partial}\theta_j} \mathbf{J}(\theta) \tag{2.3}$$

Burada eşitliğin sol tarafındaki θ_j; güncellenmiş değer, sağ tarafındaki θ_j; ilk değer, α; öğrenme adımı, **J** ise türevlenebilir bir fonksiyondur.

Bir diğer optimizasyon yöntemi ise Adam optimizatörüdür. Adam yöntemi, gradyan azalma algoritmasının geliştirilmiş halidir. Çok sayıda veri ve parametre içeren karmaşık problemlerde bile daha az bellek gerektirdiği için çok verimli çalışır. Algoritma gradyanların üstel ağırlıklı ortalamasını ele alarak gradyan azalan algoritmasını hızlandırmaktadır. Yapılan işlemler boyunca yerel minimumları aşmak için büyük adımlar atarken küresel minimuma ulaştığında minimum salınım olacak şekilde gradyan azalma hızını kontrol etmektedir (GeeksforGeeks 2020).

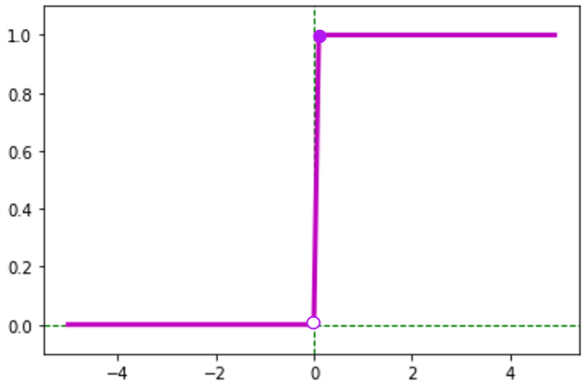
2.1.1 Aktivasyon Fonksiyonları

Aktivasyon fonksiyonları bir nöronun aktive edilip edilmeyeceğini karar verir. Aktivasyon fonksiyonlarının doğrusal ve doğrusal olmayan fonksiyon tipleri vardır. Bu fonksiyonların görevi, düğümden alınan toplam ağırlıklı girdiyi bir sonraki gizli katmana veya çıktı değerine dönüştürmektir. Aktivasyon işlemi uygulanmadığında çıkış değerimiz basit bir doğrusal fonksiyon şekildedir yani tek dereceli polinomdur. Sinir ağına kaç tane gizli katman eklendiği önemli değildir.

Çünkü eklenen tüm gizli katmanlar aynı şekilde davranacaktır. İki doğrusal fonksiyonun bileşimi yine doğrusal bir fonksiyon verecektir. Bu durum sinir ağının karmaşık bilgileri öğrenmesinde bir engeldir. Bunun için doğrusal olmayan fonksiyonlara da ihtiyaç vardır (Baheti 2022).

Denklem (2.4)'te yer alan ikili adım fonksiyonu, bir nöronun aktive edilip edilmeyeceğine karar veren bir eşik değere bağlıdır. Eğer girdi eşik değerinden büyükse nöron aktive edilir, eşik değerinden küçük ise nöron aktive edilmez yani çıktı bir sonraki gizli katmana iletilemez. Şekil 2.2'de görüldüğü üzere ikili değer alan bir fonksiyon olduğu için ikili sınıflayıcı olarak kullanılır. Bu sebeple genellikle çıkış katmanlarında kullanılır. Bu fonksiyonun türevi olmadığı için gradyanı sıfırdır ve bu durum geri yayılım sürecinde bir engeldir (Baheti 2022).

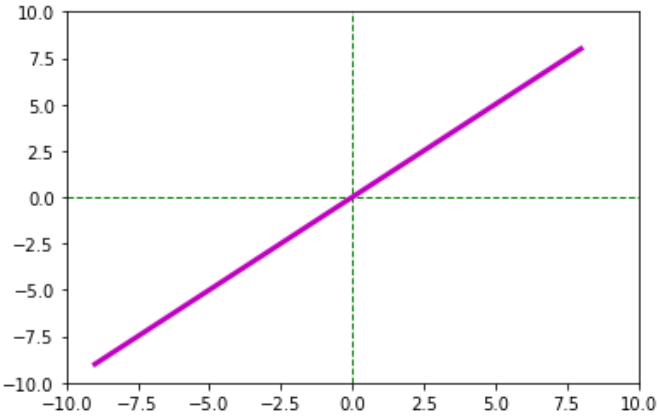
$$F(\mathbf{x}) = \begin{cases} 0, & eğer\ \mathbf{x} < 0 \\ 1, & eğer\ \mathbf{x} \geq 0 \end{cases} \tag{2.4}$$



Şekil 2.2: İkili adım fonksiyonu.

Şekil 2.3'te yer alan doğrusal fonksiyonunun çıkışı girdilerin k sabitiyle çarpılmasıyla elde edilir. Denklem (2.5)'te yer alan ifadenin $\mathbf{x}$'e göre türevi alındığında k değerini elde ederiz. Fonksiyonun türevi sabit bir değer olacağından ve $\mathbf{x}$ ile hiçbir ilişki olmadığından öğrenme işlemi gerçekleşmez. Eğer tüm katmanlarda doğrusal fonksiyon kullanılırsa sinir ağının tüm katmanları tek bir katmana düşecektir. Katman sayısı ne olursa olsun son katman ilk katmanın doğrusal bir işlevi olacaktır (Sharma 2017).

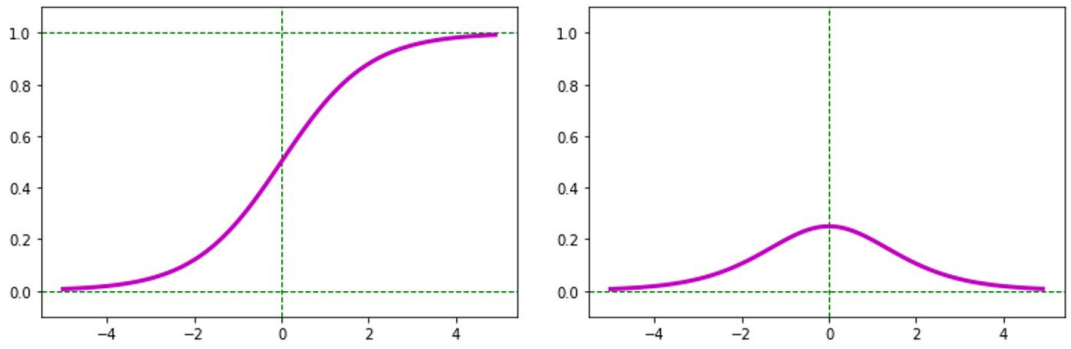
$$F(\mathbf{x}) = k\mathbf{x} \tag{2.5}$$



Şekil 2.3: Doğrusal fonksiyon.

Denklem (2.6)'da yer alan sigmoid fonksiyonu herhangi bir girdi değerine karşılık (0,1) aralığında bir çıktı verir. Girdi değeri ne kadar büyükse çıktı 1'e, girdi ne kadar küçük olursa çıktı 0'a o kadar yakın olur. Sigmoid fonksiyonu türevlenebilirdir. Ancak **x**'in değerine karşılık y değerinin gösterdiği değişikliğin az olduğu grafikten görülebilmektedir. Bu noktalarda çok küçük gradyanlara sahip olacağı için türev değerleri 0'a yakınsayacaktır ve gradyanların ölmesine sebep olacaktır. Bu sebeple de sinir ağı öğrenmeyi durduracaktır. Sigmoid fonksiyonu Şekil 2.4'te yer almaktadır (Bag 2021).

$$F(\mathbf{x}) = \frac{1}{1 + \mathrm{e}^{-\mathbf{x}}} \tag{2.6}$$

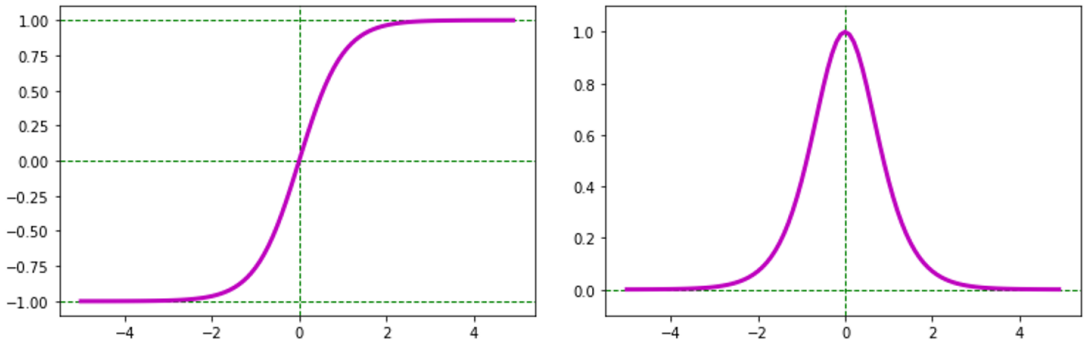


Şekil 2.4: Sigmoid fonksiyonu ve türevi.

Denklem (2.7)'de yer alan (-1, +1) aralığındaki çıkış değerine sahip hiperbolik tanjant fonksiyonu ile sigmoid fonksiyonuna benzer bir yapıya sahiptir. Şekil 2.5'te yer alan hiperbolik tanjantın türevi sigmoide göre daha diktir. Bu sebeple

daha çok değer alabilmesi söz konusudur. Sigmoid gibi gradyanların ölmesi problemi bu fonksiyonda da yer almaktadır (Baheti 2022).

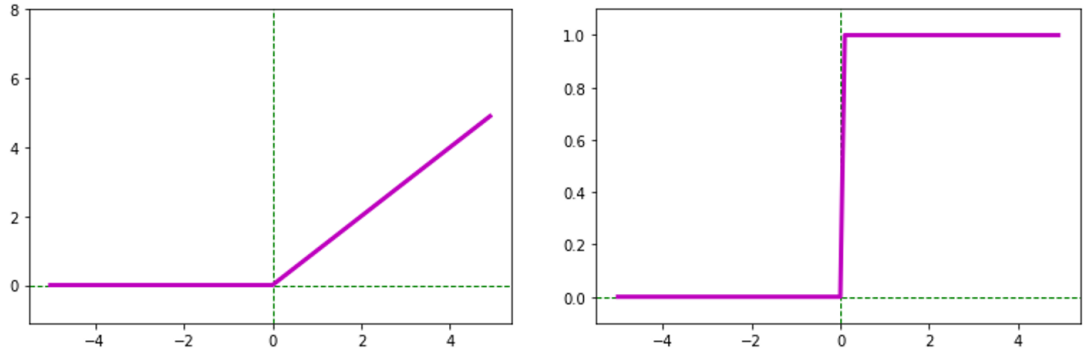
$$F(\mathbf{x}) = \frac{\mathrm{e}^{\mathbf{x}} - \mathrm{e}^{-\mathbf{x}}}{\mathrm{e}^{\mathbf{x}} + \mathrm{e}^{-\mathbf{x}}} \tag{2.7}$$



Şekil 2.5: Hiperbolik tanjant fonksiyonu ve türevi.

Denklem (2.8)'de yer alan ReLU fonksiyonu doğrusal fonksiyon izlenimi vermesine rağmen doğrusal değildir. Türev işlevine sahiptir ve geri yayılıma izin verir. ReLU fonksiyonu tüm nöronların aynı anda aktive olmamasına sebep olur. Nöronlar, çıktı 0'dan küçük ise devre dışı bırakılır. Belirli sayıda nöron aktive ettiği için sigmoid ve hiperbolik tanjanta göre daha verimlidir. ReLU fonksiyonunun dezavantajı grafiğin negatif tarafının gradyan değerinin sıfır olmasıdır. ReLU fonksiyonu ve türevi Şekil 2.6'da yer almaktadır (Sharma 2017).

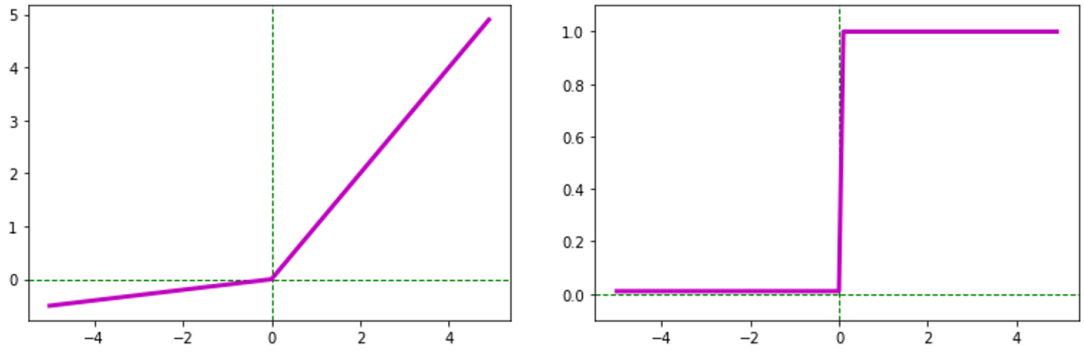
$$F(\mathbf{x}) = \max(0, \mathbf{x}) \tag{2.8}$$



Şekil 2.6: ReLU fonksiyonu ve türevi.

Denklem (2.9)'da yer alan sızıntılı ReLU fonksiyonu, ReLU fonksiyonundaki gradyan ölme problemini ortadan kaldırmak için oluşturulmuştur. Sızıntı değeri 0.1 olarak verilir. Negatif giriş değerleri için gradyan sıfırdan farklı bir değer döndürmektedir. Bu sayede bu bölgede ölü nöronlarla karşılaşılmaz. Sızıntılı ReLU fonksiyonu ve türevi Şekil 2.7'de yer almaktadır (Sharma 2017).

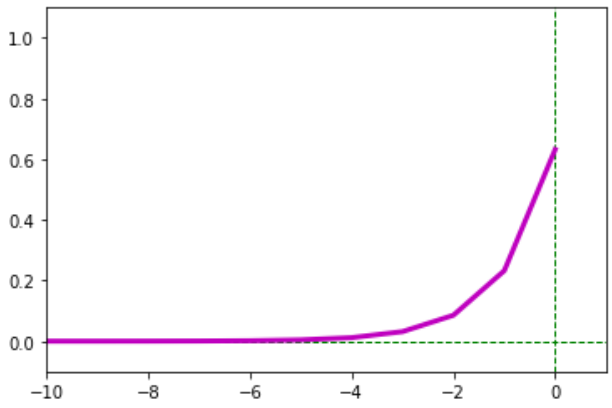
$$F(\mathbf{x}) = \max(0.1\mathbf{x}, \mathbf{x}) \tag{2.9}$$



Şekil 2.7: Sızıntılı ReLU fonksiyonu ve türevi.

Denklem (2.10)'da yer alan softmax fonksiyonu çoklu sigmoid fonksiyonu olarak tanımlanır. Şekil 2.8'de yer alan softmax fonksiyonu sigmoid fonksiyonunda olduğu gibi sınıflandırıcı olarak kullanılır. Sigmoid fonksiyonundan farkı ikiden fazla sınıf olduğunda her bir sınıfın olma olasılığını döndürmesidir. Bu sebeple çok sınıflı sınıflandırma durumunda sinir ağının son katmanda kullanılır (Bag 2021).

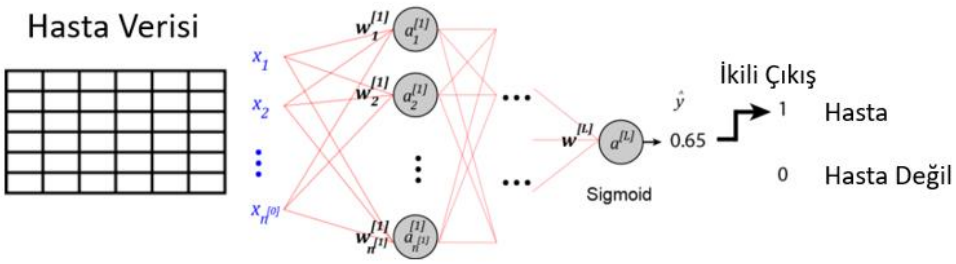
$$softmax(\boldsymbol{x_i}) = \frac{\mathbf{e}^{\mathbf{x}_i}}{\sum_{\boldsymbol{j=1}}^{\boldsymbol{N}} e^{\mathbf{x}_j}} \tag{2.10}$$



Şekil 2.8: Softmax fonksiyonu.

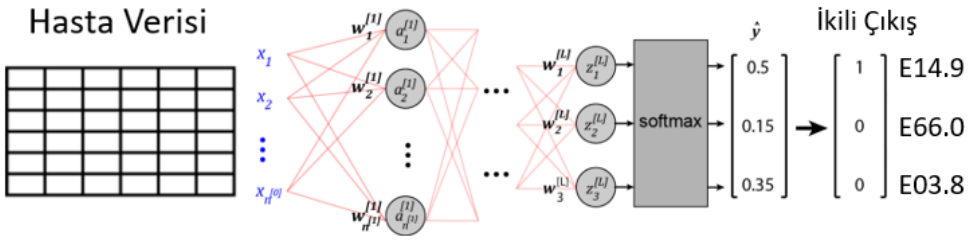
2.1.2 Yapay Sinir Ağlarında Sınıflandırma

İkili sınıflandırma probleminde hedef değişken uzayında iki sınıf bulunmaktadır ve bu iki sınıf birbirini dışlar. Çıkış değerlerinin olasılık değeri [0,1] aralığında olur. Bu durumda, yapay sinir ağının çıkış katmanında sigmoid fonksiyonu kullanılmalıdır. Aktivasyon fonksiyonları sürekli çıktı verdiği için varsayılan değer olarak 0.5 değeri eşik değeri olarak tanımlanır. Şekil 2.9'da gösterildiği gibi örneğin sınıfın çıktı değeri 0.5'e eşit veya küçükse sınıf 0'a, 0.5'ten büyük ise sınıf 1'e ait olduğu anlamına gelir (Bagheri 2020).



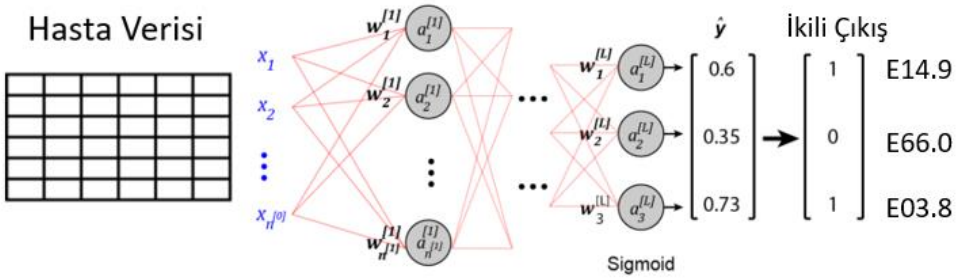
Şekil 2.9: Yapay sinir ağlarında ikili sınıflandırma.

Çok sınıflı sınıflandırma probleminde hedef değişken uzayında ikiden fazla sınıf vardır ve bu sınıflar birbirini dışlar. Her bir nöronun aktivasyonu sonucu, girdinin belirli bir sınıfa ait olup olmadığının göstergesidir. Tek sıcak kodlanmış çıktı vektörümüzde her bir sınıfın toplamı her zaman bire eşittir. Bu sınıflardan biri bile 1 olduğunda diğerlerini 0 olmaya zorlar. Çıktı katmanında sigmoid aktivasyonuna sahip nöronlar tarafından üretilen olasılıklar bağımsızdır ve olasılıkların toplamı bir ile sınırlı değildir. Bu sebeple çok sınıflı sınıflandırma probleminde her nörona bağımsız aktivasyon fonksiyonu uygulanmaz. Şekil 2.10'da görüleceği üzere tüm nöronların net girdileri bir softmax aktivasyon fonksiyonuna gider ve elde etmek istediğimiz şekilde sınıflara ait olma olasılıkları elde edilir (Bagheri 2020).



Şekil 2.10: Yapay sinir ağlarında çok sınıflı sınıflandırma.

Çok etiketli sınıflandırma probleminde hedef değişken uzayında ikiden fazla sınıf vardır ve sınıflar birbirini dışlamazlar. Aslında her sınıf bir etikettir. Her bir nöronun aktivasyonu sonucu, girdinin belirli bir sınıfa ait olup olmadığının göstergesidir. Çıktı katmanı sınıf sayısı kadar sigmoid aktivasyon fonksiyonuna sahip olmalıdır. Şekil 2.11'deki gibi aktivasyon fonksiyonu çıktısı olan vektörün çoklu sıcak kodlanmış bir vektöre dönüşebilmesi için 0.5 eşik değeri kullanılabilir (Bagheri 2020).



Şekil 2.11: Yapay sinir ağlarında çok etiketli sınıflandırma.

Tablo 2.1'de sınıflandırma problemlerinde yapay sinir ağlarının kullanımına dair tipik bir mimarisi özet şeklinde bulunmaktadır (Gèron 2019).

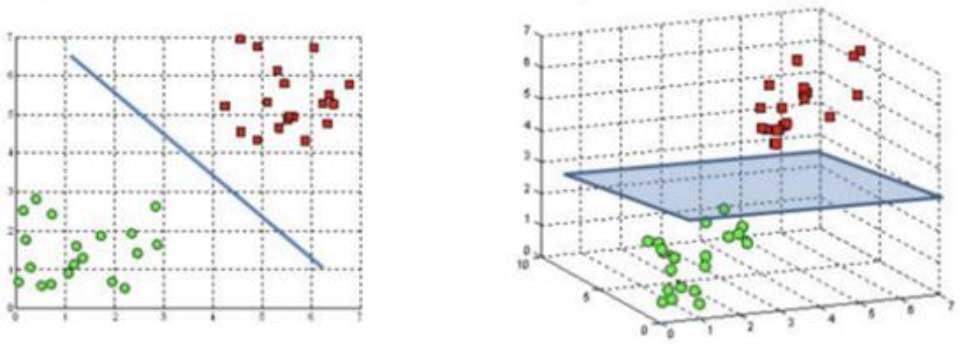
Tablo 2.1: Yapay sinir ağlarında sınıflandırma.

Hiper Parametre	**İkili Sınıflandırma**	**Çok Sınıflı Sınıflandırma**	**Çok Etiketli Sınıflandırma**
Girdi katmanı	Özellik başına 1	Özellik başına 1	Özellik başına 1
Gizli katmanlar	Probleme bağlı	Probleme bağlı	Probleme bağlı
Çıktı sinir hücreleri	1	Etiket başına 1	Sınıf başına 1
Çıktı katmanı aktivasyonu	Sigmoid/Lojistik	Sigmoid/Lojistik	Softmax
Kayıp Fonksiyonu	Çapraz entropi	Çapraz entropi	Çapraz entropi

2.2 Destek Vektör Makineleri

Destek vektör makineleri hem sınıflandırma hem de regresyon problemleri için kullanılabilen denetimli öğrenme algoritmasıdır. Sınıflandırma probleminde ana amaç en iyi ayırıcı karar sınırını bulmaktır. Destek vektör makineleri, doğrusal ve doğrusal olmayan sınıflandırma problemlerini gerçekleştirme yeteneğine sahiptirler (Cortes ve diğ. 1995). Küçük ve orta boyutlu karmaşık veri setlerini sınıflandırmaya uygundurlar. Problemin çözümünde probleme göre çekirdek fonksiyonları ve parametre ayarlaması yapılabilir (Gèron 2019).

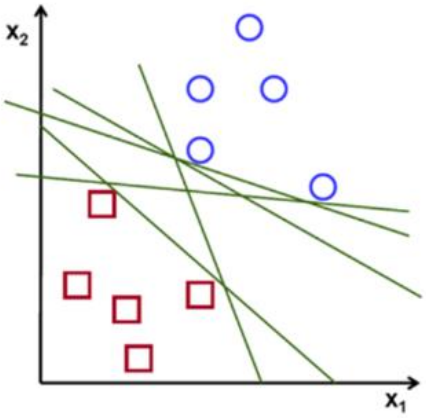
İki boyutlu uzayda yer alan veriler bir doğru aracılığıyla rahatça ayrılabilirler. Veriler daha yüksek boyutlu uzayda yer aldığı zaman sınıflandırmayı bir doğru ile gerçekleştirmek zordur. Bu durumda doğru kullanmak yerine hiper düzlem tercih edilir. Şekil 2.12a'da iki boyutlu ve 2.12b'de üç boyutlu sınıflandırma görülmektedir (Ippolito 2019).



Şekil 2.12: Farklı boyutlarda sınıflandırma a) İki boyutlu sınıflandırma, b) Üç boyutlu sınıflandırma.

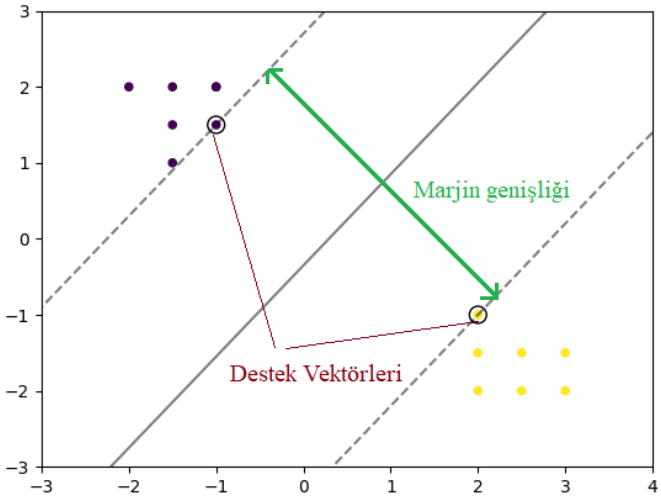
2.2.1 Doğrusal Sınıflandırma

Sınıflandırma problemi için basit bir doğrusal destek vektör makinesinin amacı iki sınıf arasına düz bir çizgi çekmektir. Bu iki sınıf arasına çizilecek olan doğru Şekil 2.13'teki gibi sonsuz sayıda farklı şekilde çizilebilir. Bu sebeple çizilecek olan bu doğrunun, iki sınıfın veri noktalarına maksimum uzaklıkta olması amaçlanır (Ippolito 2019).



Şekil 2.13: İki sınıfı ayırabilecek olan doğrular.

Doğrusal SVM sınıflandırma yapabilmek için Şekil 2.14'teki gibi iki sınıfı ayıran bir doğru çizer. Bu çizilen doğrunun +1 ve -1 uzaklığında kalan bölgeye marjin adı verilmektedir (Rasmussen 2022).



Şekil 2.14: Maksimum marjinli doğrusal sınıflandırıcı.

Marjinin genişliği sınıfların ne kadar başarılı birbirinden ayrıştırılabildiğini gösterir. Marjini maksimum yapan ve sınıfları ayırabilen hiper düzlemin denklemi Denklem (2.11)'deki gibi tanımlanabilir.

$$\mathbf{W}^{\mathbf{T}}\mathbf{x} + b = 0 \tag{2.11}$$

Burada **W** ağırlık vektörünü (hiper düzlemin normalini), b skaler değeri yanlılık sapma değerini ifade etmektedir. Bu parametrelerin alacakları değerler ile hiper düzlemin pozisyonu belirlenmiş olur. N elemandan oluşan bir veri seti $\{x_i, y_i\}$

şeklinde temsil edilsin. Buradaki $\mathbf{x_i}$ özellikler vektörü, y_i hedef değerleridir. y_i, +1 ya da -1 değeri alabilmektedir ve her bir $\mathbf{x_i}$ vektörü bir y_i sınıf değerine sahiptir. Veriler, y_i = +1 veya y_i = -1 olarak etiketlenmesiyle sınıflandırılmış olur. Sınıf etiketleri Denklem (2.12)'de belirtilmiştir.

$$\hat{y} = \begin{cases} 1, & \mathbf{W^T x} + b \geq 1 \\ -1, & \mathbf{W^T x} + b \leq -1 \end{cases} \tag{2.12}$$

Denklem (2.13)'teki ifadeler Denklem (2.14)'te olduğu gibi tek bir eşitlik ile ifade edilebilir.

$$y_i . (\mathbf{W^T} . \mathbf{x}_i - b) \geq 1 \tag{2.13}$$

Denklem (2.14)'teki koşulu sağlayan hiper düzlemin her iki tarafındaki destek vektörlerinin hiper düzleme olan uzaklıkları toplamı bize marjin değerini vermektedir. Marjin değerini maksimum yapan ise en optimal hiper düzlemdir. En uygun hiper düzlemi bulabilmek için **W** ve *b* değerlerinin belirlenmesi gerekir.

İki paralel doğru arasındaki uzaklık hesabı ile marjin değeri elde edilebilir.

$$\begin{gathered} d = \frac{|\boldsymbol{W^T x}^+ + b|}{||\boldsymbol{W}||} - \frac{|\boldsymbol{W^T x}^- + b|}{||\boldsymbol{W}||} \\ d = \frac{|1|}{||\mathbf{W}||} - \frac{|-1|}{||\mathbf{W}||} = \frac{2}{||\mathbf{W}||} \end{gathered} \tag{2.14}$$

Denklem (2.15)'te $\mathbf{x}^+$pozitif tarafta bulunan bir noktayı ve $\mathbf{x}^-$negatif tarafta bulunan bir noktayı temsil etmektedir. Elde edilen *d* değeri bize marjin genişliğini verir. Bu genişliğin maksimum değerde olabilmesi için $||\mathbf{W}||$ değerinin minimum olması gerekir. Bu durumda çözülmesi gereken bir optimizasyon problemi vardır.

$$\begin{gathered} \min \frac{1}{2} \mathbf{W^T} . \mathbf{W} \\ kısıtlar\ y_i (\mathbf{W^T x}_i + b) \geq 1 \end{gathered} \tag{2.15}$$

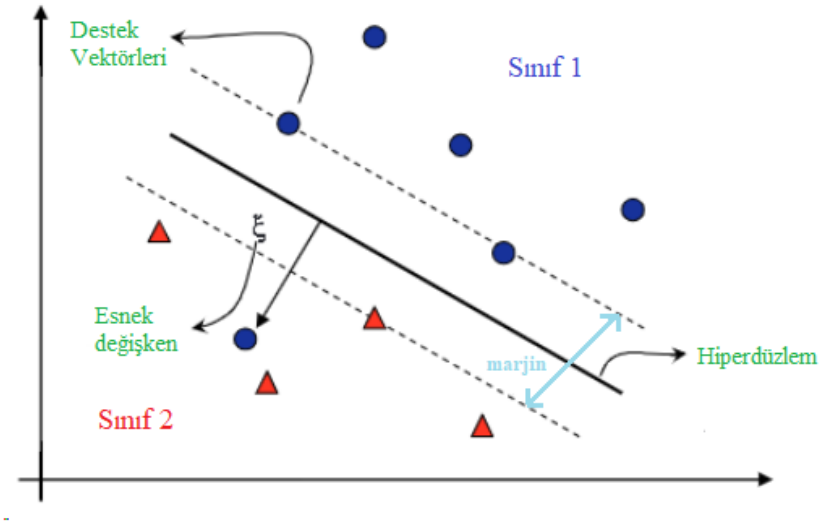
Denklem (2.15)'te yer alan problem bir karesel programlama problemidir. Burada amaç fonksiyonumuz, minimizasyon gerçekleştirileceği için dış bükey (konveks) olacaktır. Minimize etme problemi, karesel amaç fonksiyonuna ve

doğrusal bir kısıta sahip olduğu için Lagrange çarpanları kullanılarak çözümü yapılacaktır (Zhu 2020).

Denklem (2.15)'te formül sert marjinli sınıflandırıcıya aittir. Verilerin tamamen ayrılabilir olması durumunda kullanılır. Bu sınıflandırıcı aykırı değerlere karşı çok duyarlıdır. Bu sebeple sert marjinli sınıflandırıcıda yanlış sınıflandırmaya izin yoktur. Bazı durumlarda esnek marjinli sınıflandırıcı kullanılması tercih edilir.

2.2.1.1 Esnek Marjinli Sınıflandırma

Önceki bölümde belirtildiği gibi marjinin maksimum olması SVM'nin temel amaçlarındandır. Fakat bazı durumlarda sınıflardan birine ait örnekler diğer sınıfın örneklerine çok yakın olabilmektedir. Bu durum marjinin küçük ayarlanmasına sağlar. Marjinin küçük olması da modelin veri üzerinde genelleme yeteneği olumsuz etkilemektedir. Esnek marjinli sınıflandırıcı, sert marjinli sınıflandırıcı gibi sıfır hata yerine bazı örneklerin karşı sınıfın tarafında yer almasına izin vermektedir (Cortes ve diğ. 1995). Şekil 2.15'te esnek marjinli sınıflandırıcı görülmektedir (Lai 2008).



Şekil 2.15: Esnek marjinli sınıflandırıcı.

Sınıfları birbirinden ayıran hiper düzlem için gerekli denklemler Denklem (2.12)'de yer verilmiştir. Bu denklemler esnek değişken ξ_i kullanılarak düzenlenir ve Denklem (2.16)'da verilen denklemler elde edilir.

$$(\mathbf{W^T}.\mathbf{x}_i - b) \geq 1 - \xi_i$$
$$(\mathbf{W^T}.\mathbf{x}_i - b) \geq -1 + \xi_i \tag{2.16}$$

Denklem (2.16)'da yer verilen denklemler birleştirilirse Denklem (2.17)'de elde edilir.

$$\boldsymbol{y_i}.(\mathbf{W^T}.\mathbf{x_i} + \boldsymbol{b}) - \mathbf{1} + \boldsymbol{\xi_i} \geq \mathbf{0} \tag{2.17}$$

Burada dikkat edilmesi gerekilen konulardan biri esneklik payıdır. Çünkü esneklik payının çok artması sınıflandırmada yanlış yapılan örneklerin sayısının artmasına sebep olacaktır. Bu sebeple yeni bir kısıtlamamız olacaktır. Bu kısıtlama hatalı örneklerin sayısının olabildiğinde az olmasını sağlayacak Denklem (2.18)'de yer alan formüldür (Gèron 2019).

$$\min_{\text{w,b}} \frac{1}{2}\mathbf{W^T W} + C\sum_{i=1}^{n}\xi_i$$
$$kısıtlar\ y_i(\mathbf{W^T}\mathbf{x}_i + b) \geq 1 - \xi_i\ ve\ \xi_i \geq 0, \qquad i = 1,2,\dots,n \tag{2.18}$$

Burada yer alan ξ_i, i. örneğin marjı ne kadar ihlal ettiğini ifade eder. Esnek marjinin kullanılmasının iki amacı vardır. İlki sınır ihlallerini azaltmak için gevşek değişkeni yeterince küçük tutmak ikincisi ise marjini büyük tutmak için $\frac{1}{2}\mathbf{W^T W}$ değerini olabildiğince küçük tutmaktır. Denklem (2.18)'de yer alan C değeri esnek marjinli sınıflandırıcı için bir hiper parametredir. Bu hiper parametre amaçlar arasındaki dengeyi sağlar (Gèron 2019).

Problem Denklem (2.19)'da yer alan Lagrange fonksiyonunun çözülmesiyle çözülecektir. Problemin primal formu olan Denklem (2.19)'da yer alan α ve β ifadeleri Lagrange çarpanlarıdır.

$$\boldsymbol{\mathcal{L}}(\mathbf{W}, b, \xi, \alpha, \beta) = \frac{1}{2}\mathbf{W^T W} + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left[y_i\left(\mathbf{W^T}\mathbf{x}_i + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{n}\beta_i\xi_i \tag{2.19}$$

Problemin çözümü içim Karush-Kuhn-Tucker (KKT) koşullarını sağlayan çözümler gereklidir. Lagrange fonksiyonunu minimize edebilmek için $\mathbf{W}, b, \xi$ parametrelerine göre Denklem (2.20)'deki görüldüğü üzere kısmi türevleri alınır.

$$\frac{\partial}{\partial \mathbf{W}} \mathcal{L}(\mathbf{W}, b, \xi, \alpha, \beta) = 0 \Rightarrow \mathbf{W} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$
$$\frac{\partial}{\partial \mathrm{b}} \mathcal{L}(\mathbf{W}, b, \xi, \alpha, \beta) = 0 \Rightarrow 0 = \sum_{i=1}^{n} \alpha_i y_i \tag{2.20}$$
$$\frac{\partial}{\partial \xi_i} \mathcal{L}(\mathbf{W}, b, \xi, \alpha, \beta) = 0 \Rightarrow 0 = C - \alpha_i - \beta_i$$

Denklem (2.20)'de yer verilen denklemler Lagrange fonksiyonunda yerlerine konulmuş hali ve kısıtları Denklem (2.21)'de yer verilmiştir. Bu denklem dual en iyileme problemidir ve sıfır hariç α çarpanları için $\mathbf{x}_i$ örnekleri destek vektörleridir (Singh 2019).

$$\max \mathcal{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{i=1}^{n} y_i \, y_j a_i a_j \mathbf{x}_i \mathbf{x}_j$$
$$kısıtlar \;\; 0 \leq a_i \leq C, i = 1, \ldots, n \tag{2.21}$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

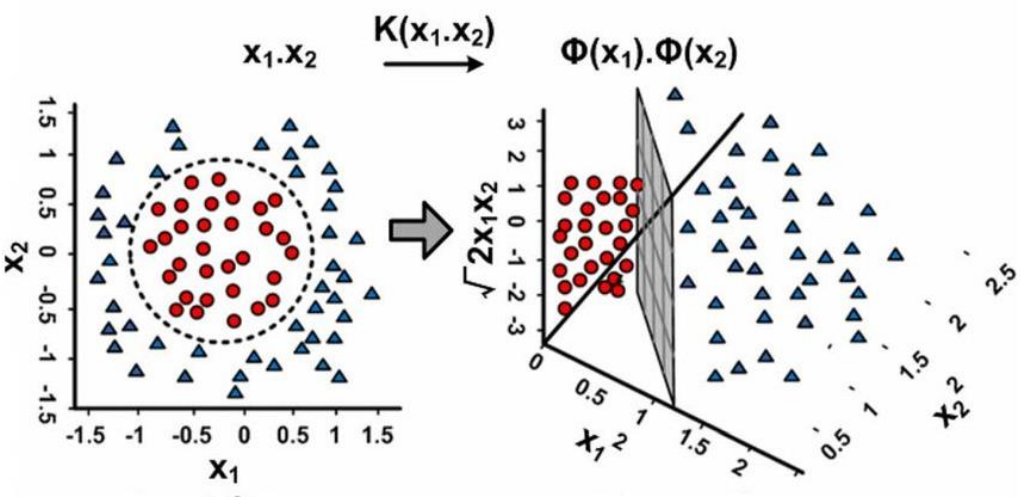
2.2.2 Doğrusal Olmayan Sınıflandırma

Gerçek hayat problemleriyle karşılaşıldığı durumlarda doğrusal sınıflandırıcılar yetersiz kalabilir. Bu sebeple bu tür problemlerde doğrusal olmayan karar yüzeyleri kullanılabilir. Doğrusal olmayan sınıflandırma yapılırken veri setindeki örnekler çekirdek fonksiyonu ile daha yüksek boyutlu ve doğrusal olarak ayrılabilecek bir uzaya taşınır ve çözüm bu yeni uzayda aranır (Khosla 2019).

Veri örneklerini girdi uzayından öznitelik uzayı olan $\mathcal{H}$ Euclid uzayına taşıyan $\mathbf{\Phi}$ fonksiyonu Denklem (2.22)'de verilmiştir (Uğuz 2019).

$$\mathbf{\Phi}: \mathfrak{R}^n \longrightarrow \mathcal{H} \tag{2.22}$$

Şekil 2.16'da giriş uzayında veri örneklerinin vektörleri çekirdek fonksiyonları sayesinde özellik uzayına aktarılmış hali bulunmaktadır.



Şekil 2.16: Giriş uzayını öznitelik uzayına taşıma.

Denklem (2.23)'te yer alan K fonksiyonuna çekirdek (kernel) adı verilir. Doğrusal olmayan SVM sınıflandırıcısı ise Denklem (2.24) ile ifade edilir. Bir fonksiyonun çekirdek fonksiyon olabilmesi için fonksiyonun sürekli, simetrik ve pozitif yarı tanımlı olması gereklidir (Mercer 1909).

$$K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\Phi}(\mathbf{x}_i)\boldsymbol{\Phi}(\mathbf{x}_j) \tag{2.23}$$

$$\boldsymbol{f}(\mathbf{x}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \alpha_i y_i \boldsymbol{\Phi}(\mathbf{x}_i)\boldsymbol{\Phi}(\mathbf{x}_j) + b \tag{2.24}$$

Çekirdek fonksiyonu olarak Denklem (2.25)'te yer alan doğrusal çekirdek fonksiyonu, Denklem (2.26) polinom çekirdek fonksiyonu ve Denklem (2.27) radyal tabanlı çekirdek fonksiyonu (RBF) kullanılabilir (Uğuz 2019).

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j) \tag{2.25}$$

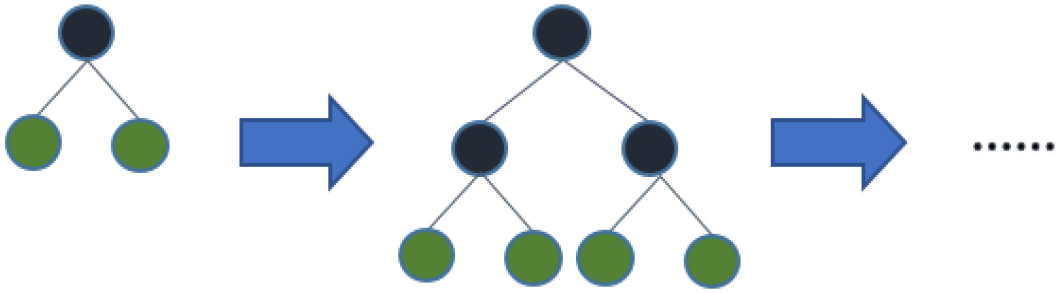
$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j) \tag{2.26}$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{e}^{\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{2\sigma^2}\right)} \tag{2.27}$$

2.3 LightGBM

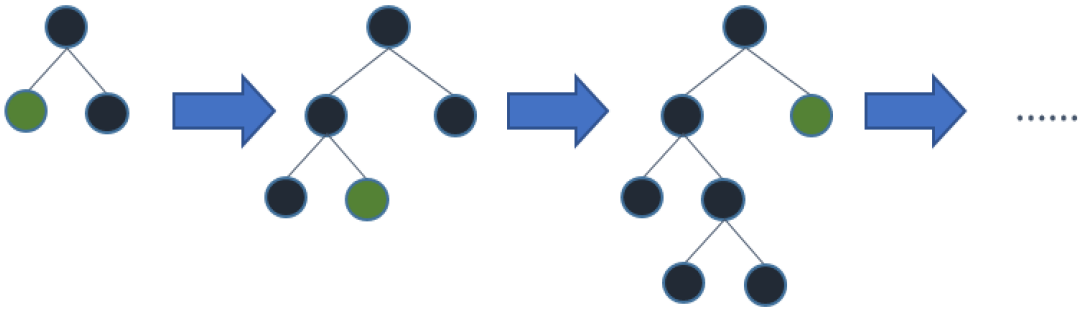
LightGBM, ağaç tabanlı öğrenme algoritmalarını kullanan bir gradyan artırma çerçevesidir. 2017 yılında Microsoft DMTK (Distributed Machine Learning Toolkit) projesi kapsamında geliştirilmiş bir algoritmadır. Diğer boosting algoritmalarına karşı daha hızlı eğitim, daha yüksek verimlilik, daha iyi doğruluk, daha düşük bellek (RAM) kullanımı, büyük ölçekli verileri işleme yeteneği, paralel öğrenme ve GPU öğrenimini destekleme özelliklerine sahiptir (LightGBM 2021).

LightGBM, sürekli öznitelik değerlerini aynı kutulara toplayan histogram tabanlı algoritmaları kullanır. Bu sürekli değere sahip değişkenleri kesikli hale getirmesiyle hesaplama maliyetini azaltmaktadır. Birçok karar ağacı algoritması Şekil 2.17'de yer alan resimdeki gibi seviye bazlı büyümektedir (LightGBM 2021).



Şekil 2.17: Seviye odaklı büyüme.

LightGBM, Şekil 2.18'deki gibi ağaçları yaprak odaklı olarak büyütür. Büyümek için maksimum delta kaybı olan yaprağı seçer. Yani kaybı azaltan yapraklardan bölünme işlemini devam ettirir. Bu sayede model daha hızlı öğrenir. Yaprak odaklı büyüme, veri seti küçük olduğunda aşırı öğrenmeye sebep olabilir. Bu yüzden LightGBM, maksimum ağaç derinliğini sınırlayabilen parametreye sahiptir. Maksimum derinlik belirtilmiş olsa bile ağaçlar yaprak odaklı büyümeye devam ederler (LightGBM 2021).



Şekil 2.18: Yaprak odaklı büyüme.

Veri setlerinde yer alan kategorik öznitelikleri makine öğrenmesi modeline vermeden önce tek sıcak kodlama yapılması gereklidir. Yüksek kardinaliteye sahip kategorik öznitelikler için tek sıcak kodlama yapılması kurulan ağaçların dengesiz olmasına sebep olmaktadır. Bu sebeple iyi bir doğruluk elde etmek için daha derinlere büyüme gerekmektedir. Buna çözüm olarak tek sıcak kodlama yapmak yerine kategorik özelliği 2 alt kümeye bölmektir (LightGBM 2021).

Boosting, kurulan ağaçların artık hataları kullanılarak yeni ağaçlar kurulmasıdır. Kurulan ağaçlar birbirlerine bağımlıdırlar. Bir önceki ağaçta az öğrenilen veri örneklerine daha fazla önem vermek için daha fazla ağırlık verilir.

LightGBM, literatürde Gradyan Artırmalı Karar Ağacı (Gradient Boosting Decision Tree-GBDT) olarak da bilinmektedir. Sırayla eğitilmiş karar ağaçlarının topluluk modelidir. Her ağaç artık hataların üzerinden veriyi öğrenmeye çalışır. Yani bir gerçek çıktı ile ağırlıklı tahminler toplamı arasındaki farkı öğrenmeye çalışır. Hataları en aza indirirken gradyan yöntemini kullanır. LightGBM'de iki yeni teknik kullanılmıştır. Bunlar Gradyan Tabanlı Tek Yönlü Örnekleme (Gradient-based One-Side Sampling-GOSS) ve Özel Özellik Paketi (Exclusive Feature Bundling-EFB)'dir.

GOSS, veri örnekleri ile ilgilenir. Burada amaç karar ağaçlarının doğruluk oranını korurken veri sayısını azaltmaktır. Gradient Boosting yöntemi bilgi kazancını hesaplarken her öznitelik için tüm veriyi tarar. Küçük gradyana sahip olmak iyi eğitilmiş olmayı, büyük gradyana sahip olmak ise yetersiz eğitimi göstermektedir. Veri setinde büyük gradyanlara sahip örnekler daha az eğitilmiş oldukları için bilgi kazanımına katkıları daha fazladır. Bu sebeple GOSS sadece önemli olan verileri

kullanır. Yani gradyanları büyük olan verileri tutar ve gradyanı küçük olan verileri rastgele düşürür.

GOSS, örneklerin gradyanların mutlak değerlerini azalan olacak şekilde sıralar. Büyük gradyanlara sahip verinin en üst kısmından $\%a$'lık bir parça alır. Verinin küçük gradyanlara sahip olan geriye kalan kısmından ise $\%b$'lik rastgele örnekler alır. Az eğitilmiş olan örneklere odaklanabilmek için küçük gradyanlara sahip veri örneklerinin katkısını $(1-a)/b$ katsayısıyla çarparak artırır. Bu işlemle az eğitilen örneklerin üzerinde durur ancak veri dağılımını az da olsa etkilenir.

EFB, değişken sayısı ile ilgilenir. Veri setinde hiçbir zaman aynı anda sıfır olmayan değerlere sahip özellikleri tek bir pakette toplar. Tüm özelliklerin sıfır olduğu örnekler için yeni paket oluşturur. Burada amaç doğruluk oranını korurken değişken sayısını azaltmaktır. Bu işlemi, seyrek özellikleri birleştirip daha yoğun özellikler oluşturarak yapar. Böylece verinin karmaşıklığının azalmasını bu sayede de düşük bellek tüketilmesini sağlar. Bu da eğitim sürecinin hızlı olmasını sağlar (Ke ve diğ. 2017).

2.4 Sınıflandırma

Sınıflandırma, makine öğrenmesi alanındaki problem türlerinden biridir. Sınıflandırmanın amacı, bir girdiye bir etiket (sınıf) atamaktır. Bir problemin sınıflandırma problemi olabilmesi için hedef değişken tipinin kategorik olması gerekmektedir. Karşılaşılan veri hikayeleri genellikle ikili sınıflandırma, çok sınıflı sınıflandırma, çok etiketli sınıflandırma veya çok çıktılı sınıflandırma tipindedir. Bu sınıflandırma tiplerinin bazı özelliklerine Tablo 2.2'de yer verilmiştir.

Tablo 2.2: Sınıflandırma tipleri ve özellikleri.

Sınıflandırma Tipi	Çıkış Sayısı	Çıkış Tipi
İkili Sınıflandırma	2	İkili
Çok Sınıflı Sınıflandırma	3 veya daha fazla	Çok değerli
Çok Etiketli Sınıflandırma	2 veya daha fazla	İkili
Çok Çıktılı Sınıflandırma	2 veya daha fazla	Çok değerli

İkili sınıflandırma problemi hedef değişkenin ait olabileceği iki sınıfın olması durumudur. Bu tür sınıflandırmada gözlemlere ait çıkış değerleri hedef değişkendeki iki sınıftan sadece biridir. Örneğin, banka veya telekomünikasyon sektöründe müşterilerin bir sonraki hizmet döneminde şirketi terk edip etmeyeceğine dair tahminleme işlemi ikili sınıflandırmaya aittir. Burada bir kişinin değerlendirmeler sonucu terk eder veya terk etmez adı altında alabileceği sadece iki durum mevcuttur. Bu durum e-posta adreslerine gelen maillerin spam veya spam değil, hastane sektöründe kişinin hasta veya hasta değil, banka sektöründe kredi çekme durumunda dolandırıcı veya dolandırıcı değil gibi farklı senaryolarla karşımıza çıkabilmektedir. İkili sınıflandırma problemlerinin değerlendirmek için kullanılan metrikler doğruluk, kesinlik (precision), duyarlılık (recall), f1-score, ROC, AUC' tur.

Çok sınıflı sınıflandırma problemi hedef değişkenin ait olabileceği ikiden fazla sınıfın olması durumudur. Bu tür sınıflandırmada gözlemlere ait çıkış değerleri hedef değişkendeki herhangi sınıftan sadece biri olduğu durumdur. Yani çıkış değerimiz çok sınıftan oluşmaktadır ama tek etikete sahiptir. Örneğin, elimizde birden fazla hayvan türüne ait resimler olsun. Bunlar kedi, güvercin, aslan, kelebek olsun. Bu resimleri sınıflandırırken kedi resmine hem kedi hem aslan diyemeyiz. Sadece bir çıktı etiketi vardır ve bu etikette kedi etiketidir. Aynı durum diğer hayvanlar için de geçerlidir. Her hayvana sadece bir sınıf etiketi atanabilir. Başka bir örnek olarak kan gruplarının durumu verilebilir. Bir insanın kan grubu A0, B0, AA, BB, 00, AB gruplarından birisi olabilir. Bir insanın bu kan gruplarından ikili, üçlü veya daha fazla kombinasyona sahip kan grubuna ait olması imkansızdır. Kısacası her insanın sadece bir kan grubuna sahip anlamı çıkarılabilir. Bu iki örnekten çıkan sonuca göre birden fazla sınıf olmasına rağmen her bir gözlemin sadece bir sınıfa ait olması çok-sınıflı sınıflandırma durumdur. Bu sınıflandırma probleminin değerlendirme metrikleri ikili sınıflandırmada olduğu gibi doğruluk, kesinlik, duyarlılık, f1-score, ROC, AUC' tur. Burada ROC grafiğinde her bir sınıf için ayrı ayrı eğriler çizilmektedir (Müller ve diğ 2016).

Çok-etiketli sınıflandırma probleminde ise hedef değişkende çok sınıflı sınıflandırmadaki gibi 2'den fazla sınıf vardır. Çok sınıflı sınıflandırmadan farkı ise bir gözlemin birden fazla sınıfa aynı anda ait olabilmesi durumudur. Örneğin, film türlerini ele alalım. Bir film aksiyon, macera, suç türlerine aynı anda ait olabilir.

Buradaki aksiyon, macera, suç türleri çok sınıflı bir durumu, film türünün bu üç sınıfa da ait olması ise çok etiketli bir durumu temsil etmektedir. Başka bir örnek olarak herhangi resimde dağ, kuş, çimen, su ve ağacın olması çok sınıflı sınıflandırmayı temsil etmektedir. Bir resmi incelediğimizde bu sınıflardan dağ, kuş, çimen ve su resimde yer alsın. Resimde birden fazla sınıf aynı anda bulunduğu için bu durum çok etiketli sınıflandırmayı göstermektedir. Bu tez çalışmasının da konusu olan hasta örneğini incelersek bir hastaya birden fazla hastalık teşhisi konulmasının çok etiketli sınıflandırmaya dahil olduğu görülmektedir. Bu sınıflandırmayı değerlendirmek için kullanılan metrikler ise daha önce bahsedilen sınıflandırma problemlerinden farklıdır. Bu metriklere ayrıca değinilecektir.

Bu tez çalışmasında iki veri seti incelenmiştir. Veri setlerinin birincisinde çok-sınıflı ve çok-etiketli sınıflandırma ikincisinde ikili sınıflandırma yöntemleri ele alınmıştır.

2.4.1 İkili Sınıflandırma İçin Değerlendirme Metrikleri

Karmaşıklık matrisi, sınıf örneklerinin hangi sınıfa dahil edildiğini açıkça gösteren bir metriktir. Tablo 2.3'te yer alan karmaşıklık matrisi ikili sınıflandırmada kullanılan gösterimidir.

Tablo 2.3: İkili sınıflandırma karmaşıklık matrisi.

		Tahmin Edilen Değerler	
		0	**1**
Gerçek Değerler	**0**	Doğru Negatif	Yanlış Pozitif
	1	Yanlış Negatif	Doğru Pozitif

Çok sınıflı sınıflandırma uygulamalarının başarılarını değerlendirmek için kullanılan metriklerden bir tanesi Denklem (2.28)'de yer alan doğruluk değeridir. Doğruluk, hedef değişkenin doğru sınıflandırma oranını gösterir.

$$\text{Doğruluk} = \frac{\text{Doğru Pozitif Sayısı} + \text{Doğru Negatif Sayısı}}{\text{Tüm Örnek Sayısı}} \tag{2.28}$$

Ancak doğruluk değerini tek başına değerlendirmek sınıflandırıcının başarısını belirlemek için iyi bir yol değildir. Ayrıca doğruluk metriği, özellikle dengesiz sınıf dağılımına sahip veri setlerinde yanıltıcı bir başarı gösterir.

Denklem (2.29)'da yer alan kesinlik (precision), sınıflandırma sonucunda doğruluğun yanında bakılması gereken başarı metriklerinden biridir. Bu metrik, pozitif sınıf tahminlerinin başarı oranıdır.

$$\text{Kesinlik} = \frac{\text{Doğru Pozitif}}{\text{Doğru Pozitif} + \text{Yanlış Pozitif}} \tag{2.29}$$

Denklem (2.30)'da yer alan duyarlılık (recall), kesinlik metriğinin yanında bakılan bir diğer başarı metriğidir. Bu metrik, pozitif sınıfın doğru tahmin edilme oranıdır.

$$\text{Duyarlılık} = \frac{\text{Doğru Pozitif}}{\text{Doğru Pozitif} + \text{Yanlış Negatif}} \tag{2.30}$$

Denklem (2.31)'de yer alan F1-score, kesinlik ve duyarlılık parametrelerini aynı anda değerlendirebilmek için kullanılan bir metriktir. Sonuçlarda yanlılığı ortadan kaldırmak için kesinlik ve duyarlılığın harmonik ortalaması alınarak hesaplanır.

$$\text{F1} - \text{Score} = 2 \text{ x } \frac{\text{Kesinlik} \times \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \tag{2.31}$$

Şimdiye kadar ele alınan metrikler bize sınıflandırıcının başarısı hakkında bilgi edinmemizi sağlar. Ancak bu metriklerin hesaplanmasında kullanılan bir eşik değeri vardır ve Scikit-learn'de varsayılan eşik değeri 0.5'tir. Bu eşik değer değiştirilebilirdir. Eşik değerin değişimi ile hedeflenen değerlendirme metriğinde gerekli skor elde edilebilir. Fakat bu yöntem sınıflandırıcının başarısını net olarak ortaya koymada yeterli değildir. Değiştirilen eşik değeri ile doğruluk, kesinlik, duyarlılık, f1-score değerleri tamamen değişmektedir.

ROC eğrisi (Receiver Operating Characteristic Curve) metriği diğer metriklerden farklı olarak eşik değerinin değişmesi ile değişmez. Bu yüzden stabil bir eğridir. Eğrinin dikey ekseninde Denklem (2.32)'de yer alan doğru pozitif oran

değeri (kesinlik, precision) yatay ekseninde Denklem (2.33)'te yer alan yanlış pozitif oran değerleri bulunmaktadır.

$$\text{Doğru Pozitif Oranı} = \frac{\text{Doğru Pozitif}}{\text{Doğru Pozitif} + \text{Yanlış Negatif}} \tag{2.32}$$

$$\text{Yanlış Pozitif Oranı} = \frac{\text{Yanlış Pozitif}}{\text{Yanlış Pozitif} + \text{Doğru Negatif}} \tag{2.33}$$

Eşik değeri 0 ile 1 arasında değer alabilmektedir. Her bir eşik değeri için sınıf tahmini yapılarak sonuçlar hazırlanmaktadır. Bu sonuçlarla karmaşıklık matrisi hesaplanmaktadır. Her eşik değeri için bir doğru pozitif ve yanlış pozitif değeri belirlenmektedir ve grafik üzerinde işaretlenmektedir. Tüm eşik değeri değerleri için aynı işlem yapılmaktadır. Grafiğin değerlendirilebilmesi için eğrinin altında kalan alanın integrali alınmaktadır. Bu alana AUC (Area Under Curve) ismi verilmektedir. AUC, ROC eğrisinin tek bir sayısal değer ile ifade ediliş biçimidir ve tüm olası sınıflandırma eşikleri için toplu bir performans ölçütüdür. AUC değeri 0 ile 1 arasında değer alabilmektedir. Değer ne kadar 1'e yakın olursa sınıflandırıcının performansının o kadar iyi olduğu anlaşılmaktadır (Albon 2018).

2.4.2 Çok-Sınıflı Sınıflandırma İçin Değerlendirme Metrikleri

Karmaşıklık matrisi, sınıf örneklerinin hangi sınıfa dahil edildiğini açıkça gösteren bir metriktir. Tablo 2.4'te çok sınıflı sınıflandırma için karmaşıklık matrisi örneği bulunmaktadır.

Tablo 2.4: Çok sınıflı sınıflandırma karmaşıklık matrisi.

		Tahmin Edilen Değerler			
		Sınıf 1	**Sınıf 2**	**Sınıf 3**	**Sınıf 4**
Gerçek Değerler	**Sınıf 1**	Hücre 1	Hücre 2	Hücre 3	Hücre 4
	Sınıf 2	Hücre 5	Hücre 6	Hücre 7	Hücre 8
	Sınıf 3	Hücre 9	Hücre 10	Hücre 11	Hücre 12
	Sınıf 4	Hücre 13	Hücre 14	Hücre 15	Hücre 16

Sınıf 1 için değerlendirme yapılırsa doğru pozitif değeri Hücre 1'den elde edilir. Yanlış negatif değeri Hücre 2, Hücre 3 ve Hücre 4'ün toplam değerinden elde

edilir. Yanlış pozitif değeri Hücre 5, Hücre 9, Hücre 13'ün toplam değerinden elde edilir. Doğru negatif değeri Hücre 6, Hücre 7, Hücre 8, Hücre 10, Hücre 11, Hücre 12, Hücre 14, Hücre 15 ve Hücre 16'nın toplam değerinden elde edilir.

Çok sınıflı sınıflandırma uygulamalarının başarılarını değerlendirmek için kullanılan metriklerden bir tanesi doğruluk değeridir. Dengeli sınıf dağılımına sahip veri setlerinde kullanımı uygundur. İkili sınıflandırma için kullanılan doğruluk, kesinlik, duyarlılık, fl-score, doğru pozitif oranı ve yanlış pozitif oranı formülleri çok sınıflı sınıflandırma için de geçerlidir. Bu metrikler çok sınıflı sınıflandırmada her bir sınıf için ayrı ayrı hesaplanmaktadır.

2.4.3 Çok-Etiketli Sınıflandırma İçin Değerlendirme Metrikleri

$\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_f$ keyfi kümeler ve f giriş özelliklerinin sayısı olmak üzere $\mathbf{X} \in \mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_f$ olan veri örnekleri ile $\boldsymbol{\mathcal{X}}$ giriş uzayıdır. Bu yüzden, her **X** örneği, bu kümelerin kartezyen ürünü olarak elde edilir (Herrera 2016).

L tüm olası etiketlerin kümesi olsun. **P**(**L**), boş küme ve *L*'nin kendisi dahil olmak üzere tüm olası $l \in \mathbf{L}$ etiket kombinasyonlarını içeren **L**'nin güç kümesini belirtir. **L'**deki toplam etiket sayısı ise k ile temsil edilir.

$\boldsymbol{\mathcal{Y}}$, tüm olası vektörleriyle birlikte çıkış uzayı olsun. **D** , $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_f$ x $\mathbf{P}(\boldsymbol{L})$'nin sonlu bir alt kümesini içeren çok etiketli bir veri kümesini göstersin. Her eleman $(\mathbf{X}, \mathbf{Y}) \in \mathbf{D} | \mathbf{X} \in \mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_f$ ve $\mathbf{Y} \in \mathbf{P}(\mathbf{L})$ veri örneği olacak. **Y**'nin uzunluğu her zaman k olacaktır. **D**'nin eleman sayısı n ile temsil edilir.

F, çok etiketli sınıflandırıcı olsun ve $\mathbf{F} : \boldsymbol{\mathcal{X}} \rightarrow \boldsymbol{\mathcal{Y}}$ olarak tanımlansın. **F**'ye girişler herhangi bir $X \in \boldsymbol{\mathcal{X}}$ örneği ve çıkışları $Z \in \boldsymbol{\mathcal{Y}}$ tahminleri olacaktır. Bu yüzden herhangi bir örnekle ilişkili etiket vektörünün tahmininin gösterimi $\mathbf{Z} = \mathbf{F}(X)$ şeklinde olabilir.

Denklem (2.34)'teki etiket kardinalitesi, veri kümesindeki örnek başına ortalama etiket sayısıdır.

$$\text{Kardinalite} = \frac{1}{n}\sum_{i=1}^{n}|\mathbf{Y}_i| \tag{2.34}$$

Buradaki n, veri seti **D**'deki örneklerin sayısını, $\mathbf{Y}_i$ ise i'inci örneğin etiket kümesini belirtir. Etiket kardinalitesi ne kadar yüksekse örnek başına etkin etiket sayısı o kadar fazladır. Kardinalitesi 1'e yakın olan düşük kardinaliteler örneklerin çoğunun yalnızca bir etikete sahip olduğunu gösterir. Bu yüzden çok etiketli veri seti sayısı azdır.

Denklem (2.35)'te yer alan etiket yoğunluğu, veri kümesindeki örnek başına ortalama etiket sayısının toplam etiket sayısına bölünmesiyle hesaplanır. Kısacası etiket kardinalitesini toplam etiket sayısına bölünerek bulunur.

$$\text{Etiket Yoğunluğu} = \frac{1}{k}\frac{1}{n}\sum_{i=1}^{n}|\mathbf{Y}_i| \tag{2.35}$$

Buradaki n, veri seti **D**'deki örneklerin sayısını, $\mathbf{Y}_i$ i. örneğin etiket kümesini, k ise veri seti **D**'deki dikkate alınan toplam etiket sayısını belirtir. Yüksek etiket yoğunluğu veri setindeki etiketlerin her durumda iyi temsil edildiğini gösterir. Etiket kardinalitesi veri kümesinde bulunan etiket sayısından bağımsızdır ancak etiket yoğunluğu içinde bulunan etiket sayısını dikkate alır.

2.4.3.1 Örnek Temelli Metrikler

Çok etiketli sınıflandırıcıların başarısını değerlendirmek için kullanılan en yaygın metrik Denklem (2.36)'da yer alan hamming kaybıdır. Formülasyonundaki Δ operatörü i. örneğinin gerçek etiket kümesi olan $\mathbf{Y}_i$ ile tahmin edilmiş olan $\mathbf{Z}_i$ arasındaki simetrik farkını verir. Bu metrik yanlış tahmin edilen etiketlerin toplam etiket sayısına oranını verir. İdeal bir sınıflandırıcının hamming kaybı 0'dır. Sınıflandırıcının performansını artırabilmek için hamming kaybının azaltılması gerekir.

$$\text{Hamming Kaybı} = \frac{1}{n}\frac{1}{k}\sum_{i=1}^{n}|\mathbf{Y}_i \Delta \mathbf{Z}_i| \tag{2.36}$$

Çok etiketli sınıflandırıcıların doğruluk hesaplaması diğer sınıflandırıcı tiplerinin doğruluk değeri hesaplamasından farklılık göstermektedir. Denklem (2.37)'de gösterildiği gibi doğru tahmin edilen etiket sayısı toplam etiket sayısına oranlanır. Bu işlem tüm örnekler için yapılır ve sonuçların ortalaması doğruluk değerini verir.

$$\text{Doğruluk} = \frac{1}{n}\sum_{i=1}^{n}\frac{|\mathbf{Y}_i \cap \mathbf{Z}_i|}{|\mathbf{Y}_i \cup \mathbf{Z}_i|} \tag{2.37}$$

Denklem (2.38)'de verilen kesinlik metriği, doğru tahmin edilen etiket sayısı ile tahmin edilen toplam etiket sayısı arasındaki oranı verir.

$$\text{Kesinlik} = \frac{1}{n}\sum_{i=1}^{n}\frac{|\mathbf{Y}_i \cap \mathbf{Z}_i|}{|\mathbf{Z}_i|} \tag{2.38}$$

Denklem (2.39)'da yer alan duyarlılık, doğru tahmin edilen etiket sayısı ile gerçek değerlerin toplam etiket sayısı arasındaki oranı verir.

$$\text{Duyarlılık} = \frac{1}{n}\sum_{i=1}^{n}\frac{|\mathbf{Y}_i \cap \mathbf{Z}_i|}{|\mathbf{Y}_i|} \tag{2.39}$$

Denklem (2.40)'ta yer alan F1-Score, kesinlik ile duyarlılığın harmonik ortalaması ile bulunur.

$$\text{F1} - \text{Score} = 2 \times \frac{\text{Kesinlik} \times \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \tag{2.40}$$

Denklem (2.41)'de yer alan alt küme doğruluğu, bahsedilen diğer metriklerin yanında en katı değerlendirme ölçütüdür. Denklemde görüldüğü üzere tahmin edilen ve gerçek etiket kümeleri için tam eşitlik beklenmektedir. Bu metrik sınıflandırma doğruluğu veya etiket kümesi doğruluğu olarak da bilinir.

$$\text{Alt küme doğruluğu} = \frac{1}{n}\sum_{i=1}^{n}[[\mathbf{Y}_i = \mathbf{Z}_i]] \tag{2.41}$$

Alt küme doğruluğunun tam olarak nasıl hesaplandığı Tablo 2.5 üzerinden gösterilmiştir.

Tablo 2.5: Alt küme doğruluğu örneği.

Örnekler	Özellikler	Gerçek Etiketler	Tahmin Edilen Etiketler	Alt Küme Doğruluğu
X_1	…	[1,0,0,1]	[0,0,1,1]	0
X_2	…	[1,0,0,0]	[1,1,0,0]	0
X_3	…	[0,1,0,1]	[0,1,0,1]	1
X_4	…	[1,1,1,1]	[1,1,0,0]	0
X_5	…	[0,1,1,0]	[0,1,1,0]	1

Veri setimiz metriğin nasıl çalıştığını daha kolay anlayabilmek için veri setimizde beş örneğin olduğunu varsayılsın. Bu örneklere ait gerçek ve tahmin edilen etiketler mevcut. X_1 örneği için dört etiketten iki tanesi doğru tahmin edilmiş olmasına rağmen alt küme doğruluğu bunu tamamen yanlış olarak değer döndürür ve bu işlem sonucuna sıfır verir. Örnek iki incelenildiğinde dört etiketten üç doğru 1 yanlış tahmin yapılmasına rağmen alt küme doğruluğu bu işleme de sıfır değeri döndürür. Örnek üçte ise tüm etiketler doğru tahmin edilmiş ve alt küme doğruluğu 1 olarak hesaplanmıştır. Bunun sebebi alt küme doğruluğunun örneklere ait etiketlerin tahmin edilen etiketlerle tam eşleşmesini beklemesidir. Bu veri setinin alt küme doğruluğu ise beş örnekten iki tanesi doğru olduğu için %40'tır.

Scikit-learn kütüphanesinde metrikler dokümantasyonunda yer alan accuracy_score fonksiyonu ikili ve çok sınıflı sınıflandırma için Denklem (2.28)'deki gibi hesaplama yaparken çok etiketli sınıflandırma için Denklem (2.41)'de yer alan alt küme doğruluğu hesaplaması yapmaktadır (Scikit-learn 2011).

Bu değerlendirme metriklerinden doğruluk, kesinlik, duyarlılık, f1-score ve alt küme doğruluğunun yüksek, hamming kaybının düşük olması beklenir.

2.4.4 Ölçeklendirme

Ölçeklendirme, veri setindeki farklı değer aralıklarına sahip değişkenleri belli bir standartta olmasını sağlar. Özellikle uzak temelli yöntemlere sahip ve gradyan azalan yöntemini kullanan algoritmalar için ölçeklendirme yapılması gerekir.

Ölçeklendirme sayesinde makine öğrenmesi modeli daha iyi performans gösterir ve sonuca daha hızlı yakınsar. Bu tezde kullanılan ölçeklendirme yöntemi Denklem (2.42)'de yer alan Robust ölçeklendirmedir. Bu ölçeklendirmenin kullanılmasının sebebi medyan temelli olduğu için aykırı değerlere karşı dayanıklı olmasıdır (Brownlee 2020).

$$X_{yeni} = \frac{X - X_{medyan}}{\text{Çeyrekler aralığı (IQR)}} \tag{2.42}$$

2.4.5 Çapraz Doğrulama

Makine öğrenmesi uygulamalarında veri analizi ve veri ön işleme adımlarından sonra verinin eğitim ve test verisi olarak ikiye bölünmektedir. Burada önemli olan nokta oluşturulan modelin test verilerini daha önce görmemiş olmasıdır. Ayrılan eğitim seti üzerinde algoritmaların en iyi sonuçları veren parametre değerleri bulunduktan sonra model test verisi üzerinde test edilir. Eğitim verisi bazı durumlarda veri setinin tüm veriyi temsil edemeyebilir. Bu durumlarda k-katlı çapraz doğrulama yöntemi kullanılmaktadır. Bu yöntem eğitim verisini belirlenen k değeri kadar parçaya ayırır ve her bir parçayı ayrı bir eğitim sürecinde kullanarak modele en uygun olacak parametreleri belirler. Burada elde edilen performans ölçütü, döngüde hesaplanan değerlerin ortalamasıdır. Son olarak parametrelerle güncellenen model test verisi üzerinde test edilir (Scikit-learn 2011).

2.4.6 Bire Karşı Geriye Kalan Sınıflandırması

Bire karşı geriye kalan sınıflandırması aslında çok sınıflı sınıflandırma için kullanılan bir yöntemdir. Bu yöntem sınıf başına bir sınıflandırıcı oluşturur. Her sınıflandırıcı için, bir sınıf diğer tüm sınıflara karşı uyarlanır. Her sınıf için bir sınıflandırıcıya ihtiyaç olduğu için hesaplama verimliliği sağlamaktadır ve sınıflar hakkında bilgi edinilebilmektedir. Ayrıca bu yöntem çok etiketli sınıflandırma için de kullanılabilir. Sınıflandırma yönteminin kullanılabilmesi için hedef etiketleri içeren bir matris oluşturulur. Yani hedef etiketler 2 boyutlu ve 0-1 değerlerini içeren matris olarak biçimlendirilmelidir. Bu sınıflandırıcı, her etiket için bağımsız olarak

ikili sınıflandırıcıyı eğitmeyi içeren çok etiketli sınıflandırmayı gerçekleştirmek için ikili uygunluk yöntemini kullanır (Scikit-learn 2011).

3. DENEYSEL SONUÇLAR

Bu bölümde Pamukkale Üniversitesi Hastanesi İç Hastalıkları Polikliniği'ne 2021 yılında başvurmuş kişilerin kan testlerini içeren veri seti üzerinde yapay sinir ağları ve destek vektör makineleri olmak üzere iki adet geleneksel ve hafif gradyan artırma makineleri olmak üzere bir adet modern makine öğrenmesi algoritması kullanılarak hastalık sınıflandırma uygulaması yapılmıştır. Bu bölümde yapılan veri anlama, veri hazırlama, veri modelleme ve elde edilen sonuçlar incelenecektir. Veri modellemesinden önceki veri anlama ve veri hazırlama adımları makine öğrenmesi ve veri madenciliği gibi uygulamaların %80'ini oluşturmaktadır. Bu adımlar direkt olarak model performansını etkilediği için önemi çok büyüktür. Bu tez çalışmasında, elde edilmiş olan veri seti üzerinde çok-sınıflı sınıflandırma ve çok-etiketli sınıflandırma olmak üzere iki farklı sınıflandırma çalışması yapılmıştır.

3.1 Kullanılan Donanımlar ve Yazılımlar

Günümüzde makine öğrenmesi modellerini oluşturmak için kullanılan birçok programlama dili bulunmaktadır. Bunlardan en popüleri yapay zeka çalışmaları için gerekli tüm kütüphaneleri barındırması, öğrenmesinin ve okunmasının kolay olması, taşınabilir olması, açık kaynak kodlu olması sebebiyle Python'dur.

Bu tez çalışmasında hem endokrinoloji alanına ait hem de literatürde sıkça kullanılmış bir diyabet veri seti ele alınarak, Python programlama dili ve Visual Studio Code editörü kullanılarak yapay sinir ağları, destek vektör makineleri ve hafif gradyan artırma makinesi makine öğrenmesi modelleri kurulmuş ve eğitilmiştir. Model eğitimi Intel i7-10875H işlemcili ve 32 GB RAM bulunan bilgisayar üzerinde gerçekleştirilmiştir.

Veri bilimi ve makine öğrenmesi için açık kaynak Python kütüphaneleri mevcuttur. Scikit-learn, Tensorflow, Keras, PyTorch, Caffee bu kütüphanelerden bazılarıdır. Bu çalışmada veri analizi için NumPy ve pandas, veri görselleştirme için

matplotlib ve seaborn, veri modelleme için Scikit-learn, Keras ve lightgbm kütüphaneleri kullanılmıştır.

3.2 Endokrinoloji Veri Seti

Pamukkale Üniversitesi Hastanesi Bilgi İşlem veri tabanından 2021 yılına ait Pamukkale Üniversitesi Hastanesi Endokrinoloji Polikliniği'ne başvurmuş kişilerin bilgilerini içeren iki adet veri dosyası alınmıştır.

Birinci veri dosyası sırasıyla 'HASTA_NO, 'PROTOKOL_NO', 'ADI', 'SOYADI', 'DOGUM_TARIHI', 'GELIS_TARIHI', 'KOD', 'TANI_ADI' sütunlarından oluşan hasta bilgilerini ve hastaya konulan tanı bilgilerini içermekte olup .xls formatındadır. İkinci veri dosyası sırasıyla 'HASTA_NO', 'PROTOKOL_NO', 'RESMI_KOD', 'GELIS_TARIHI', 'ISLEM_TARİHİ', 'PARAMETRE_ADI', 'SONUC' sütunlarından oluşan hasta bilgilerini ve hastalara uygulanmış kan testleri sonuçlarını içermekte olup .xlsx formatındadır.

Tezin bundan sonraki bölümlerinde karışıklık olmaması açısından birinci dosya için 'Tanı Verileri', ikinci dosya için ise 'Sonuç Verileri' isimlendirilmeleri kullanılacaktır.

3.2.1 Veriyi Anlama

Veri seti bölümünde belirtilmiş olan Sonuç Verilerinde 966021 satır ve 7 sütun bulunmaktadır. Tanı Verilerinde ise 26589 satır ve 8 sütun mevcuttur. Veri setinde tanısı koyulmuş toplam 35 hastalık bulunmaktadır. Bu hastalıklara ait ICD-10 kodları ve bu kodlara uygun hastalık isimleri veri setinde yer almaktadır.

Sonuç Verilerinde 'SONUC' kolonu haricindeki kolonları kapsayacak şekilde tekrarlayan satırlar tespit edilmiştir. 966021 satırdan 13837 satırı tekrarlı olduğu için bu satırlar veri setinden atılmıştır. Yeni durumda satır sayımız 952184'tür. Sonuç Verilerine ait sütunların hangi tip veri içerdiği Tablo 3.1'de verilmiştir.

Tablo 3.1: Sonuç verilerine ait veri tipleri.

SÜTUN ADI	VERİ TİPİ
HASTA_NO	int64
PROTOKOL_NO	int64
RESMI_KODU	float64
GELIS_TARIHI	datetime64[ns]
ISLEM_TARIHI	datetime64[ns]
PARAMETRE_ADI	object
SONUC	object

Bu tabloya göre ‘SONUC’ kolonu nümerik değere dönüştürme işlemi yapılması gerekmektedir. Bu işlem sonucunda ‘SONUC’ kolonu float64 formatına dönüştürülmüştür.

Sonuç Verileri dosyasında kan testlerinin uygulandığı kişilerin test sonuçları her bir satırda verilmiştir. Veri analizi ve modelleme yapılabilmesi için kan testlerinin uygulandığı tarihe göre her bir kişinin bilgileri ve kan testleri sonuçları sadece bir satırda yer almalıdır. Bunun için veri tablomuz yeniden yapılandırılmalıdır. Bu işlemi yapabilmek için ‘HASTA_NO’, ‘PROTOKOL_NO’, ‘GELIS_TARIHI’, ‘ISLEM_TARİHİ’ kolonları indeks varsayılarak ‘PARAMETRE_ADI’ kolonundaki kan testleri isimleri yeni sütunlara dönüştürülmüştür ve toplama fonksiyonu ‘SONUC’ kolonuna uygulanarak kan testleri isimleri ile yeni oluşturduğumuz sütunlara sonuç değerleri atanmıştır. Artık veri setimizde her bir satırda bir kişinin kan testleri sonuçları yer almıştır. Bir kişi farklı tarihlerde doktor randevusu aracılığıyla kan testi yaptırabildiği için veri setimizde aynı kişiler birden fazla satırdan bulunmaktadır ancak uygulanan işlem tarihleri farklı olduğu için bu satırlar tekrarlayan satır olarak kabul edilmemiştir. Son durumda Sonuç Verileri adlı veri tablomuzda 27283 satır 61 sütun bulunmaktadır. Veriye ait değişkenler 'PROTOKOL_NO', 'HASTA_NO', 'ADI', 'SOYADI', 'DOGUM_TARIHI', 'ISLEM_TARIHI', ' ALT (Alanin Aminotransferaz)', 'ALP (Alkalen Fosfataz)', 'AST (Aspartat Transaminaz)', 'Anti TPO', 'Anti Tg (Anti Tiroglobulin Antikor)', 'BASO#', 'BASO%', 'BUN', 'Büyüme Hormonu (BH) - GH', 'EO#', 'EO%', 'Estradiol', 'FSH', 'Glukoz', 'HBA1C (%)', 'HBA1C (mmol/mol)', 'HCT', 'HDL Kolesterol', 'HGB', 'IMG#', 'IMG%', 'Kalsiyum (Ca)', 'Kolesterol', 'Kortizol', 'Kreatinin', 'LDL kolesterol', 'LH (Lüteinleştiren Hormon)', 'LYM#',

'LYM%', 'MCH', 'MCHC', 'MCV', 'MONO#', 'MONO%', 'MPV', 'NEU#', 'NEU%', 'NLR', 'P-LCC', 'P-LCR', 'PCT', 'PDW', 'PLT', 'Prolaktin', 'RBC', 'RDW', 'RDW-SD', 'Serbest T3', 'Serbest T4', 'TSH', 'TSH RESEPTOR BLOKE EDICI ANTIKOR', 'Total Testosteron', 'Trigliserid', 'VLDL Kolesterol', 'WBC', 'Üre', 'İnsülin', 'KOD' adlı sütunlardan oluşmaktadır.

Tanı Verileri veri setinin içerdiği farklı hastalıklara ait kişi sayısı Tablo 3.2'de belirtildiği gibidir.

Tablo 3.2: Veri setindeki hastalıklara ait gözlem sayıları.

ICD-10 KODU	KİŞİ SAYISI
E14.9	7753
E11.9	4688
E03.9	4629
E10.9	1377
E66.0	1251
E03.8	901
E05.9	492
E66.9	490
E66.2	348
E11.7	222
E22.1	214

Veri modellemesinin sağlıklı sonuç verebilmesi için hastalığa sahip kişi sayısı 100 üzerinde olan Tablo 3.2'de yer alan ICD-10 kodları ile işlemlere devam edilmiştir.

Tanı Verileri ile Sonuç Verilerini tek bir tabloda birleştirmek için her iki veri tablosundaki benzersiz olan 'PROTOKOL_NO' değerleri bulundu. Her iki veride de ortak olan 'PROTOKOL_NO' değerleri bulundu ve bu değerlerin indeks olarak tanımlanması sağlandı. Sonuç Verileri içerisinden indeks olarak belirlediğimiz 'PROTOKOL_NO' ya sahip kişiler filtrelendi. Aynı işlem Tanı Verileri üzerinde de uygulandı. Veri setleri birleştirilirken 'PROTOKOL_NO', 'HASTA_NO' ve 'GELİS_TARIHI' sütunları baz alınmıştır.

Birleştirilmiş veri seti üzerinde ‘ISLEM_TARİHİ’ ve ‘DOGUM_TARIHI’ kolonları üzerinden her bir hasta için yaş bilgilerini tutacak olan ‘YAS’ isimli yeni değişken üretilmiştir.

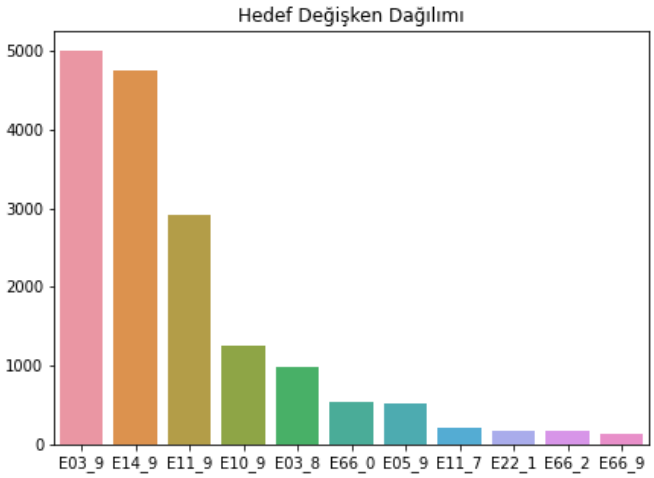
3.2.2 Çok Sınıflı Sınıflandırma Sonuçları

Bu çalışmada hedef değişken Tablo 3.3’teki ICD-10 kodları ile verilmiş farklı hastalık türlerini içerecek şekilde oluşturulmuştur. Çok sınıflı sınıflandırma açısından ele alındığında veri setindeki her bir örnek satırı hedef değişkendeki sınıflardan sadece birini işaret eder. Bu veriyi çok sınıflı olarak değerlendirebilmek için bir kişiye bir hastalık düşecek şekilde diğer hastalık verileri silinmiştir. Böylece her kişinin sadece bir hastalığı olacak şekilde veri seti düzenlenmiştir. Yeni düzende hastalıklara ait kişi sayısı Tablo 3.3’te verilmiştir.

Tablo 3.3: Çok sınıflı sınıflandırma verisinde sınıflara ait gözlem sayıları.

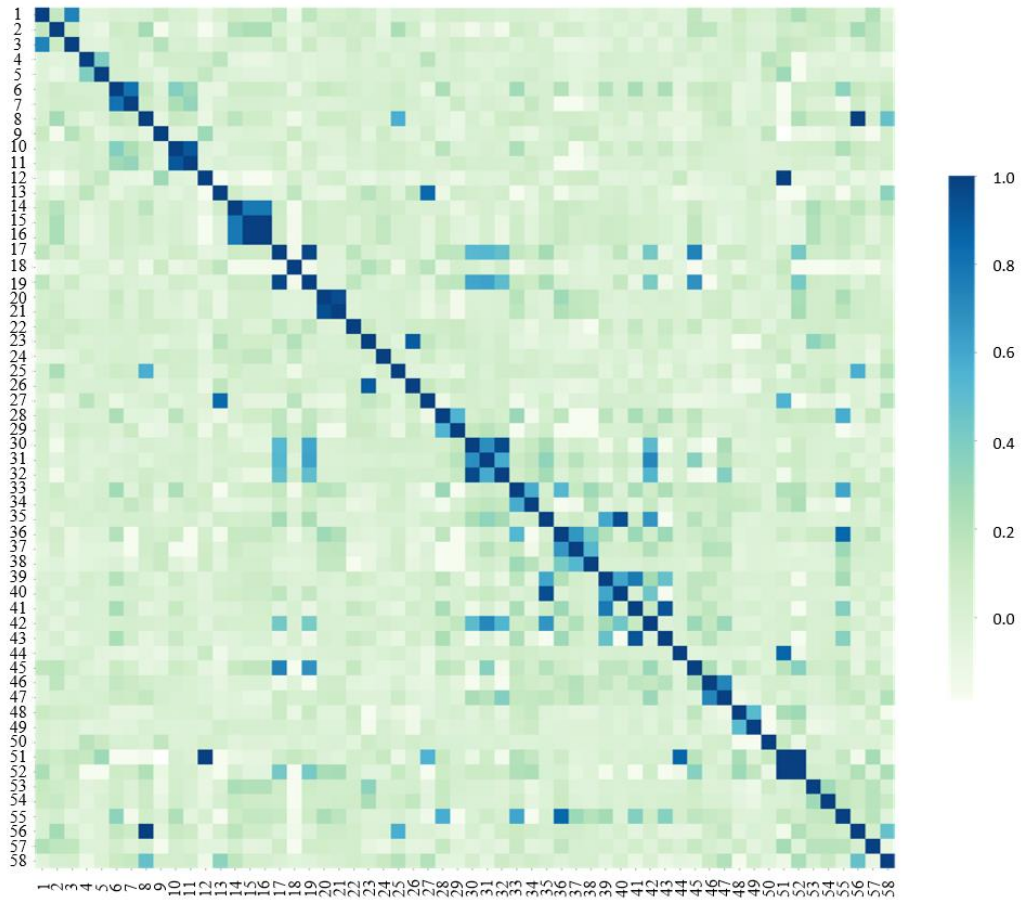
ICD-10 Kodları	Kişi Sayısı
E03.9	4997
E14.9	4737
E11.9	2906
E10.9	1256
E03.8	974
E66.0	534
E05.9	514
E11.7	212
E22.1	178
E66.2	163
E66.9	140

Hastalıklara ait kişi dağılımı Şekil 3.1’de verilmiştir.



Şekil 3.1: Hedef değişkendeki sınıfların dağılımı.

Veri setinde 16621 gözlem ve 1'i kategorik, 58'i nümerik, 8'i kardinal olmak üzere toplam 67 değişken vardır. Kardinal değişkenler eşsiz değerlerden oluştuklarından dolayı veri setinden çıkartılmıştır.



Şekil 3.2: Veriye ait korelasyon matrisi.

Şekil 3.2'deki korelasyon matrisi değişkenlerin ikili ilişkilerini temsil eden eder ve matristeki değerler +1 ile -1 arasında yer alır. Değişkenlerin kendileri ile olan ilişkileri 1 olduğu için çapraz hat tamamen koyu mavi renktedir. Korelasyon matrisindeki herhangi iki değişken arasında pozitif yönde bir ilişki varsa değer +1'e, negatif bir ilişki varsa -1'e yakın çıkacaktır.

Tablo 3.4: Korelasyon matrisinde yer alan değişkenler ve sıra numaraları.

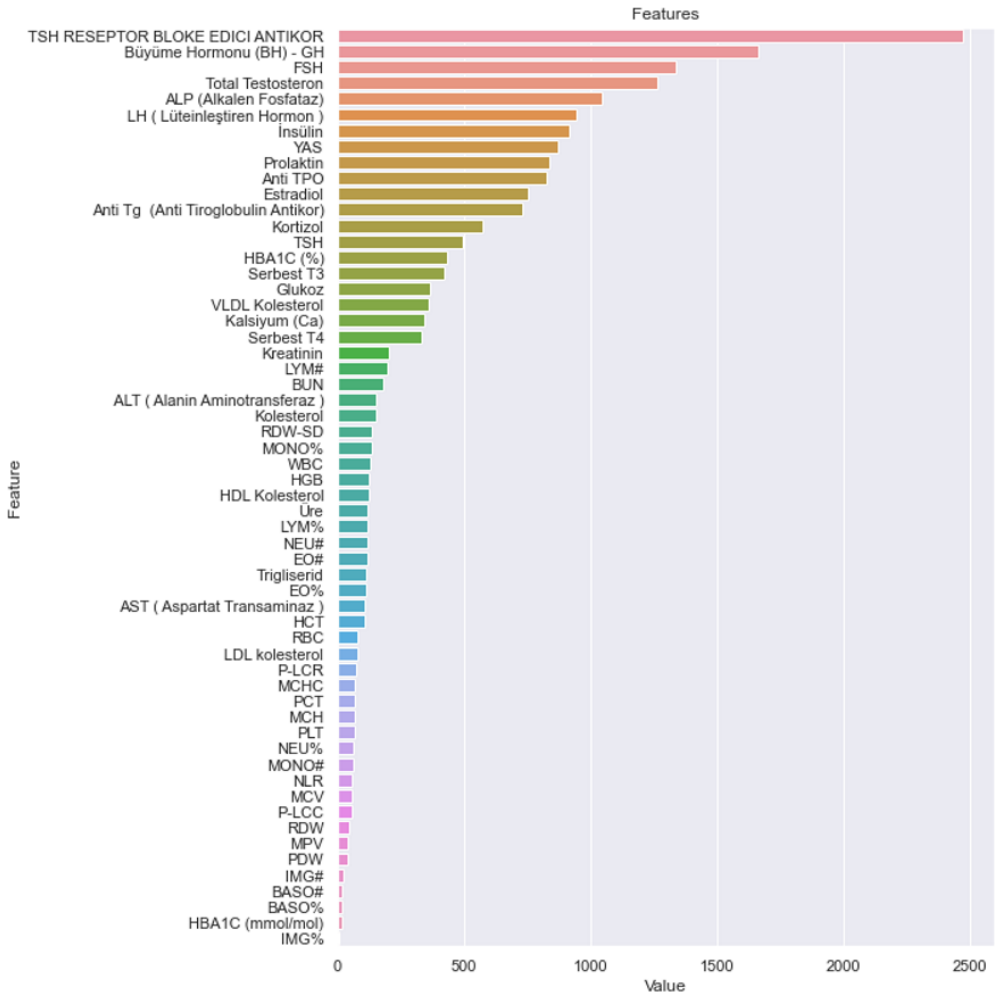
No	Değişken Adı	No	Değişken Adı
1	ALT (Alanin Aminotransferaz)	**30**	MCH
2	ALP (Alkalen Fosfataz)	**31**	MCHC
3	AST (Aspartat Transaminaz)	**32**	MCV
4	Anti TPO	**33**	MONO#
5	Anti Tg(Anti Tiroglobulin Antikor)	**34**	MONO%
6	BASO#	**35**	MPV
7	BASO%	**36**	NEU#
8	BUN	**37**	NEU%
9	Büyüme Hormonu (BH)-GH	**38**	NLR
10	EO#	**39**	P-LCC
11	EO%	**40**	P-LCR
12	Estradiol	**41**	PCT
13	FSH	**42**	PDW
14	Glukoz	**43**	PLT
15	HBA1C (%)	**44**	Prolaktin
16	HBA1C (mmol/mol)	**45**	RBC
17	HCT	**46**	RDW
18	HDL Kolesterol	**47**	RDW-SD
19	HGB	**48**	Serbest T3
20	IMG#	**49**	Serbest T4
21	IMG%	**50**	TSH
22	Kalsiyum (Ca)	**51**	TSH RESEPTOR BLOKE EDICI ANTIKOR
23	Kolesterol	**52**	Total Testosteron
24	Kortizol	**53**	Trigliserid
25	Kreatinin	**54**	VLDL Kolesterol
26	LDL kolesterol	**55**	WBC
27	LH (Lüteinleştiren Hormon)	**56**	Üre
28	LYM#	**57**	İnsülin
29	LYM%	**58**	YAS

Korelasyon matrisinde değişkenlere 1 ile 58 arasında numaralandırma yapılmış olup bu numaralara karşılık gelen değişken adlarına Tablo 3.4'te yer verilmiştir. Korelasyon matrisine bakıldığında 14., 15. ve 16. değişkenleri arasında, 51. ve 12. değişkenleri arasında, 56. ve 8. değişkenleri arasında güçlü bir korelasyon olduğu koyu mavi renginden anlaşılmaktadır.

Korelasyon matrisi yardımıyla korelasyonları düşük olan değişkenler kullanılarak modeli olumlu etkileyebileceği düşünülen yeni değişkenler elde

edilmiştir. Elde edilen değişkenlerin ham veri setinde olan değişkenlerle karışmaması için yeni değişkenlerin başına NEW yazısı eklenmiştir.

Veri setinden çıkarılacak değişkenlere LightGBM ile kurulan temel bir model üzerinden elde edilen Şekil 3.3'teki değişkenlerin önem sıralaması grafiğine bakılarak karar verilmiştir.



Şekil 3.3: Çok sınıflı için kurulan temel modele göre değişkenlerin önem sırası.

Veri setinde yer alan 'BASO#', 'VLDL Kolesterol', 'BASO%', 'PDW', 'PCT', 'LYM#', 'MCH', ' ALT (Alanin Aminotransferaz)', 'HBA1C (mmol/mol)', 'Trigliserid', 'Kolesterol', 'P-LCC', 'P-LCR', 'EO#', 'EO%', 'MONO#', 'MONO%', 'NLR', 'HDL Kolesterol', 'MPV', 'HCT', 'HGB', 'MCHC', 'MCV', 'MCH', 'RDW', 'WBC', 'PLT', 'RBC', 'RDW-SD', 'Üre', 'NEU#', 'AST (Aspartat Transaminaz)', 'LDL kolesterol', 'IMG#', 'IMG%', 'NEU%', 'LYM%', 'EO#', 'NLR' sütunları çıkarılmıştır. Son durumda veri setinde 27 değişken kalmıştır. Değişkenlerin 0 değeri

alması normal bir durum olmadığı için 0 değerleri eksik değere dönüştürülmüştür. Veri setinin 0.1 kuantil ve 0.9 kuantil noktaları çeyrekler açıklığı için sınır kabul edilerek aykırı değerler bulunmuştur ve bu değerler aykırı değerleri baskılama yöntemiyle düzenlenmiştir. Veri setinde kalmış olabilecek aykırı değerler için güçlü bir yöntem olan Robust ölçeklendirme yöntemi uygulanmıştır. Veri seti modelleme için %75'i eğitim %25'i test olacak şekilde iki parçaya ayrılmıştır.

3.2.2.1 Yapay Sinir Ağı ile Sınıflandırma

Yapay sinir ağı mimarilerinden olan çok katmanlı algılayıcı kullanılarak yapılan eğitimin sonucunda elde edilen değerler Tablo 3.5'te verilmiştir. Sinir ağının giriş katmanı 64 nöron, gizli katman boyutu 32 nöron, çıkış katmanı 11 nöron ve softmax aktivasyon fonksiyonu ile oluşturulmuştur. Eğitim adım sayısı 300, aktivasyon fonksiyonu ReLU, optimizatör olarak ADAM, tek seferde modele verilen örnek sayısı 128, eğitim setinin %30'u validasyon için seçilmiştir. Test verisinin doğruluğu ise %96.86'dır.

Tablo 3.5: Çok sınıflı YSA sonuçları.

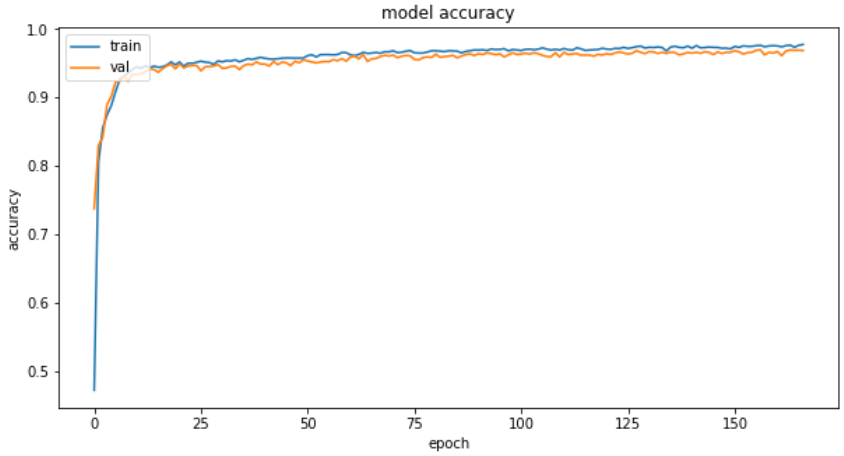
Sınıf	Kesinlik	Duyarlılık	F1-Score	Örnek Sayısı
E03_9	0.95	0.99	0.97	1249
E14_9	0.99	0.97	0.98	1184
E11_9	1.00	0.97	0.98	727
E10_9	1.00	0.99	0.99	314
E03_8	0.97	0.92	0.95	244
E66_0	0.98	0.94	0.96	134
E05_9	0.98	1.00	0.99	128
E11_7	1.00	0.98	0.99	53
E22_1	0.82	0.64	0.72	44
E66_2	1.00	0.90	0.95	41
E66_9	0.97	0.94	0.96	35

Tablo 3.6'da eğitim sonuçlarına ait karmaşıklık matrisi yer almaktadır. Bu matris her sınıfa ait hasta sayısının ne kadarının doğru tahmin edilip edilemediğini gösterir. E03_9 hastalığı için hastalığa sahip olmayan kişilerden 2884 tanesi hasta değil, 60 tanesi hasta olarak sınıflandırılmıştır. Aynı şekilde bu hastalığa sahip kişilerden 1235 tanesi hasta, 14 tanesi hasta değil olarak sınıflandırılmıştır.

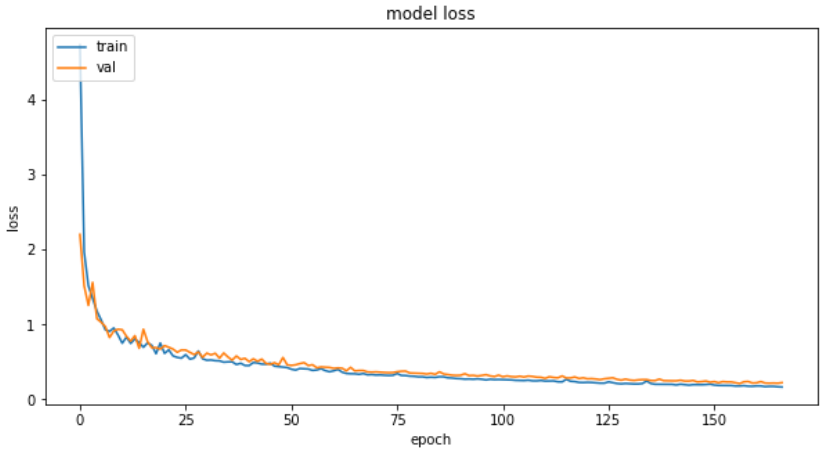
Tablo 3.6: Çok sınıflı YSA için karmaşıklık matrisi.

E03_9	**0**	**1**	**E05_9**	**0**	**1**
0	2844	60	**0**	4022	2
1	14	1235	**1**	0	128
E14_9	**0**	**1**	**E11_7**	**0**	**1**
0	2954	15	**0**	4100	1
1	38	1146	**1**	1	52
E11_9	**0**	**1**	**E22_1**	**0**	**1**
0	3424	3	**0**	4103	6
1	24	702	**1**	16	28
E10_9	**0**	**1**	**E66_2**	**0**	**1**
0	3839	0	**0**	4112	0
1	4	310	**1**	4	37
E03_8	**0**	**1**	**E66_9**	**0**	**1**
0	3902	7	**0**	4117	1
1	19	225	**1**	2	33
E66_0	**0**	**1**			
0	4017	2			
1	8	126			

Şekil 3.4 ve Şekil 3.5'te eğitim ve validasyon setlerinin doğruluk ve kayıp grafikleri yer almaktadır. Eğitim ve validasyon setlerine ait kayıplar birlikte azalmaktadır.

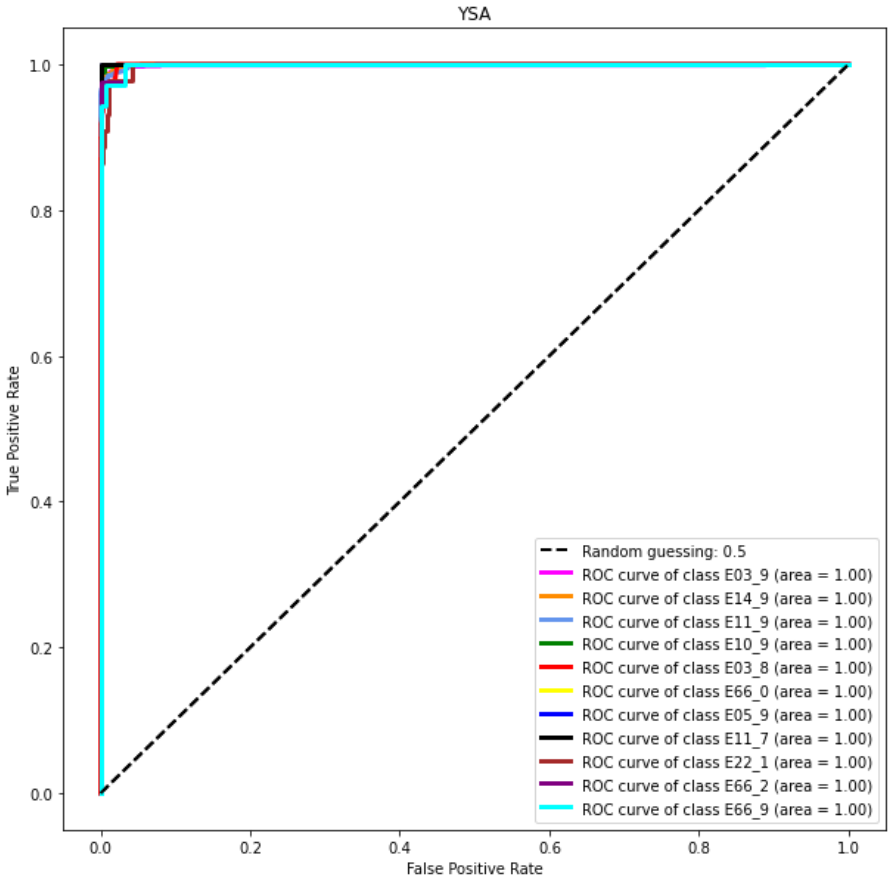


Şekil 3.4: Çok sınıflı YSA eğitim ve doğrulama setleri için doğruluk grafiği.



Şekil 3.5: Çok sınıflı YSA eğitim ve doğrulama setleri için kayıp grafiği.

Şekil 3.6'da eğitilen modelde bulunan 11 hastalık ait ROC eğrileri görülmektedir. Her sınıf için eğri altında kalan alan 1 olduğundan yapay sinir ağının sınıflandırmada başarılı olduğu görülmektedir.



Şekil 3.6: Çok sınıflı YSA için ROC eğrileri.

3.2.2.2 Destek Vektör Makineleri ile Sınıflandırma

Destek vektör makineleri ile yapılan eğitimin sonucunda test setinde elde edilen performansın sonuçları Tablo 3.7'de görülmektedir. En iyi sonuçlar için GridSearch ile yapılan en iyi hiperparametre aramasında SVM için C ceza katsayısı 100, gamma katsayısı 0.001, çekirdek seçimi ise RBF olarak seçilmiştir. 5 katlı çapraz doğrulama sonucundaki ortalama doğruluk %94,94'tür. Test verisi doğruluğu ise %95,4'tür.

Tablo 3.7: Çok sınıflı SVM sonuçları.

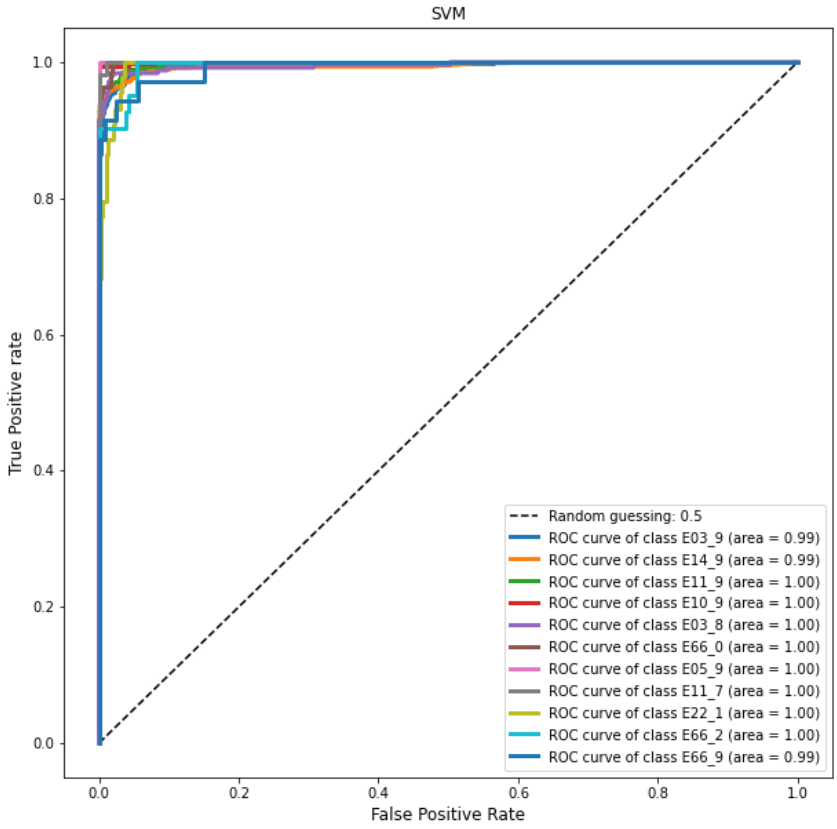
Sınıf	Kesinlik	Duyarlılık	F1-Score	Örnek Sayısı
E03_9	0.92	0.97	0.94	1249
E14_9	0.97	0.95	0.96	1184
E11_9	0.98	0.95	0.97	727
E10_9	0.99	0.99	0.99	314
E03_8	0.96	0.92	0.94	244
E66_0	1.00	0.93	0.97	134
E05_9	1.00	0.95	0.97	128
E11_7	1.00	0.96	0.98	53
E22_1	0.75	0.68	0.71	44
E66_2	0.93	0.90	0.91	41
E66_9	0.97	0.86	0.91	35

Tablo 3.8'de test verisine ait karmaşıklık matrisi yer almaktadır. Tablo 12'deki sınıf sırasına göre karmaşıklık matrisi 0-10 arası numaralandırılmıştır. Bu karmaşıklık matrisindeki 0. yani E03_9 hastalığına sahip 1249 kişiden 1211 kişi doğru sınıflandırılmıştır. 24 kişi 1. hastalık sınıfına, 6 kişi 2. hastalık sınıfına, 4 kişi 4. hastalık sınıfına ve 1'er kişi de 8.ve 9. Hastalık sınıflarına ait olarak sınıflandırılmıştır.

Tablo 3.8: Çok sınıflı SVM için karmaşıklık matrisi.

		Tahmin Edilen Değerler											Örnek Sayısı
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
Gerçek Değerler	**0**	1211	24	6	0	6	0	0	0	1	1	0	1249
	1	44	1129	5	0	2	0	0	0	3	1	0	1184
	2	20	8	693	1	0	0	0	0	5	0	0	727
	3	1	0	1	311	0	0	0	0	0	0	1	314
	4	18	2	0	0	224	0	0	0	0	0	0	244
	5	8	0	1	0	0	125	0	0	0	0	0	134
	6	7	0	0	0	0	0	121	0	0	0	0	128
	7	0	1	0	0	1	0	0	51	0	0	0	53
	8	10	2	1	0	0	0	0	0	30	1	0	44
	9	2	1	0	0	0	0	0	0	1	37	0	41
	10	1	0	2	2	0	0	0	0	0	0	30	35

Şekil 3.7'de destek vektör makineleriyle eğitim sonucu sınıflara ait ROC eğrileri görülmektedir. AUC değerlerine göre sınıflandırıcının fazlasıyla başarılı olduğu görülmüştür.



Şekil 3.7: Çok sınıflı SVM için ROC eğrileri.

3.2.2.3 Hafif Gradyan Artırma Makineleri ile Sınıflandırma

Hafif gradyan artırma makineleri ile yapılan eğitimde kullanılacak olan hiper parametrelerin en optimal değerlerini bulunması için 5 katlı GridSearch araması yapılmıştır. Arama sonucunda öğrenme adımı için 0.01 ve kullanılacak karar ağacı sayısı için 280 değeri bulunmuştur. Bulunan hiper parametrelerle model güncellenmiştir. Uygulanan 5 katlı çapraz doğrulama sonucu doğruluk %99.91 olarak bulunmuştur. Elde edilen sonuçlar Tablo 3.9'da verilmiştir. Test verisinde %99.92 doğruluk elde edilmiştir.

Tablo 3.9: Çok sınıflı LightGBM sonuçları.

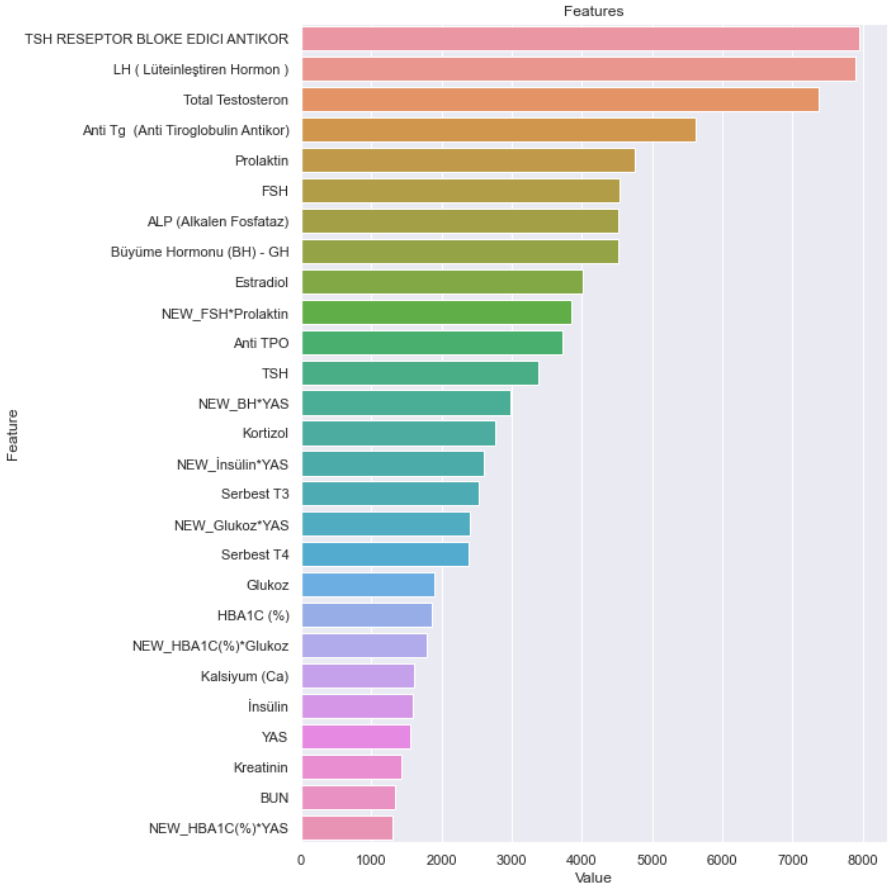
Sınıf	Kesinlik	Duyarlılık	F1-Score	Örnek Sayısı
E03_9	1.00	1.00	1.00	1249
E14_9	1.00	1.00	1.00	1184
E11_9	1.00	1.00	1.00	727
E10_9	1.00	1.00	1.00	314
E03_8	1.00	1.00	1.00	244
E66_0	1.00	0.99	1.00	134
E05_9	1.00	0.99	1.00	128
E11_7	1.00	1.00	1.00	53
E22_1	1.00	1.00	1.00	44
E66_2	1.00	1.00	1.00	41
E66_9	1.00	1.00	1.00	35

Tablo 3.10'da test verisine ait karmaşıklık matrisi yer almaktadır. Bu karmaşıklık matrisindeki 5. yani E66_0 hastalığına sahip 134 kişiden 133 kişi doğru sınıflandırılmıştır.

Tablo 3.10: Çok sınıflı LightGBM için karmaşıklık matrisi.

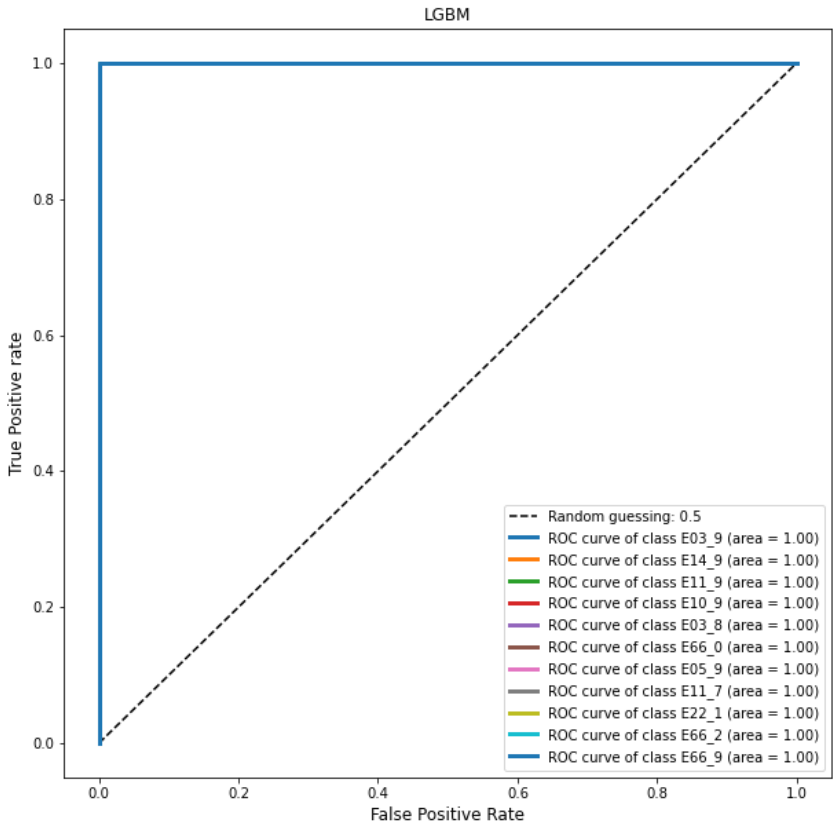
		Tahmin Edilen Değerler											Örnek Sayısı
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
Gerçek Değerler	**0**	1248	0	0	0	1	0	0	0	0	0	0	1249
	1	0	1184	0	0	0	0	0	0	0	0	0	1184
	2	0	0	727	0	0	0	0	0	0	0	0	727
	3	0	0	0	314	0	0	0	0	0	0	0	314
	4	0	0	0	0	244	0	0	0	0	0	0	244
	5	0	1	0	0	0	133	0	0	0	0	0	134
	6	0	1	0	0	0	0	127	0	0	0	0	128
	7	0	0	0	0	0	0	0	53	0	0	0	53
	8	0	0	0	0	0	0	0	0	44	0	0	44
	9	0	0	0	0	0	0	0	0	0	41	0	41
	10	0	0	0	0	0	0	0	0	0	0	35	35

Şekil 3.8'de modeli etkileyen değişkenlerin önemli olma değerleri çoktan aza doğru sıralaması yer almaktadır.



Şekil 3.8: LightGBM için değişkenlerin önem sırası.

Şekil 3.9'da eğitilen LightGBM'e ait ROC eğrileri görülmektedir. Bu veride LightGBM tüm sınıflar için olabilecek en iyi sonuçları vermiştir.



Şekil 3.9: Çok sınıflı LightGBM için ROC eğrileri.

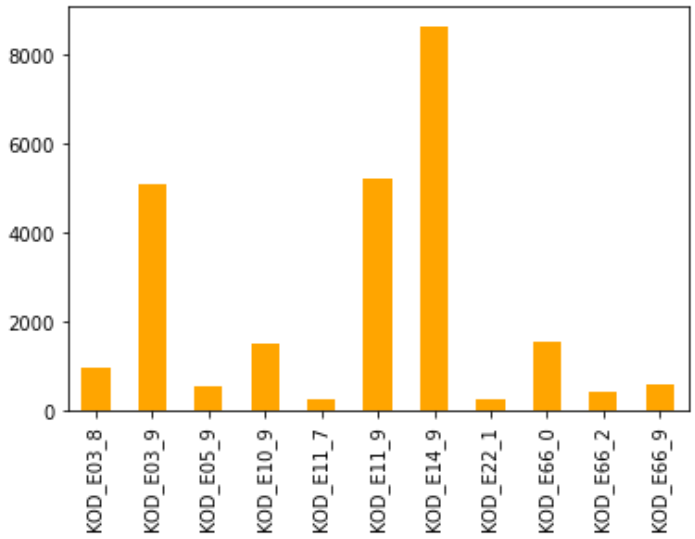
3.2.3 Çok Etiketli Sınıflandırma Sonuçları

Veri setinde bir kişinin birden fazla hastalığa sahip olması bu verinin çok etiketli olacağına işaret eder. Bu sebeple veri setindeki kişilere etiketlenen hastalıklar işlem tarihlerine göre tek bir satırda toplanmıştır. Veri setinin hedef değişkeni iki boyutlu ve 0-1 değerlerinden oluşacak şekilde düzenlenmiştir. Yeni düzende hastalıklara ait kişi sayısı Tablo 3.11'de verilmiştir.

Tablo 3.11: Çok sınıflı sınıflandırma verisinde sınıflara ait gözlem sayıları.

ICD-10 Kodları	Kişi Sayısı
E03.8	975
E03.9	5103
E05.9	550
E10.9	1479
E11.7	260
E11.9	5223
E14.9	8633
E22.1	263
E66.0	1542
E66.2	398
E66.9	575

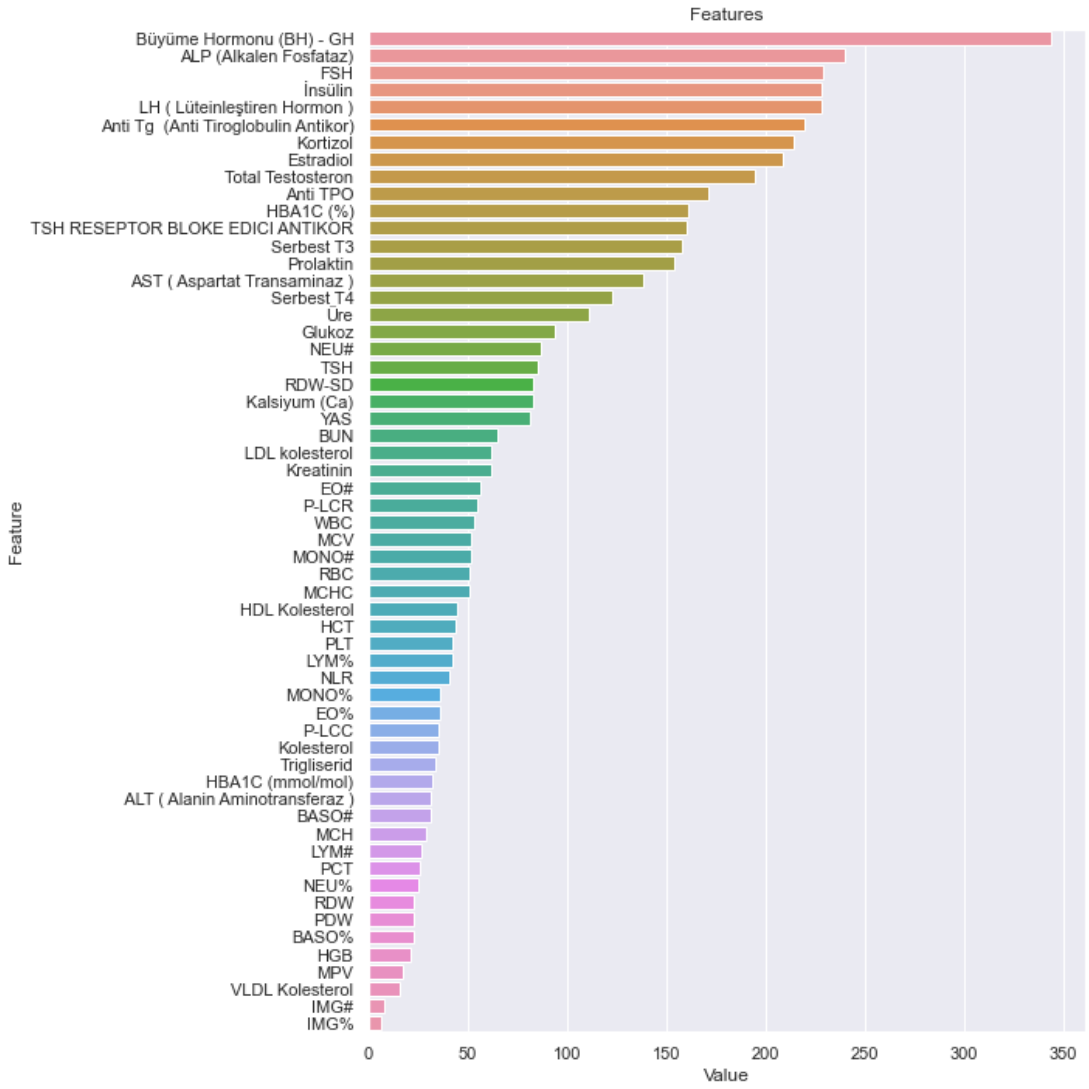
Hastalıklara ait kişi dağılımı Şekil 3.10'da verilmiştir.



Şekil 3.10: Etiketlerdeki hasta sayısı dağılımı.

Veri setinde 16621 gözlem 77 değişken bulunmaktadır. Bu değişkenlerden 11 tanesi hedef değişken uzayını oluşturmaktadır. Veri setine ait etiket kardinalitesi 1.504, etiket yoğunluğu 0.136 olarak hesaplanmıştır. Veri setinde bilgi taşımayan sütunlar atılmıştır. Değişkenlerin 0 değeri alması normal bir durum olmadığı için 0 değerleri eksik değere dönüştürülmüştür. Veri setinin 0.1 kuantil ve 0.9 kuantil noktaları çeyrekler açıklığı için sınır kabul edilerek aykırı değerler bulunmuştur ve bu değerler aykırı değerleri baskılama yöntemiyle düzenlenmiştir. Veri setinde fazlaca eksik veri bulunduğu için bunun üstesinden gelebilmek için hedef değişken

kümesi oluşturulmuştur ve her eşsiz kümeye göre her değişkenin var olan verilerinin ortanca değeri ile doldurulma işlemi yapılmıştır. Veri setinden çıkarılacak değişkenlere hafif gradyan artırma makineleri ile kurulan temel bir model üzerinden elde edilen Şekil 3.11'deki değişkenlerin önem sıralaması grafiğine bakılarak karar verilmiştir.



Şekil 3.11: Çok etiketli için kurulan temel modele göre değişkenlerin önem sırası.

Veri setinden 'BASO#', 'VLDL Kolesterol', 'BASO%', 'PDW', 'PCT', 'LYM#', 'MCH', ' ALT (Alanin Aminotransferaz)', 'HBA1C (mmol/mol)', 'Trigliserid', 'Kolesterol', 'P-LCC', 'P-LCR', 'EO#', 'EO%', 'MONO#', 'MONO%', 'NLR', 'HDL Kolesterol', 'MPV', 'HCT', 'HGB', 'MCHC', 'MCV', 'RDW', 'WBC', 'PLT', 'RBC', 'IMG#', 'IMG%', 'NEU%', 'LYM%' sütunları çıkarılmıştır. Son durumda veri setinde 26 değişken kalmıştır. Şekil 3.10'de yer alan korelasyon

matrisi yardımıyla modeli olumlu etkileyebileceği düşünülen yeni değişkenler elde edilmiştir. Elde edilen değişkenlerin ham veri setinde olan değişkenlerle karışmaması için yeni değişkenlerin başına NEW yazısı eklenmiştir. Çok etiketli sınıflandırma için kullanılan veriler çok sınıflı sınıflandırma için kullanılan verilerle aynı olduğundan Şekil 3.2'de yer verilmiş olan korelasyon matrisleri de aynıdır.

Veri setinde kalmış olabilecek aykırı değerler için güçlü bir yöntem olan Robust ölçeklendirme yöntemi uygulanmıştır. Veri seti modelleme için %75 eğitim %25 test olacak şekilde iki parçaya ayrılmıştır.

3.2.3.1 Yapay Sinir Ağı ile Sınıflandırma

Yapay sinir ağı mimarilerinden olan çok katmanlı algılayıcı kullanılarak yapılan eğitimin sonucunda elde edilen değerler Tablo 3.12'de verilmiştir. Sinir ağının giriş katmanı 70, gizli katman boyutu 40 ve çıkış katmanı 11 nörondan oluşmaktadır. Eğitim adım sayısı 500, tek seferde modele verilen örnek sayısı 128, çıkış katmanında sigmoid, diğer katmanlarda ReLU aktivasyon fonksiyonu, optimizatör ise ADAM seçilmiştir. Eğitim setinin %30'u validasyon seti için ayrılmıştır. Aşırı öğrenmeyi engellemek için erken durdurma yapılmıştır. Hamming kaybı 0.014 bulunmuştur. Test setinin doğruluğu %91.21'dir.

Tablo 3.12: Çok etiketli YSA sonuçları.

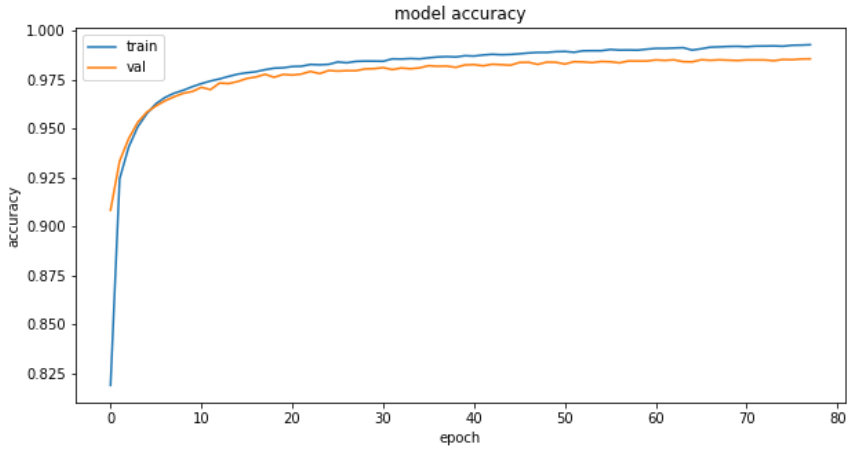
Sınıf	Kesinlik	Duyarlılık	F1-Score	Örnek Sayısı
E03_8	0.94	0.81	0.87	250
E03_9	0.94	0.96	0.95	1301
E05_9	0.96	0.90	0.93	132
E10_9	0.96	0.94	0.95	355
E11_7	0.92	0.67	0.77	66
E11_9	0.98	0.95	0.96	1286
E14_9	0.97	0.97	0.97	2118
E22_1	0.82	0.60	0.69	68
E66_0	0.94	0.88	0.91	394
E66_2	0.82	0.70	0.76	80
E66_9	0.93	0.79	0.86	145

Her etiket için elde edilen karmaşıklık matrisi Tablo 3.13'te yer almaktadır. E03_8 hastalığı için hastalığa sahip olmayan kişilerden 3893 tanesi hasta değil, 13 tanesi hasta olarak sınıflandırılmıştır. Aynı şekilde bu hastalığa sahip kişilerden 203 tanesi hasta, 47 tanesi hasta değil olarak sınıflandırılmıştır.

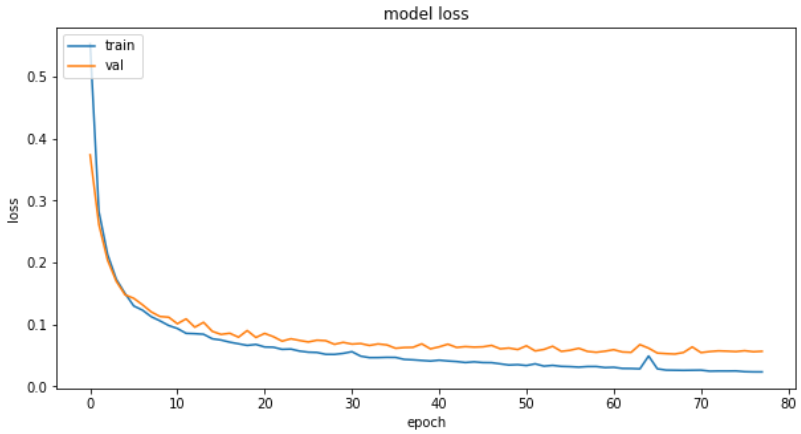
Tablo 3.13: Çok etiketli YSA için karmaşıklık matrisi.

E03_8	**0**	**1**	**E14_9**	**0**	**1**
0	3893	13	**0**	1971	67
1	47	203	**1**	59	2059
E03_9	**0**	**1**	**E22_1**	**0**	**1**
0	2779	76	**0**	4079	9
1	49	1252	**1**	27	41
E05_9	**0**	**1**	**E66_0**	**0**	**1**
0	4019	5	**0**	3740	22
1	13	119	**1**	47	347
E10_9	**0**	**1**	**E66_2**	**0**	**1**
0	3788	13	**0**	4064	12
1	22	333	**1**	24	56
E11_7	**0**	**1**	**E66_9**	**0**	**1**
0	4086	4	**0**	4003	8
1	22	44	**1**	30	115
E11_9	**0**	**1**			
0	2841	29			
1	62	1224			

Şekil 3.12 ve Şekil 3.13'te eğitim ve validasyon setlerinin doğruluk ve kayıp grafikleri yer almaktadır. Eğitim ve validasyon setlerinin kayıp değeri birlikte düştüğü görülmektedir. Bir noktadan sonra validasyon kaybı arttığı için model eğitimi erken durdurma yapılarak durdurulmuş olduğu görülmektedir.

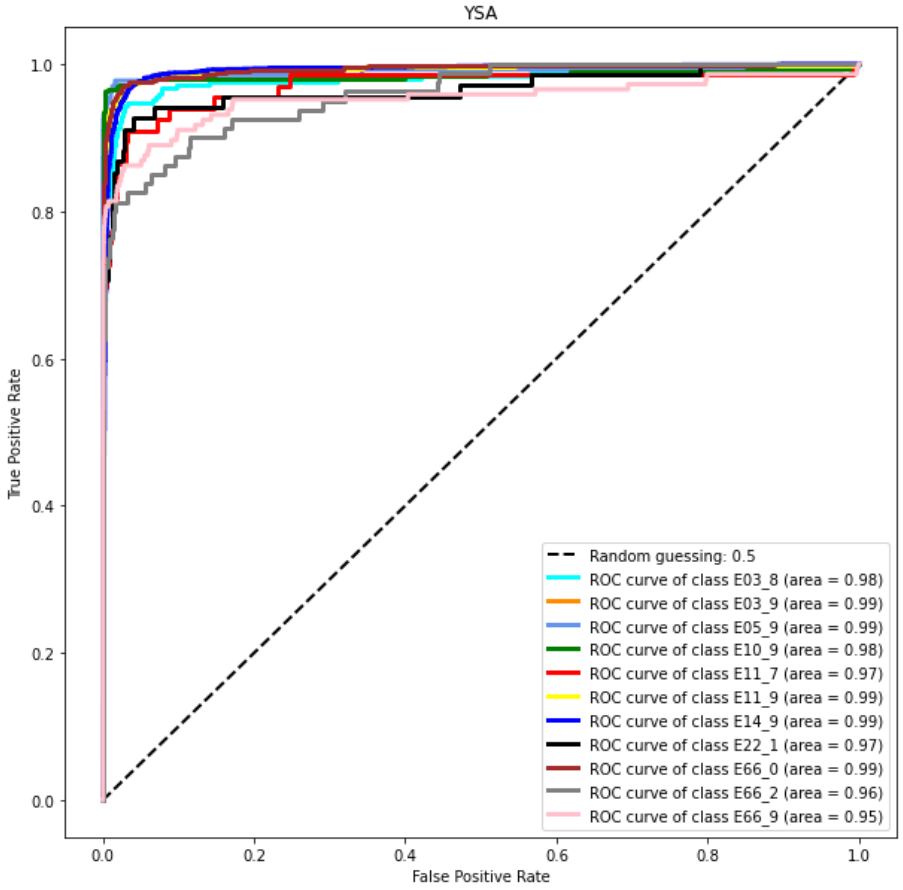


Şekil 3.12: Çok etiketli YSA için eğitim ve doğrulama setlerinin doğruluk grafiği.



Şekil 3.13: Çok etiketli YSA için eğitim ve doğrulama setleri kayıp grafiği.

Şekil 3.14'te modele ait ROC eğrileri görülmektedir. Eğri altında kalan alanlardan da görülmektedir ki E66_2, E66_9 etiketleri diğer etiketlere göre daha az başarılı bir şekilde tahmin edilmiştir ancak sınıflandırıcının genel performansı başarılıdır.



Şekil 3.14: Çok etiketli YSA için ROC eğrileri.

3.2.3.2 Destek Vektör Makineleri ile Sınıflandırma

Destek vektör makineleri ile yapılan eğitimin sonucunda test setinde elde edilen performansın sonuçları Tablo 3.14'te görülmektedir. En iyi sonuçlar için GridSearch ile yapılan en iyi hiper parametre aramasında SVM için C ceza katsayısı 100, gamma katsayısı 0.001 ve çekirdek seçimi ise RBF seçilmiştir. 5 katlı çapraz doğrulama sonucu ortalama doğruluk %91.1'dir. Hamming kaybı 0.013 bulunmuştur. Test setinin doğruluğu %92.13'dür.

Tablo 3.14: Çok etiketli SVM sonuçları.

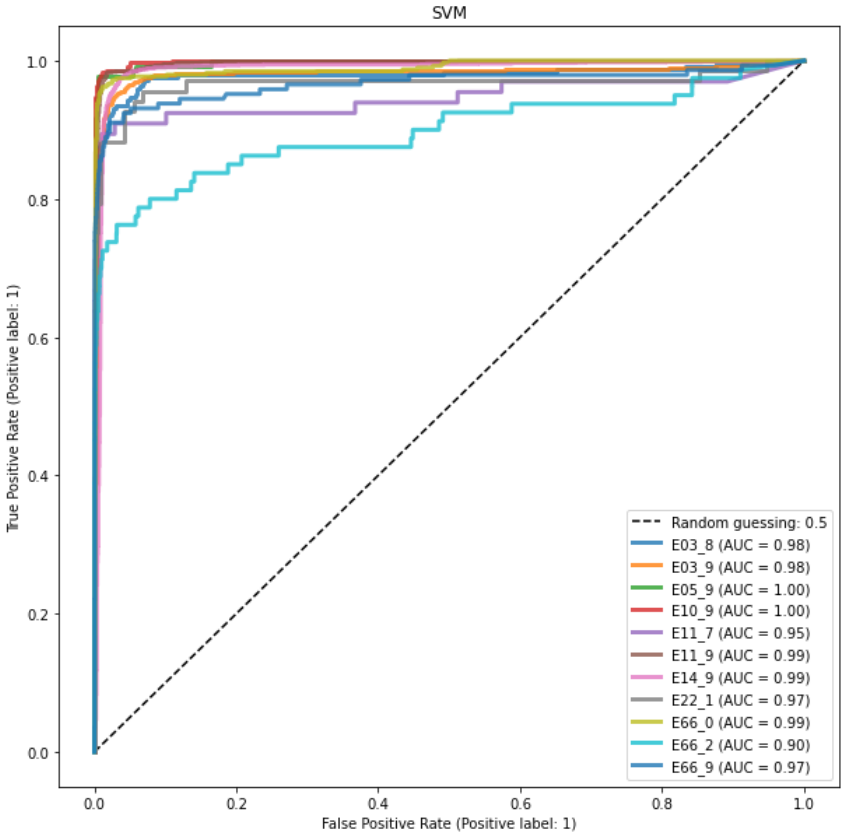
Sınıf	Kesinlik	Duyarlılık	F1-Score	Örnek Sayısı
E03_8	0.97	0.85	0.90	250
E03_9	0.94	0.94	0.94	1301
E05_9	0.98	0.90	0.94	132
E10_9	0.96	0.96	0.96	355
E11_7	0.90	0.70	0.79	66
E11_9	0.97	0.97	0.97	1286
E14_9	0.96	0.97	0.97	2118
E22_1	0.90	0.68	0.77	68
E66_0	0.95	0.94	0.95	394
E66_2	0.82	0.59	0.69	80
E66_9	0.93	0.77	0.85	145

Her etiket için elde edilen karmaşıklık matrisi Tablo 3.15'te yer almaktadır. E14_9 hastalığı için hastalığa sahip olmayan kişilerden 1960 tanesi hasta değil, 78 tanesi hasta olarak sınıflandırılmıştır. Aynı şekilde bu hastalığa sahip kişilerden 2064 tanesi hasta, 54 tanesi hasta değil olarak sınıflandırılmıştır.

Tablo 3.15: Çok etiketli SVM için karmaşıklık matrisi.

E03_8	**0**	**1**	**E14_9**	**0**	**1**
0	3899	7	**0**	1960	78
1	38	212	**1**	54	2064
E03_9	**0**	**1**	**E22_1**	**0**	**1**
0	2779	76	**0**	4083	5
1	74	1227	**1**	22	46
E05_9	**0**	**1**	**E66_0**	**0**	**1**
0	4022	2	**0**	3742	20
1	13	119	**1**	23	371
E10_9	**0**	**1**	**E66_2**	**0**	**1**
0	3788	13	**0**	4066	10
1	13	342	**1**	33	47
E11_7	**0**	**1**	**E66_9**	**0**	**1**
0	4085	5	**0**	4003	8
1	20	46	**1**	35	112
E11_9	**0**	**1**			
0	2829	41			
1	33	1253			

Şekil 3.15'te modele ait ROC eğrileri görülmektedir. Sınıflandırıcı E11.7, E66.2, E66.9, E22.1 etiketlerinde diğer etiketlere göre daha az başarılı olduğu görülmektedir.



Şekil 3.15: Çok etiketli SVM için ROC eğrileri.

3.2.3.3 Hafif Gradyan Artırma Makineleri ile Sınıflandırma

Hafif gradyan artırma makineleri ile yapılan eğitimde kullanılacak olan hiper parametrelerin en optimal değerlerini bulunması için 5 katlı GridSearch araması yapılmıştır. Arama sonucunda kullanılacak karar ağacı sayısı için 270 değeri ve artırma tipi olarak GOSS yöntemi belirlenmiştir. Belirlenen bu hiper parametrelerle model güncellenmiştir. Uygulanan 5 katlı çapraz doğrulama sonucu ortalama doğruluk %98.33 olarak bulunmuştur. Elde edilen sonuçlar Tablo 3.16'da verilmiştir. Test verisinde %98.21 doğruluk elde edilmiştir. Hamming kaybı 0.0029 bulunmuştur.

Tablo 3.16: Çok etiketli LGBM sonuçları.

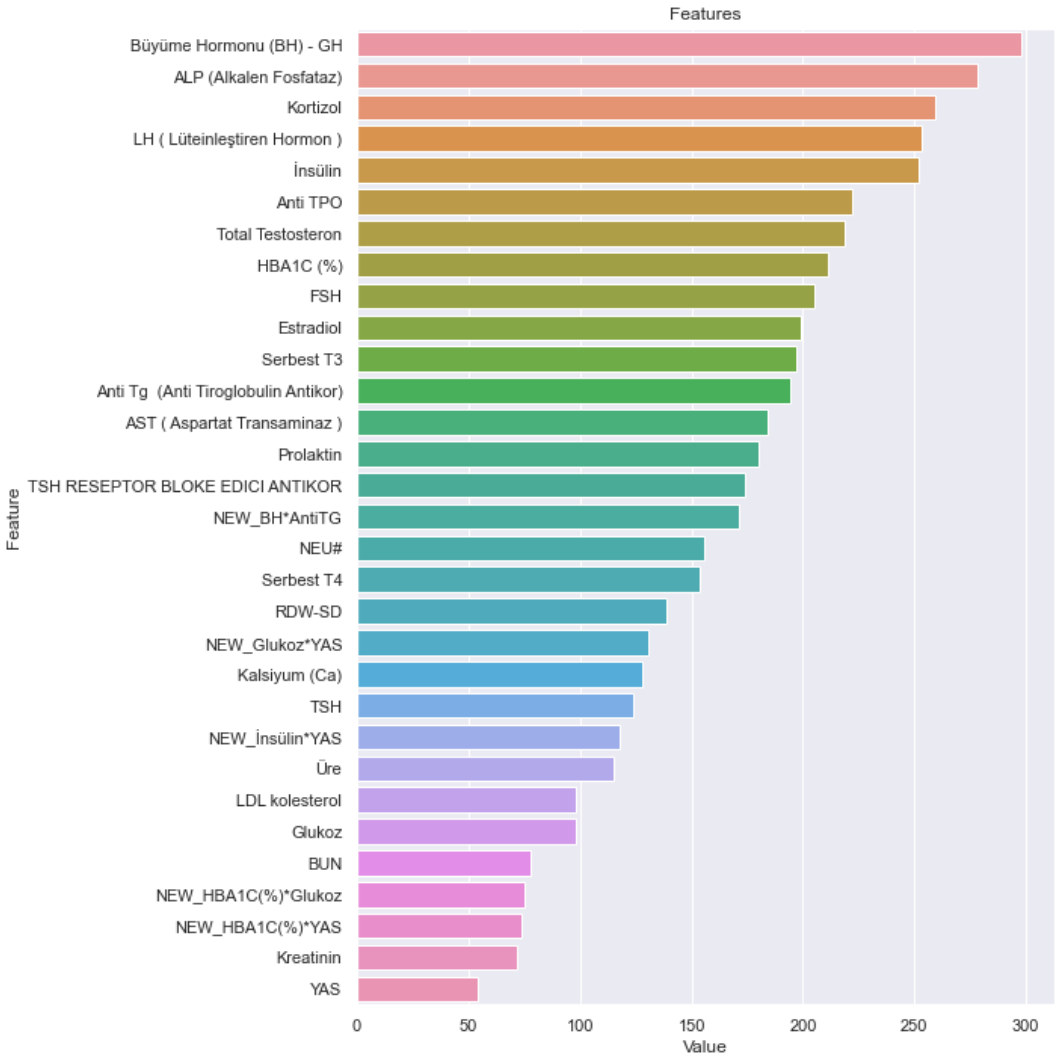
Sınıf	Kesinlik	Duyarlılık	F1-Score	Örnek Sayısı
E03_8	0.99	0.96	0.98	250
E03_9	0.99	0.99	0.99	1301
E05_9	1.00	0.96	0.98	132
E10_9	1.00	0.98	0.99	355
E11_7	0.93	0.85	0.89	66
E11_9	1.00	0.99	0.99	1286
E14_9	1.00	1.00	1.00	2118
E22_1	0.98	0.88	0.93	68
E66_0	0.99	0.97	0.98	394
E66_2	1.00	0.85	0.92	80
E66_9	0.98	0.90	0.94	145

Her etiket için elde edilen karmaşıklık matrisi Tablo 3.17'de yer almaktadır. E66_9 hastalığı için hastalığa sahip olmayan kişilerden 4009 tanesi hasta değil, 2 tanesi hasta olarak sınıflandırılmıştır. Aynı şekilde bu hastalığa sahip kişilerden 131 tanesi hasta, 14 tanesi hasta değil olarak sınıflandırılmıştır.

Tablo 3.17: Çok etiketli LGBM için karmaşıklık matrisi.

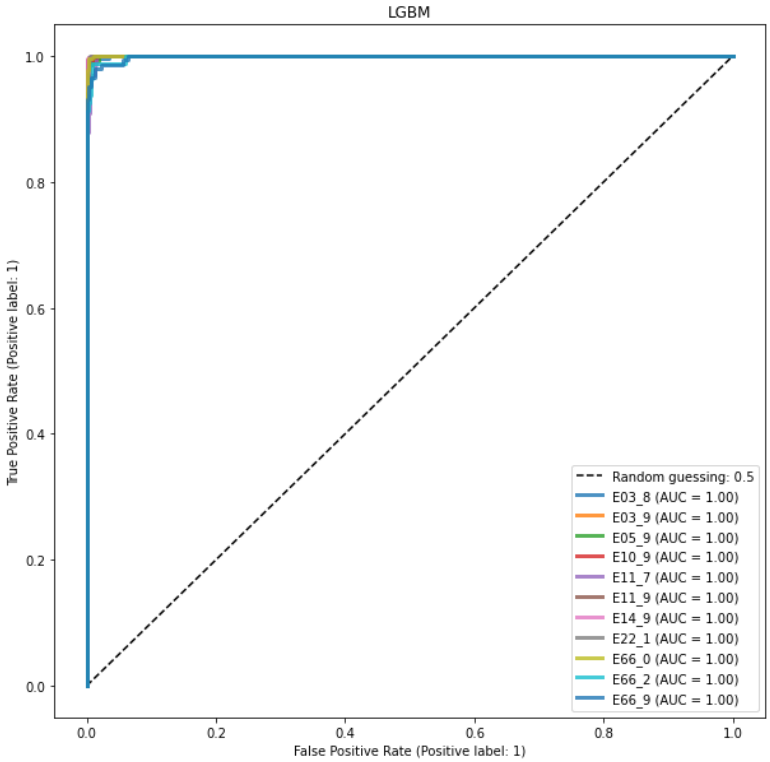
E03_8	**0**	**1**	**E14_9**	**0**	**1**
0	3904	2	**0**	2030	8
1	9	241	**1**	4	2114
E03_9	**0**	**1**	**E22_1**	**0**	**1**
0	2847	8	**0**	4087	1
1	11	1290	**1**	8	60
E05_9	**0**	**1**	**E66_0**	**0**	**1**
0	4024	0	**0**	3758	4
1	5	127	**1**	12	382
E10_9	**0**	**1**	**E66_2**	**0**	**1**
0	3800	1	**0**	4076	0
1	6	349	**1**	12	68
E11_7	**0**	**1**	**E66_9**	**0**	**1**
0	4086	4	**0**	4009	2
1	10	56	**1**	14	131
E11_9	**0**	**1**			
0	2865	5			
1	9	1277			

Şekil 3.16'da modeli etkileyen değişkenlerin sıralaması verilmiştir. Oluşturulan yeni değişkenlerin modeli olumlu etkilediği görülmektedir.



Şekil 3.16: LGBM için değişkenlerin önem sıralaması.

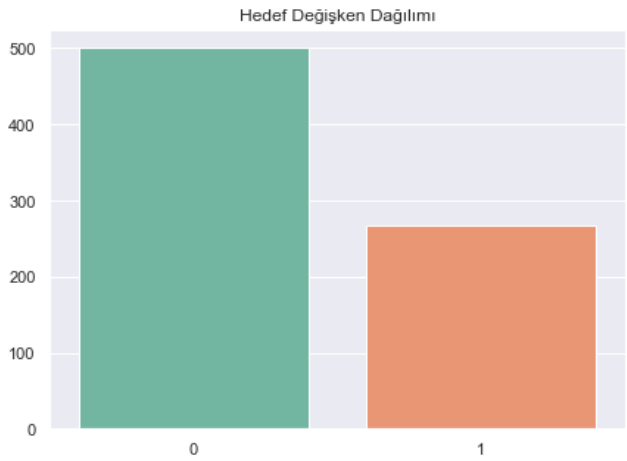
Şekil 3.17'de modele ait ROC eğrileri görülmektedir. Eğrilerin altında kalan alanlar 1 olduğu için eğitilen model bu veri seti için başarılı olmuştur.



Şekil 3.17: Çok etiketli LightGBM için ROC eğrileri.

3.3 Pima Indians Diabetes Veri Seti

Pima Indians Diabetes veri seti ABD'deki Ulusal Diyabet-Sindirim-Böbrek Hastalıkları Enstitüleri'nde tutulan büyük veri setinin parçasıdır. ABD'deki Arizona Eyaleti'nin en büyük 5. şehri olan Phoenix şehrinde yaşayan 21 yaş ve üzerinde olan Pima Indian kadınları üzerinde yapılan diyabet araştırması için kullanılan verilerdir. Veri setinde 9 değişken, 768 gözlem bulunmaktadır. Hedef değişken "Outcome" olup, 1 değeri olması diyabet test sonucunun pozitif oluşunu, 0 değeri olması ise negatif oluşunu belirtmektedir. Hedef değişkeni Outcome, 500'ü hasta değil ve 268'i hasta olarak etiketlidir. Şekil 3.18'de sınıf dağılımının grafiği verilmiştir.



Şekil 3.18: Hedef değişkenin sınıf dağılımı.

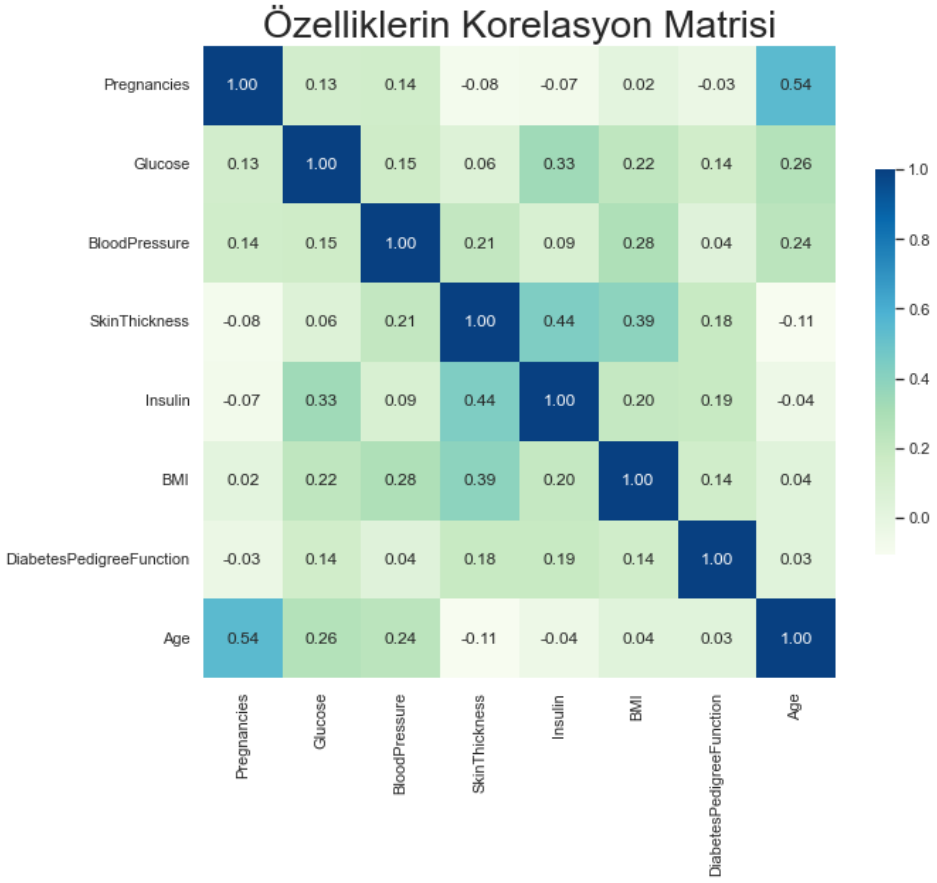
Tablo 3.18'de 9 değişkenlere ait açıklamalar yer almaktadır.

Tablo 3.18: Veri setine ait değişkenlerin açıklamaları.

Özellikler	Tanımlar
Pregnancies	Hamilelik sayısı
Glucose	Oral glikoz tolerans testinde 2 saatlik plazma glikoz konsantrasyonu
Blood Pressure	Kan basıncı (Küçük tansiyon) (mm Hg)
SkinThickness	Cilt Kalınlığı
Insulin	2 saatlik serum insülini (mu U/ml)
DiabetesPedigreeFunction	Soydaki kişilere göre diyabet olma ihtimalini hesaplayan fonksiyon
BMI	Vücut kitle indeksi
Age	Yaş
Outcome	Hastalığa sahip (1) ya da değil (0)

Veri seti 1 tane kategorik, 8 tane nümerik kolona sahiptir. Veri setindeki 9 değişkenden 2'si (BMI ve DiabetesPedigreeFucntion) float64, diğerleri int64 değerlere sahiptir ve veri setinde eksik değer bulunmamaktadır. Veri setinde eksik değer olmadığı görülmüştür ancak değişkenlerin minimum değerleri incelenildiğinde Glucose, BloodPressure, SkinThickness, Insulin, BMI sütunlarında 0 değerleriyle karşılaşılmıştır. BMI değeri 12'den küçük olan, BloodPressure değeri 60'tan küçük olan değerler ile diğer değişkenlerde yaşayan bir insan için bu değişkenlerin 0 değerinde olması normal olmadığı için bu değerler 0 yerine eksik değer olarak değiştirilmiştir. Oluşan eksik değerler ise değişkenlerin hedef değişken bazındaki kırılımlarına bakılarak ortanca değerleriyle doldurulmuştur.

Veri setinde yer alan özelliklerin hedef değişkene olan etkileri incelenilmiştir. Şekil 3.19'da yer alan korelasyon matrisi yardımıyla değişkenlerin birbirleriyle olan çarpımlarından yeni değişkenler elde edilmiştir. Glucose, Age, BloodPressure, Insulin, BMI değişkenleri referans değer aralıklarına göre kategorileştirilerek yeni değişkenler oluşturulmuştur. Bu yeni değişkenlere One Hot Encoding (OHE) uygulanmıştır. Ancak bu işlemin modellere yeterli katkı sağlamadığı gözlemlenerek veri setinin boyutunu büyütmemek amacıyla kategorileştirilerek oluşturulan değişkenler veri setinden çıkarılmıştır.



Şekil 3.19: Veri setine ait korelasyon matrisi.

Veri setinde farklı oranlardaki aralık değerleri incelenildiğinde Insülin ve Glucose dışında aykırı değere rastlanılmamıştır. Bu iki değişkenlerin 0.10'luk diliminin altında ve 0.90'lık diliminin üstünde kalan örnekler baskılama yöntemi ile değiştirilmiştir. Veri setinde aykırı değerlerden daha az etkilenilmesi için Robust ölçeklendirme yöntemi kullanılmıştır. Veri seti 576'sı eğitim, 192'si test olacak şekilde ayrılmıştır.

3.3.1 Yapay Sinir Ağı ile Sınıflandırma

Yapay sinir ağının giriş katmanında 32 adet, gizli katmanlarda sırasıyla 16 ve 8 adet, çıkış katmanında ise 1 adet nöron kullanılmıştır. Giriş katmanında ve gizli katmanlarda ReLU aktivasyon fonksiyonu, çıkış katmanında ise sigmoid aktivasyon fonksiyonu, optimizasyon için ise öğrenme adımı 0.01 seçilerek Adam optimizasyon yöntemi kullanılmıştır.

Validasyon için eğitim setinin %20'si alınmıştır. Eğitim adım sayısı 500 olarak belirlenmiştir ancak arka arkaya 30 eğitim sonucunda daha iyi bir sonuç olmaması durumunda erken durdurma yapılmıştır. Elde edilen doğruluk %86.45'tir. YSA ile yapılan eğitimin sonucunda test setinde elde edilen performansın sonuçları Tablo 3.19'da görülmektedir.

Tablo 3.19: İkili sınıflandırma için YSA sonuçları.

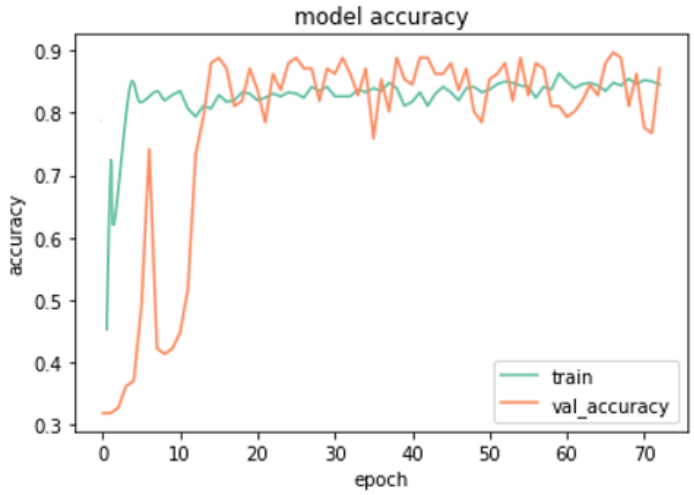
Sınıf	Kesinlik	Duyarlılık	F1-Score
Sınıf 0	0.91	0.88	0.89
Sınıf 1	0.80	0.85	0.82

Tablo 3.20'de test verisine ait karmaşıklık matrisi yer almaktadır. Hasta olmayan 121 kişiden 106'sı, hasta olan 71 kişiden 60'ı doğru tahmin edilmiştir.

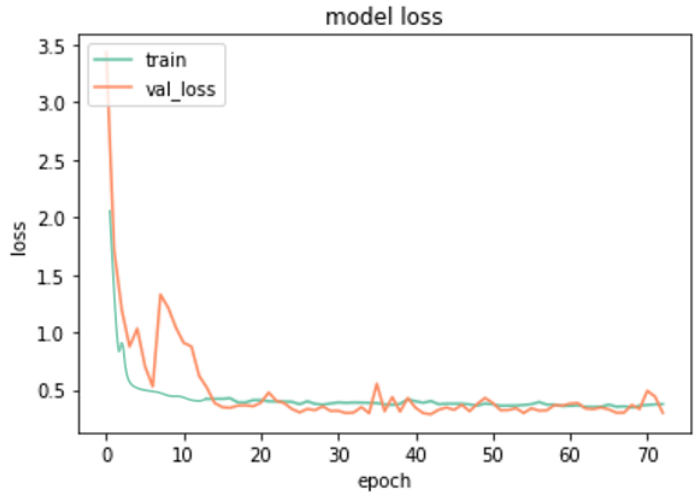
Tablo 3.20: İkili sınıflandırma için YSA karmaşıklık matrisi.

		Tahmin Edilen Değerler		Örnek Sayısı
		0	**1**	
Gerçek Değerler	**0**	106	15	121
	1	11	60	71

Şekil 3.20 ve Şekil 3.21'de eğitim ve validasyon setlerinin doğruluk ve kayıp grafikleri yer almaktadır.

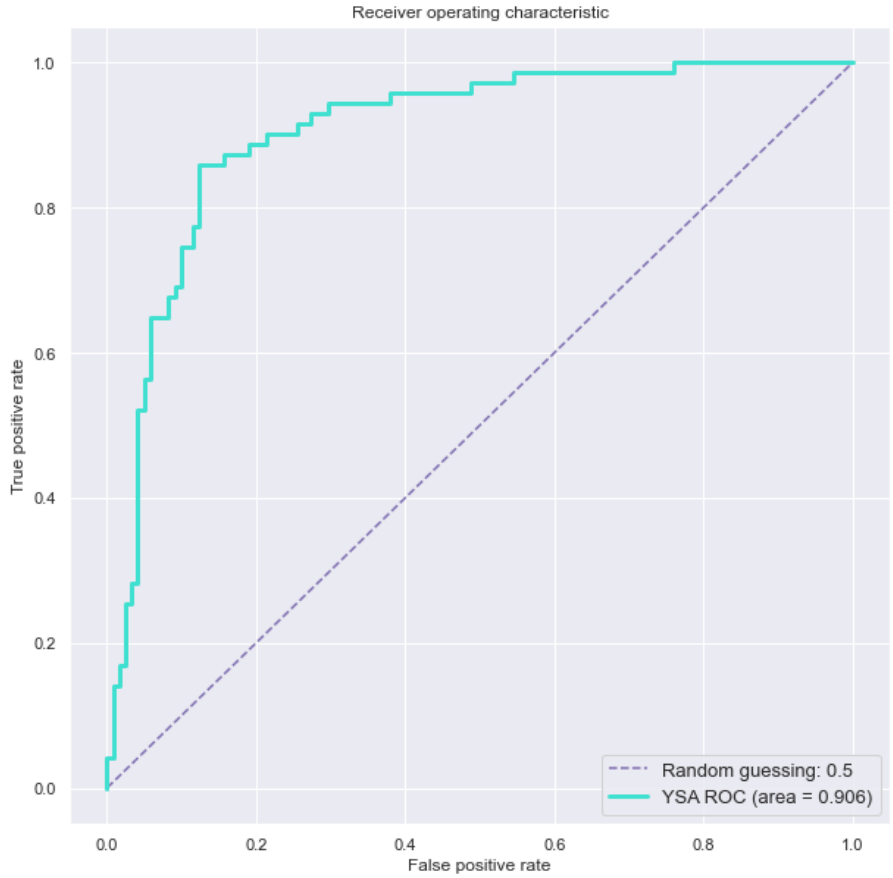


Şekil 3.20: Pima Indian Diabetes eğitim ve doğrulama setleri için doğruluk grafiği.



Şekil 3.21: Pima Indian Diabetes eğitim ve doğrulama setleri için kayıp grafiği.

Şekil 3.22'de modele ait ROC eğrisi görülmektedir. Eğrinin altında kalan alan 0.906 olarak bulunmuştur. Bu alan 1'e yakın olduğu için model başarılıdır.



Şekil 3.22: Yapay sinir ağına ait ROC eğrisi.

3.3.2 Destek Vektör Makineleri ile Sınıflandırma

Veri seti üzerinde yapılan veri analizi ve özellik mühendisliğinden sonra destek vektör makineleri ile yapılan eğitimin sonucunda test setinde elde edilen performansın sonuçları Tablo 3.21'de görülmektedir. Buradaki sonuçlar algoritmanın varsayılan parametre değerleriyle yapılan eğitimlerden elde edilmiştir. GridSearch ile yapılan en iyi hiperparametre aramasında SVM için C ceza katsayısı 100, çekirdek seçimi ise RBF olmuştur. 10 katlı çapraz doğrulama sonucu doğruluk %83.1'dir. Test setinin doğruluğu %82.29'dur.

Tablo 3.21: İkili sınıflandırma için SVM sonuçları.

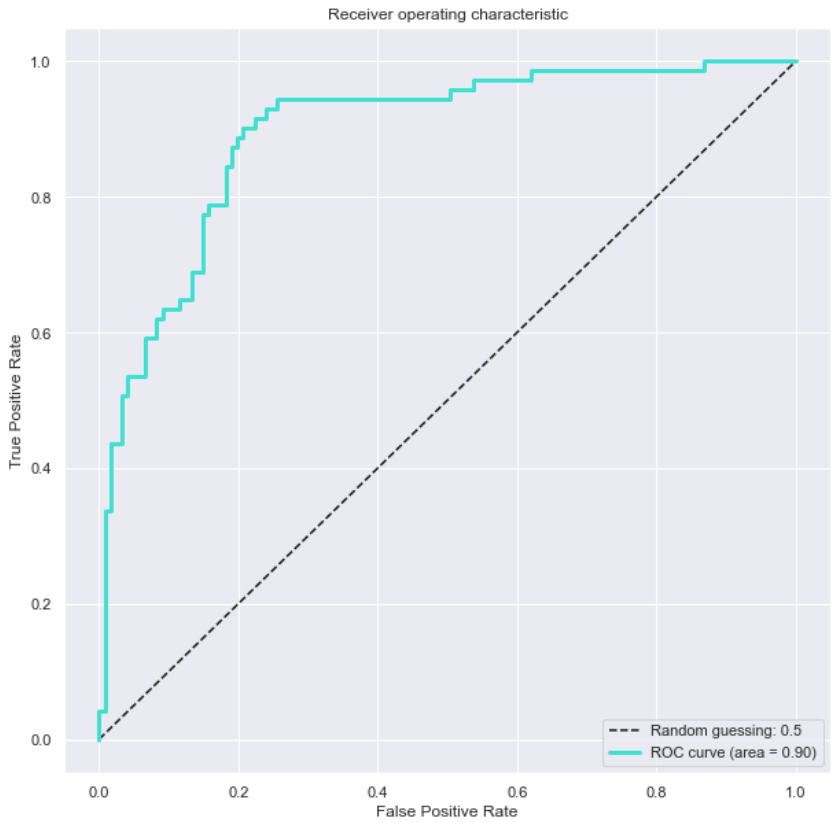
Sınıf	Kesinlik	Duyarlılık	F1-Score
Sınıf 0	0.87	0.84	0.86
Sınıf 1	0.75	0.79	0.77

Tablo 3.22'de eğitim sonuçlarına göre elde edilen karmaşıklık matrisi yer almaktadır. Hasta olmayan 121 kişiden 102'si, hasta olan 71 kişiden 56'sı doğru tahmin edilmiştir.

Tablo 3.22: İkili sınıflandırma için SVM karmaşıklık matrisi.

		Tahmin Edilen Değerler		**Örnek Sayısı**
		0	**1**	
Gerçek Değerler	**0**	102	19	121
	1	15	56	71

Şekil 3.23'te modele ait ROC eğrisi görülmektedir. Bu eğrinin altında kalan alan ise 0.90 olarak bulunmuştur. Eğri altında kalan alan çalışılan veri için modelin başarılı olduğunu göstermektedir.



Şekil 3.23: Destek vektör makineleri için ROC eğrisi.

3.3.3 Hafif Gradyan Artırma Makineleri ile Sınıflandırma

Hiper parametrelerin en optimal değerlerini bulunması için 10 katlı GridSearch araması yapılmıştır. Arama sonucunda öğrenme adımı için 0.01 ve kullanılacak karar ağacı sayısı için 250 değeri bulunmuştur. Bulunan hiper parametrelerle model güncellenmiştir. Uygulanan 10 katlı çapraz doğrulama sonucu doğruluk %89.82 olarak bulunmuştur. Test setine ait doğruluk %90.1'dir. Elde edilen sonuçlar Tablo 3.23'te verilmiştir.

Tablo 3.23: İkili sınıflandırma için LGBM sonuçları.

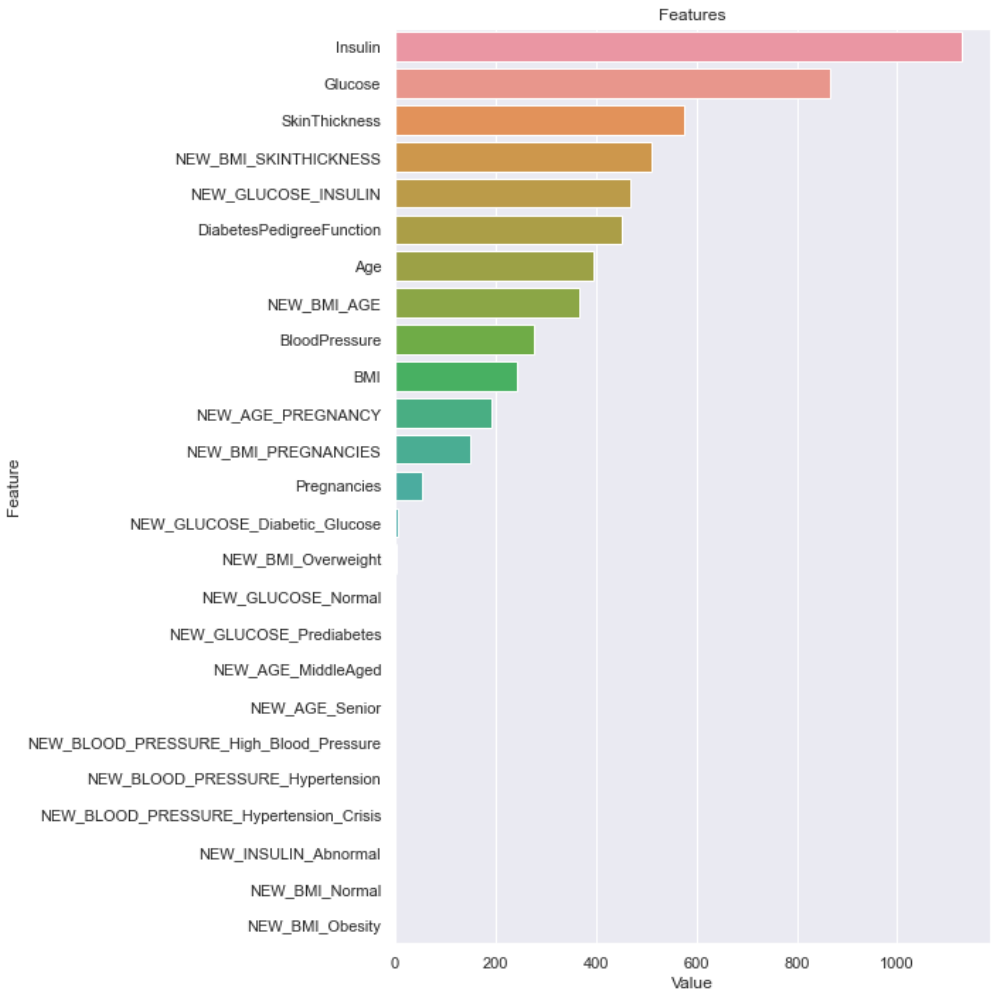
Sınıf	Kesinlik	Duyarlılık	F1-Score
Sınıf 0	0.89	0.95	0.92
Sınıf 1	0.90	0.80	0.85

Tablo 3.24'te test setine göre elde edilen karmaşıklık matrisi yer almaktadır. Hasta olmayan 121 kişiden 1156'i, hasta olan 71 kişiden 57'si doğru tahmin edilmiştir.

Tablo 3.24: İkili sınıflandırma için LGBM karmaşıklık matrisi.

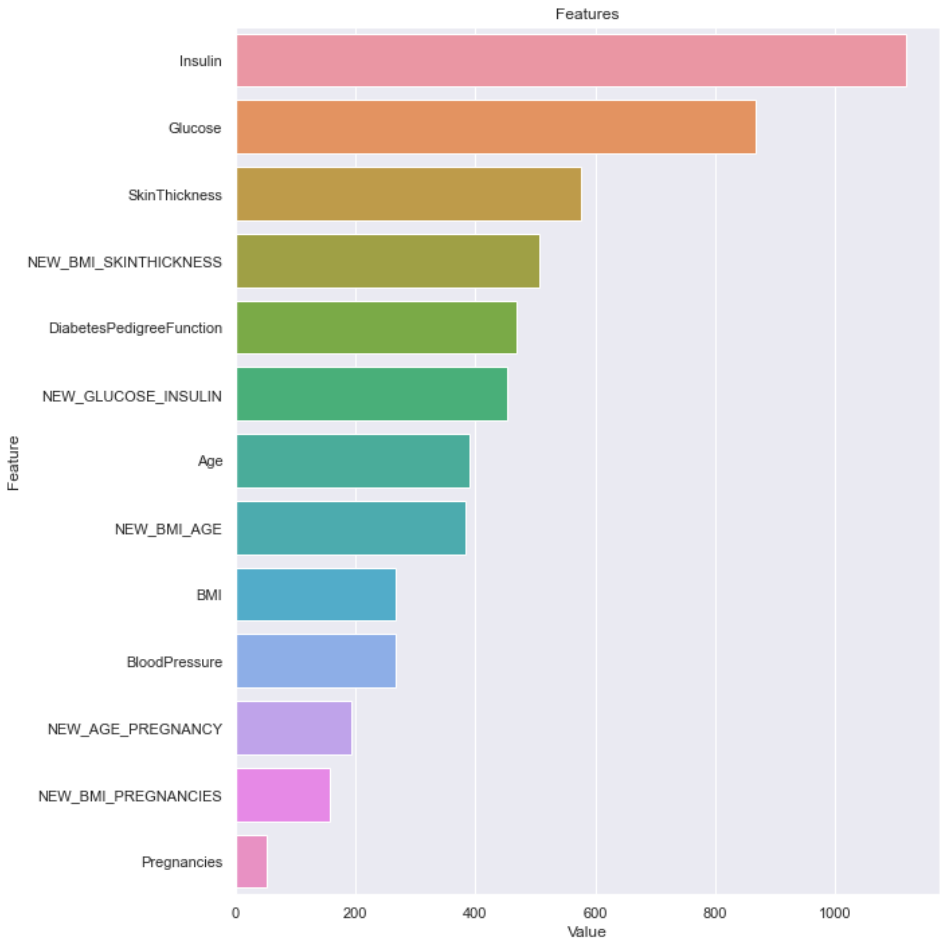
		Tahmin Edilen Değerler		Örnek Sayısı
		0	**1**	
Gerçek Değerler	**0**	115	6	121
	1	14	57	71

Şekil 3.24'te türetilen değişkenlerin modeli nasıl etkilediğini göstermektedir. Daha önce de bahsedildiği gibi kategorileştirilerek türetilen değişkenlerin modele katkısı olmadığı görüldüğü için veri setinden çıkarılmıştır.



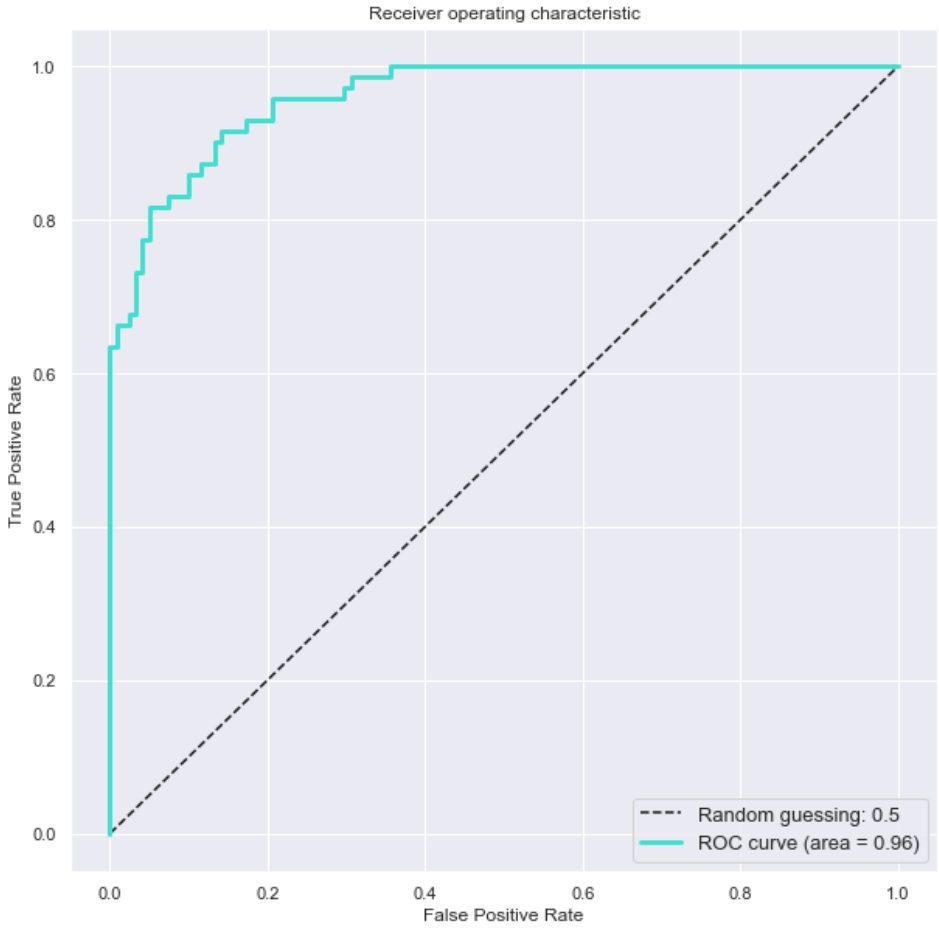
Şekil 3.24: Kategorik değişkenlerle oluşturulan modeldeki değişkenlerin önemi.

Şekil 3.25'te değişkenlerin modeli nasıl etkilediğini göstermektedir. Oluşturulan 5 yeni değişkenin modeli olumlu etkilediği görülmektedir.



Şekil 3.25: İkili sınıflandırma LGBM için değişkenlerin önem sıralaması.

Şekil 3.26'da modele ait ROC eğrisi görülmektedir. Bu eğrinin altında kalan alan ise 0.96 olarak bulunmuştur. Bu alanın değeri 1'e oldukça yakın olduğu için modelin sınıflandırma performansı başarılıdır.



Şekil 3.26: İkili sınıflandırma LGBM için ROC eğrisi.

3.4 Sonuçların Karşılaştırılması

Yapılan deneysel çalışmaların sonucunda kullanılan makine öğrenmesi yöntemlerinin çok sınıflı tipindeki veri için doğruluk oranları Tablo 3.25'te görülmektedir. Elde edilen sonuçlara göre modellerin çok sınıflı veri üzerindeki sınıflandırma başarımları birbirine yakın çıkmıştır. En iyi model ise hafif gradyan artırma makineleri olmuştur.

Tablo 3.25: Çok sınıflı sınıflandırma verisi için model sonuçları.

Model	Doğruluk Oranı
Yapay Sinir Ağı	%96.86
Destek Vektör Makineleri	%95.4
Hafif Gradyan Artırma Makineleri	%99.92

Kullanılan makine öğrenmesi yöntemlerinin çok etiketli tipindeki veri için doğruluk oranları Tablo 3.26'da görülmektedir. Elde edilen sonuçlara göre modellerin çok etiketli veri üzerinde yapay sinir ağı ve destek vektör makineleri neredeyse aynı sonucu vermiştir. En iyi model ise diğer modellere göre daha yüksek doğrulukla en iyi model ise hafif gradyan artırma makineleri olmuştur.

Tablo 3.26: Çok etiketli sınıflandırma verisi için model sonuçları.

Model	Doğruluk Oranı
Yapay Sinir Ağı	%91.21
Destek Vektör Makineleri	%92.13
Hafif Gradyan Artırma Makineleri	%98.21

Kullanılan makine öğrenmesi yöntemlerinin ikili sınıflandırma tipindeki veri için doğruluk oranları Tablo 3.27'de görülmektedir. Elde edilen sonuçlara göre modellerin iki sınıflı veri üzerindeki sınıflandırma başarımları arasında en iyi model ise hafif gradyan artırma makineleri olmuştur.

Tablo 3.27: İkili sınıflandırma verisi için model sonuçları.

Model	Doğruluk Oranı
Yapay Sinir Ağı	%86.45
Destek Vektör Makineleri	%82.29
Hafif Gradyan Artırma Makineleri	%90.1

4. SONUÇLAR

Bu tez çalışması, hekimlerin hastanın problemini daha kolay anlamasını, hızlı tespit edebilmesini ve sonuçları yorumlayabilmesini yardımcı olmak sağlayabilmeleri amacıyla yapılmıştır. Çalışmada iki farklı veri seti ele alınmıştır. Temel veri seti 2021 yılında Pamukkale Üniversitesi Hastanesi İç Hastalıkları Polikliniği'ne başvurmuş hastaların kan test bilgilerini içeren gerçek hayat verisidir. Bu veri üzerinde çok sınıflı ve çok etiketli sınıflandırma çalışması yeniden düzenlenmiştir. Diğer veri seti geçmiş literatür çalışmalarına bakıldığında en fazla çalışılan veri seti olan Pima Indian Diabetes veri setidir. Bu veri seti algoritmaların performanslarının karşılaştırılması amacıyla kullanılmıştır.

Makine öğrenmesi modellerinin başarım değerini etkileyen en önemli aşamalar veriyi hazırlama, temizleme ve işlenebilecek formata getirebilme aşamalarıdır. İyi verinin olmadığı çalışmada iyi bir modelin olması yeterli olmayacaktır. Bu sebeple bu çalışmada veri ön işleme adımlarının çokça üzerinde durulmuştur. Bu tezde çalışılmış olan veri setlerinde bulunan kan testleri veya kan basıncı gibi değişkenlerin normal olmayan değerlerine baskılama yöntemiyle düzeltilmiştir. Çalışmada uzaklık ve gradyan azalan temelli algoritmalar kullanıldığı için veri setlerinde aykırı olabilecek değerler analiz edilmiştir. Veri setlerindeki eksik değerler için hedef değişkenler baz alınarak doldurma işlemi yapılmıştır. Pamukkale Üniversitesi Hastanesi İç Hastalıkları Polikliniği'nden elde edilen veri setine ağaç temelli algoritma olan hafif gradyan artırma makinelerinin varsayılan parametreleri ile temel bir model eğitiminden sonra değişkenlerin modeli ne kadar etkilediği incelenmiştir. Değişkenlerin önem sıralamasına göre gereksiz görülen değişkenler veri setinden çıkarılmış ve korelasyon matrisi yardımıyla yeni değişkenler türetilmiştir. Modelleme öncesinde verileri ölçeklendirme için Robust ölçeklendirme yöntemi kullanılmıştır. Tüm veri setleri %75'i eğitim, %25'i test olacak şekilde ayrılmıştır. Yapay sinir ağları yöntemi uygulanırken Pamukkale Üniversitesi Hastanesi İç Hastalıkları Polikliniği'nden alınan veri setinden seçilen eğitim setinin %30'u, Pima Indian Diabetes veri setinden seçilen eğitim setinin %20'si doğrulama için kullanılması amacıyla ayrılmıştır. Pamukkale Üniversitesi Hastanesi İç Hastalıkları Polikliniği'nden elde edilen verilerin eğitim setine 5 katlı çapraz

doğrulama yapılmıştır, Pima Indian Diabetes veri setinin eğitim setine ise 10 katlı çapraz doğrulama yapılmıştır. Makine öğrenmesi geleneksel modellerinden olan yapay sinir ağları ile destek vektör makineleri, modern modellerinden olan hafif gradyan artırma makineleri yöntemleri olmak üzere üç yöntem uygulanmıştır. En iyi sonuçları verecek hiper parametreleri almak için GridSearch araması yapılmıştır. En iyi sonucu verecek olan hiper parametre değerleri ile modeller güncellenerek test verileri üzerinde test edilmiştir.

Bu çalışmada elde edilen sonuçlara göre hafif gradyan artırma makineleri en iyi sonucu verirken ardından sırasıyla yapay sinir ağları ve destek vektör makineleri gelmektedir. Hem başarı açısından hem süre açısından hafif gradyan makineleri daha performanslı olduğu için bu çalışma için daha tercih edilebilir bir yöntemdir. Çalışmada kullanılan ikinci veri seti olan Pima Indian Diabetes veri setinde ise en iyi sonuç hafif gradyan makineleri modelinde görülmüştür. Yapay sinir ağı ve destek vektör makineleri modellerinin sonuçları, yapılan geçmiş çalışmalarda elde edilen sonuçlara benzer iken hafif gradyan artırma makineleri modelinde ise bu çalışmada küçük bir farkla daha iyi sonuç elde edilmiştir. Geçmiş çalışmalara bakıldığında farklı makine öğrenmesi modelleri kullanarak daha iyi sonuçlar elde edildiği gözlemlenmiştir. Bu sebeple bu veri setinde doğruluk oranını artırmak için başka makine öğrenmesi modelleri kullanılabilir ya da en iyi sonuç veren iki model alınarak topluluk öğrenme modeli kurularak sınıflandırma performansı artırılabilir. Çalışmadan elde edilen sonuçlar ile modellerin hastalık tespitinde doktorlara yardımcı olabileceği görülmüştür. Bu modeller hastane sistemine entegre edilerek hastalara uygulanan kan testlerinin kısa sürede analizini yapıp hastalarda mevcut olan iç hastalıklarını tespit edebilecektir.

5. KAYNAKLAR

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

Albon., C., *Python Machine Learning Cookbook*, O'Reilly Media, (2018).

AlKaabi, L. A., Ahmed, L.S., Al Attiyah, M.F.A., Abdel-Rahman,M. E., "Predicting hypertension using machine learning: Findings from Qatar Biobank Study", *PLoS ONE*, 15(10), (2020).

Bag., S., "Activation functions-all you need to know! [Online]", (2022), https://medium.com/analytics-vidhya/activation-functions-all-you-need-to-know-355a850d025e, (2021).

Bagheri, R., "An introduction to deep feedforward neural networks [Online]", (2022), https://towardsdatascience.com/an-introduction-to-deep-feedforward-neural-networks-1af281e306cd, (2020).

Baheti, P., "Activation functions in neural networks [12 Types & Use Cases] [Online]", (2022), https://www.v7labs.com/blog/neural-networks-activation-functions, (2022).

Bromuri, S., Zufferey, D., Hennebert, J., Schumacher, M., "Multi-label classification of chronically ill patients with bag of words and supervised dimensionality reduction algorithms", *Journal of Biomedical Informatics*, 51, 165–175, https://doi.org/10.1016/J.JBI.2014.05.010, (2014).

Brownlee., J., "How to scale data with outliers for machine learning [Online]", (2022), https://machinelearningmastery.com/robust-scaler-transforms-for-machine-learning/, (2020).

Chen, P., Pan, C., "Diabetes Classification Model Based on Boosting Algorithms", *BMC Bioinformatics*, 19 (1), 1-9, (2018).

Cortes, C., Vapnik, V. and Saitta, L., "Support-Vector Networks", *Machine Leaming*, 20, 273–297, (1995).

Garcia, S. I., "*An introduction to gradient descent algorithm [Online]*", (2022), https://montjoile.medium.com/an-introduction-to-gradient-descent-algorithm-34cf3cee752b, (2018).

GeekforGeeks, “Intuition of adam optimizer [Online]”, (2022), https://www.geeksforgeeks.org/intuition-of-adam-optimizer/, (2020).

Gèron, A., *Hands-on machine learning with scikit-learn, keras & tensorflow*, O’Reilly Media, (2019).

Herrera, F., Charte, F., Rivera, A. J., & del Jesus, M. J. “*Multilabel classification: problem analysis, metrics and techniques”*, Springer, 1–194, (2016).

Ippolito, P. P., “SVM:Feature Selection and Kernels [Online]”, (2022), https://towardsdatascience.com/svm-feature-selection-and-kernels-840781cc1a6c, (2019).

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., *Lightgbm: A highly efficient gradient boosting decision tree*, Advances in Neural Information Processing Systems, 30, (2017).

Khosla, S., “ML-using SVM to perform classification on a non-linear dataset [Online]”, (2022), https://www.geeksforgeeks.org/ml-using-svm-to-perform-classification-on-a-non-linear-dataset/, (2019).

Lai, D. T. H., Begg, R., Palaniswami, M. “SVM models for diagnosing balance problems using statistical features of the MTC signal”, *International Journal of Computational Intelligence and Applications*, *7*(3), 317–331, (2008).

Lai, H., Huang, H., Keshavjee, K., Guergachi, A., Gao, X., “Predictive models for diabetes mellitus using machine learning techniques”, *BMC Endocr Disord*, 19(101), (2019).

LightGBM, “Features [Online]”, (2022), https://lightgbm.readthedocs.io/en/latest/Features.html, (2021).

Li, R., Liu, W., Lin, Y., Zhao, H., Zhang, C., Li, R., Liu, W., Lin, Y., Zhao, H. and Zhang, C., “An Ensemble Multilabel Classification for Disease Risk Prediction”, *Journal of Healthcare Engineering*, https://doi.org/10.1155/2017/8051673, (2017).

Li, Y., Ma, H., Shen, Z., Xu, C. and Yu, C., “Application of machine learning techniques for clinical predictive modeling: A cross-sectional study on nonalcoholic fatty liver disease in China”, *BioMed Research International*, 2018, 1-9, (2018).

Mercer, J., “XVI. Functions of positive and negative type, and their connection the theory of integral equations”, 209, 415-446, DOI: 10.1098/rsta.1909.0016., (1909).

Karakoyun, M. ve Hacıbeyoğlu, M., "Biyomedikal veri kümeleri ile makine öğrenmesi sınıflandırma algoritmalarının istatistiksel olarak karşılaştırılması", *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, 16 (48), 30-41, (2014).

Müller., A.C., Guido, S., *Introduction to Machine Learning with Python*, O'Reilly Media, (2016).

Nahzat, S., Yağanoğlu, M. "Diabetes Prediction Using Machine Learning Classification Algorithms", *European Journal of Science and Technology (EJOSAT)*, (24), 53-59, (2021).

Nuric, "Imperial college machine learning - neural networks [Online]", (2022), https://www.doc.ic.ac.uk/~nuric/teaching/imperial-college-machine-learning-neural-networks.html, (2022).

Panchal, S., "Artificial neural networks – mapping the human brain[online]", (2022), https://medium.com/predict/artificial-neural-networks-mapping-the-human-brain-2e0bd4a93160, (2018).

Pandey, S., Gour, D. K. and Sharma, V., "Comparative study on classification of thyroid diseases", *Uluslararası Mühendislik Eğilimleri ve Teknoloji Dergisi (IJETT)*, 28(9), (457-460), (2015).

Rasmussen, J. T., "Understanding the hyperplane of scikit-learn's SVC Model [Online]", (2022), https://towardsdatascience.com/understanding-the-hyperplane-of-scikit-learns-svc-model-f8515a109222 , (2022).

Scikit-learn, "Cross-validation: evaluating estimator performance [Online]", (2022), https://scikit-learn.org/stable/modules/cross_validation.html, (2011).

Scikit-learn, "Metrics and scoring:quantifying the quality of predictions [Online]",(2022), https://scikit-learn.org/stable/modules/model_evaluation.html#accuracy-score, (2011).

Scikit-learn, "sklearn.multiclass.OneVsRestClassifier [Online]", (2022), https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html#:~:text=OneVsRestClassifier%20can%20also%20be%20used,label%20j%20in%20sample%20i, (2011).

Sharma, S., “Activation functions in neural networks [Online]”, (2022), https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6, (2017).

Singh, R., “Mathematics behind support vector machine [Online]”, (2022), https://medium.com/analytics-vidhya/mathematics-behind-support-vector-machine-4e8130b83840, (2019).

TURHAN, S., ÖZKAN, Y., YÜREKLİ, B. S., SUNER, A., DOĞU, E., “Comparison of Ensemble Learning Methods for Disease Diagnosis in Presence of Class Unbalanced: Case of Diabetes”, *Turkiye Klinikleri Journal of Biostatistics*, 12(1), 16–26, https://doi.org/10.5336/BIOSTATIC.2019-66816, (2020).

Uğuz, S., *Makine Öğrenmesi Teorik Yönleri ve Python Uygulamaları ile Bir Yapay Zeka Ekolü*, Ankara, (2019).

Venkatesan, R., Er, M. J., “Multi-label classification method based on extreme learning machines”, 2014 13th International Conference on Control Automation Robotics and Vision, ICARCV 2014, https://doi.org/10.1109/ICARCV.2014.7064375, (2014).

Yamini, “Logistic regression-first classification model in ML [Online]”, (2022), https://medium.com/geekculture/logistic-regression-first-classification-model-in-ml-5642b28298d5, (2021).

Yıldız, A., “Makine öğrenmesi yöntemleri ile tiroit hastalığının teşhisi”, Yüksek Lisans Tezi, *Sakarya Uygulamalı Bilimler Üniversitesi Lisansüstü Eğitim Enstitüsü*, Elektrik-Elektronik Mühendisliği Ana Bilim Dalı, Sakarya, (2019).

Zhou, L., Zheng, X., Yang, D., Wang, Y., Bai, X., Ye, X., “Application of multi-label classification models for the diagnosis of diabetic complications”, *BMC Medical Informatics and Decision Making*, 21(1), 1–10, https://doi.org/10.1186/S12911-021-01525-7/TABLES/4, (2021).

Zhu, Z., “Explain support vector machines in mathematic details [Online]”, (2022), https://towardsdatascience.com/explain-support-vector-machines-in-mathematic-details-c7cc1be9f3b9, (2020).

Zufferey, D., Hofer, T., Hennebert, J., Schumacher, M., Ingold, R., Bromuri, S., “Performance comparison of multi-label learning algorithms on clinical data for chronic diseases”, *Computers in Biology and Medicine*, 65, 34–43. https://doi.org/10.1016/J.COMPBIOMED.2015.07.017, (2015).