



T.C.

İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA



YÜKSEK LİSANS TEZİ

**ENTÜBASYON İŞLEMİNİN YAPAY ZEKA ALGORİTMALARI İLE
MODELENMESİ**

Muhammed Emin ŞENOL

DANIŞMAN
Doç. Dr. Aysel ERSOY

II. DANIŞMAN
Dr. Öğr. Üyesi Hülya YILMAZ AK

Biyomedikal Mühendisliği Anabilim Dalı

Biyomedikal Mühendisliği Programı

İSTANBUL-2022

Bu çalışma, 31.01.2022 tarihinde aşağıdaki jüri tarafından Biyomedikal Mühendisliği Anabilim Dalı, Biyomedikal Mühendisliği Programında Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Doç. Dr. Aysel ERSOY (Danışman)

İstanbul Üniversitesi-Cerrahpaşa

Mühendislik Fakültesi

Dr. Öğr. Üyesi İnci ZAIM GÖKBAYI

Doç. Dr. Abdürrahim AKGÜNDOĞDU

İstanbul Üniversitesi

İstanbul Üniversitesi-Cerrahpaşa

Fen Bilimleri Enstitüsü

Mühendislik Fakültesi

(||)

(||)

| |

| |

| |

| |

ÖNSÖZ

Bu çalışmamın her aşamasında ve her konuda bana destek olan, deneyimlerini paylaşan ve yardımlarını esirgemeyen hocalarım Doç. Dr. Aysel ERSOY ve Dr. Öğr. Üyesi Hülya YILMAZ AK'a sonsuz şükranlarımı sunarım. Ayrıca tez yazımı konusunda ve fikirlerimin olgunlaşmasında bana yardımcı olan tüm hocalarıma, abim Arş. Gör. Halil İbrahim ŞENOL'a ve arkadaşlarıma teşekkürü bir borç bilirim. Araştırma ve veri toplama sürem boyunca bana destek olan İstanbul Üniversitesi-Cerrahpaşa Kardiyoloji Enstitüsü'nün yapmış oldukları yardımlar için teşekkür ederim. Bununla birlikte tüm tez yazım süresi boyunca yanımda olan ve desteklerini esirgemeyen karım Tuğba TAŞTAN ŞENOL'a ve aileme çok teşekkür ederim.

Ocak 2022

Muhammed Emin ŞENOL



İÇİNDEKİLER

Sayfa No

ÖNSÖZ	iv
İÇİNDEKİLER.....	v
ŞEKİL LİSTESİ	vii
TABLO LİSTESİ.....	x
SİMGE VE KISALTMA LİSTESİ.....	xii
ÖZET	xvi
SUMMARY	xviii
1. GİRİŞ.....	1
2. GENEL KISIMLAR.....	3
2.1. HASTALIĞIN TANIMI	3
2.2. HASTALIĞIN TEŞHİSİNDE KULLANILAN PARAMETRELER.....	3
2.3 YAPAY ZEKA ALGORİTMALARININ KULLANIMI VE YAPAY ZEKANIN SAĞLIK ALANINDAKİ ÖNEMİ	9
2.4. ZOR ENTÜBASYONDA YAPAY ZEKA ALGORİTMALARININ KULLANIMI	13
3. MALZEME VE YÖNTEM.....	15
3.1. HAM VERİLER	15
3.2. HAM VERİLERİN KULLANILDIĞI KÜTÜPHANELER VE YAPAY ZEKA ALGORİTMALARI	16
3.2.1.Yapay Sinir Ağları.....	17
3.2.2.Yapay Sinir Ağları Aktivasyon Fonksiyonları	17
3.2.3.Karar Ağaçları Ve Rastgele Orman Algoritması.....	19
3.2.4.XGBOOST Algoritması	20
3.2.5.K-En Yakın Komşu Algoritması	22
3.2.6.Destek Vektör Makineleri Algoritması	23
3.3. VERİLERİ İŞLEMEDE KULLANILAN YARDIMCI KÜTÜPHANELER VE FONKSİYONLARI	24
3.4. EĞİTİM, VALİDASYON VE TEST VERİ SETİ ANALİZİ VE VERİ SETİNİ AYRIŞTIRMA YÖNTEMLERİ	24
3.5. KERAS VE SCİKİT-LEARN KÜTÜPHANESİNDE KULLANILAN PARAMETRELER	25

3.5.1.Keras Yapay Sinir Ağlarında Kullanılan Hiperparametreler	26
3.5.2.SCİKİT-LEARN Çok Katmanlı Algılayıcılarda Kullanılan Hiperparametreler	28
3.5.3.Rastgele Ormanda Kullanılan Hiperparametreler	28
3.5.4.Karar Ağaçlarında Kullanılan Hiperparametreler	29
3.5.5.XGBOOST'a Kullanılan Hiperparametreler	30
3.5.6.K-En Yakın Komşuda Kullanılan Hiperparametreler	30
3.5.7.Destek Vektör Makinesinde Kullanılan Hiperparametreler	31
3.6. ALGORİTMALARDA KULLANILAN VALİDASYON YÖTEMLERİ.....	32
3.6.1.Hold-Out Validasyon Yöntemi.....	32
3.6.2.K-Kat Çapraz Validasyon Yöntemi.....	32
3.6.3.Birini Dışarıda Bırakma Yöntemi	33
4. BULGULAR.....	35
4.1. KERAS YAPAY SİNİR AĞLARI ÖĞRENME SONUÇLARI.....	35
4.1.1.Zor Entübe Veri Setinin Öğrenme Sonuçları	36
4.1.2.Entübe Veri Setinin Sonuçları	38
4.2. VERİ SETİNİN EĞİTİM, VALİDASYON VE TEST SONUÇLARI.....	40
4.2.1.Zor Entübe Veri Seti Eğitim Validasyon Ve Test Sonuçları.....	40
4.2.2.Entübe Veri Setinin Eğitim, Validasyon Ve Test Sonuçları	50
4.2.3.Birini Dışarıda Bırakma Çapraz Validasyon Yöntemi İle ÇKAS Modeli	60
4.3. KARAR AĞAÇLARI TABANLI ALGORİTMALARDA KULLANILAN PARAMETRELERİN ÖNEM DÜZEYLERİ.....	62
4.4. YAPAY ZEKA ALGORİTMALARIN AİK-EAA GRAFİKLERİ	64
4.4.1. Zor Entübe Veri Setinin ROC Eğrisi.....	65
4.4.2. Entübe Veri Setinin ROC Eğrisi.....	73
5. TARTIŞMA VE SONUÇ	78
KAYNAKLAR.....	80
EKLER	85
EK-1	85
ÖZGEÇMİŞ	86

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1. (a) Samsoun ve Young tarafından değiştirilen Mallampati sınıflandırması [11]. Sınıf I- Tonsillar pillar kolaylıkla görselleştirilir. Sınıf II- Tüm uvula görselleştirildi. Sınıf III- Yalnızca uvula'nın tabanı görselleştirilmiştir. Sınıf IV- sadece kemikli parça görünür hale gelir. (b) Cormack ve Lehane'ye göre laringoskopik derece. Derece I- Tüm glottic orifis görülüyor. Derece II- Sadece giyottik deliğin arka yüzü görülür. Derece III- Giottik açıklık görülmez. Derece IV- Epiglottis görülmez [12].....	5
Şekil 2.2. Üst solunum yolunun anatomisi [12].	6
Şekil 2.3. Tiromental mesafe.....	7
Şekil 2.4. Sternomental mesafenin ölçümü.	8
Şekil 2.5. Otonom algılayıcı model (Autonomous Perceptron Model,OAM) modelinin mimarisi. Sagheer 2019, makalesinden yeniden üretilmiştir [24].	10
Şekil 2.6. Bir makine öğrenimi modelinin uygulama metodolojisi. V, veritabanı; MÖ, makine öğrenimi [26].	11
Şekil 2.7. Çok katmanlı algılayıcı. Yeşil renk giriş katmanı, pembe renk gizli katmanı bu gizli katmanlar artırılabilir ve mavi renk çıktı katmanını temsil eder. Her çizgi ağ için ağırlığı temsil eder ve bu ağırlıklar eğitim sürecinde belirlenir.....	12
Şekil 3.1. Keras yapay sinir ağlarında oluşturulan modelin özeti.	27
Şekil 3.2. K-kat çapraz validasyon yönteminin şematik gösterimi [49].....	33
Şekil 4.1. Keras yapay sinir ağları ile oluşturulan zor entübe veri seti modelinin eğitim ve validasyon doğruluk karşılaştırma grafiği.	36
Şekil 4.2. Keras yapay sinir ağları ile oluşturulan zor entübe veri seti modelin eğitim ve validasyon kayıp grafiği.	37
Şekil 4.3. Keras yapay sinir ağları ile oluşturulan entübe veri seti modelinin eğitim ve validasyon doğruluk karşılaştırma grafiği.	38
Şekil 4.4. Keras yapay sinir ağları ile oluşturulan entübe veri seti modelin eğitim ve validasyon kayıp grafiği.....	39
Şekil 4.5. Keras yapay sinir ağları ile oluşturulan modelinin zor entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.	41

Şekil 4.6. Scikit-Learn ile oluşturulan ÇKAS modelinin zor entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.	42
Şekil 4.7. Scikit-Learn ile oluşturulan RA modelinin zor entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.	43
Şekil 4.8. Scikit-Learn ile oluşturulan KA modelinin zor entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.	45
Şekil 4.9. Scikit-Learn ile oluşturulan XGBoost modelinin zor entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.	46
Şekil 4.10. Scikit-Learn ile oluşturulan KEK modelinin zor entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.	48
Şekil 4.11. Scikit-Learn ile oluşturulan DVM modelinin zor entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.	49
Şekil 4.12. Scikit-Learn zor entübe verisine göre modellerin doğruluk oran grafiği.	50
Şekil 4.13. Keras yapay sinir ağları ile oluşturulan modelinin entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.	51
Şekil 4.14. Scikit-Learn ile oluşturulan ÇKAS modelinin entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.	52
Şekil 4.15. Scikit-Learn ile oluşturulan RA modelinin entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.	54
Şekil 4.16. Scikit-Learn ile oluşturulan KA modelinin entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.	55
Şekil 4.17. Scikit-Learn ile oluşturulan XGBoost modelinin entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.	56
Şekil 4.18. Scikit-Learn ile oluşturulan KEK modelinin entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.	58
Şekil 4.19. Scikit-Learn ile oluşturulan DVM modelinin entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.	59
Şekil 4.20. Scikit-Learn entübe verisine göre modellerin doğruluk oran grafiği.	60
Şekil 4.21. Scikit-Learn ile oluşturulan ÇKAS modelinin zor entübe veri setinin birini dışarıda bırakma çapraz validasyon yöntemine göre test sonuçlarının karışıklık matrisi sonuçları.	61
Şekil 4.22. Oluşturulan Scikit-Learn rastgele orman modelinde verideki değişkenleri kullanmadaki önem düzeyi.	62
Şekil 4.23. Oluşturulan Scikit-Learn karar ağaçları modelinde verideki değişkenleri kullanmadaki önem düzeyi.	63

Şekil 4.24. Oluşturulan Scikit-Learn XGBoost modelinde verideki değişkenleri kullanmadaki önem düzeyi.....	64
Şekil 4.25. Keras yapay sinir ağları sınıflandırma modelinin zor entübe veri setinin sonuçlarının AİK eğrisi.	65
Şekil 4.26. Scikit-Learn çok katmanlı algılayıcılar sınıflandırma modelinin zor entübe veri setinin sonuçlarının AİK eğrisi.....	66
Şekil 4.27. Scikit-Learn rastgele orman sınıflandırma modelinin zor entübe veri setinin sonuçlarının AİK eğrisi.	67
Şekil 4.28. Scikit-Learn karar ağaçları sınıflandırma modelinin zor entübe veri setinin sonuçlarının AİK eğrisi.	68
Şekil 4.29. Scikit-Learn XGBoost sınıflandırma modelinin zor entübe veri setinin sonuçlarının AİK eğrisi.	69
Şekil 4.30. Scikit-Learn k-en yakın komşu sınıflandırma modelinin zor entübe veri setinin sonuçlarının AİK eğrisi.	70
Şekil 4.31. Scikit-Learn destek vektör makinesi sınıflandırma modelinin zor entübe veri setinin sonuçlarının AİK eğrisi.	71
Şekil 4.32. Scikit-Learn çok katmanlı sınıflandırma modelinin BDB çapraz validasyon yöntemi ile zor entübe veri setinin sonuçlarının AİK eğrisi.....	72
Şekil 4.33. Keras yapay sinir ağları sınıflandırma modelinin entübe veri setinin sonuçlarının AİK eğrisi.	73
Şekil 4.34. Scikit-Learn çok katmanlı algılayıcılar sınıflandırma modelinin entübe veri setinin sonuçlarının AİK eğrisi.	74
Şekil 4.35. Scikit-Learn rastgele orman sınıflandırma modelinin entübe veri setinin sonuçlarının AİK eğrisi.	74
Şekil 4.36. Scikit-Learn karar ağaçları sınıflandırma modelinin entübe veri setinin sonuçlarının AİK eğrisi.	75
Şekil 4.37. Scikit-Learn XGBoost sınıflandırma modelinin entübe veri setinin sonuçlarının AİK eğrisi.	75
Şekil 4.38. Scikit-Learn k-en yakın komşu sınıflandırma modelinin entübe veri setinin sonuçlarının AİK eğrisi.	76
Şekil 4.39. Scikit-Learn destek vektör makinesi sınıflandırma modelinin entübe veri setinin sonuçlarının AİK eğrisi.	76

TABLO LİSTESİ

	Sayfa No
Tablo 2.1 Karşılaşılan problemlere göre zor entübasyon insidansı [12].	4
Tablo 2.2. Havayolu Zorluk Skoru (Airline Difficulty Score ,HZS). Eğer Skor ≤ 8 olursa ventilasyon ve entübasyon muhtemelen zor olur [12].	8
Tablo 3.1. İstanbul Üniversitesi-Cerrahpaşa Kardiyoloji Enstitüsü'nden alınan rastgele 5 hastanın parametrik tablosu (Ek-2).	16
Tablo 4.1. Keras YSA modelinin zor entübe veri setine göre eğitim ve validasyon işlem sonuçları.	40
Tablo 4.2. Scikit-Learn ÇKAS modelinin zor entübe veri setine göre eğitim ve validasyon işlem sonuçları.	41
Tablo 4.3. Scikit-Learn RA modelinin zor entübe veri setine göre eğitim ve validasyon işlem sonuçları.	43
Tablo 4.4. Scikit-Learn KA modelinin zor entübe veri setine göre eğitim ve validasyon işlem sonuçları.	44
Tablo 4.5. Scikit-Learn XGBoost modelinin zor entübe veri setine göre eğitim ve validasyon işlem sonuçları.	46
Tablo 4.6. Scikit-Learn KEK modelinin zor entübe veri setine göre eğitim ve validasyon işlem sonuçları.	47
Tablo 4.7. Scikit-Learn DVM modelinin zor entübe veri setine göre eğitim ve validasyon işlem sonuçları.	48
Tablo 4.8. Keras YSA modelinin entübe veri setine göre eğitim ve validasyon işlem sonuçları.	50
Tablo 4.9. Scikit-Learn ÇKAS modelinin entübe veri setine göre eğitim ve validasyon işlem sonuçları.	52
Tablo 4.10. Scikit-Learn RA modelinin entübe veri setine göre eğitim ve validasyon işlem sonuçları.	53
Tablo 4.11. Scikit-Learn KA modelinin entübe veri setine göre eğitim ve validasyon işlem sonuçları.	54
Tablo 4.12. Scikit-Learn XGBoost modelinin entübe veri setine göre eğitim ve validasyon işlem sonuçları.	56

Tablo 4.13. Scikit-Learn KEK modelinin entübe veri setine göre eğitim ve validasyon işlem sonuçları.	57
Tablo 4.14. Scikit-Learn DVM modelinin entübe veri setine göre eğitim ve validasyon işlem sonuçları.	58
Tablo 4.15. Scikit-Learn ÇKAS modelinin zor entübe veri setine göre eğitim ve birini dışarıda bırakma validasyon işlem sonuçları.	60



SİMGE VE KISALTMA LİSTESİ

Simgeler	Açıklama
X_n	: Girişler (Yapay Sinir Ağları)
W_n	: Ağırlık (Yapay Sinir Ağları)
Σ	: Toplama fonksiyonu (Yapay Sinir Ağları)
F_j	: Aktivasyon fonksiyonu (Yapay Sinir Ağları)
b	: Bias (Yapay Sinir Ağları)
Maks	: Sıfır ya da daha büyük değeri çıktı verir (Yapay Sinir Ağları)
Σ	: Toplama fonksiyonu (Yapay Sinir Ağları)
F_j	: Aktivasyon fonksiyonu (Yapay Sinir Ağları)
ẑ	: Softmax işlevine giriş vektörü (Yapay Sinir Ağları)
z_i	: Softmax işlevine giriş vektörünün öğeleridir (Yapay Sinir Ağları)
e^{z_i}	: Giriş vektörünün her elemanına standart üstel fonksiyonu uygulanır (Yapay Sinir Ağları)
Σ_{j=1}^K z_j	: Formülün altındaki terim normalizasyon terimidir (Yapay Sinir Ağları)
K	: Çok sınıflı sınıflandırıcıdaki sınıf sayısı (Yapay Sinir Ağları)
n_{i,j}	: j düğümünün önemi (Kara Ağaçları)
w_j	: j düğümüne ulaşan ağırlık sayısı (Kara Ağaçları)
C_j	: j düğümünün saflık derecesi (Kara Ağaçları)
sol(j)	: j düğümünde sola bölünmüş alt bölüm (Kara Ağaçları)

sağ(j)	: j düğümünde sağa bölünmüş alt bölüm (Kara Ağaçları)
f_i	: i özelliğinin önemi (Kara Ağaçları)
n_j	: j düğümünün önemi (Kara Ağaçları)
ROf_i	: Rastgele Orman modelinde tüm ağaçlardan hesapladığımız özelliğin önemi (Rastgele Orman)
normf_{ij}	: j ağacında i için normalleştirilmiş özellik önemi (Rastgele Orman)
T	: Toplam ağaç sayısı (Rastgele Orman)
y_i	: Gerçek etiket değerleri (Rastgele Orman)
ŷ_i	: Tahminin etiket değerleri (Rastgele Orman)
h_t(x)	: Köklerden birinin oyu (XGBoost)
α_t	: Köklerin oyunun ağırlığı (XGBoost)
ε_t	: Kökün hata payı (XGBoost)
C	: Sınıf sayısı (XGBoost)
W_{new}	: Yeni örnek ağırlıkları (XGBoost)
S_w	: Örnek ağırlıkları (XGBoost)
K_w	: Kökün ağırlığı (XGBoost)
x₁	: A noktasındaki x mesafesi (K-En yakın komşu)
x₂	: B noktasındaki x mesafesi (K-En yakın komşu)
y₁	: A noktasındaki y mesafesi (K-En yakın komşu)

y_2	: B noktasındaki y mesafesi (K-En yakın komşu)
k	: Komşu sayısı (K-En yakın komşu)
xi	: i kadar x noktaları (K-En yakın komşu)
yi	: i kadar y noktaları (K-En yakın komşu)
q	: Kaç Sayıda değişkene bağlı olduğunu gösterir (K-En yakın komşu)
w	: Ağırlık (Destek Vektör Makinesi)
x	: Girdi vektörü (Destek Vektör Makinesi)
b	: Sapma (Destek Vektör Makinesi)

Kısaltmalar

Açıklama

AAD	: Amerikan Anesteziyologlar Derneği
ABD	: Amerika Birleşik Devletleri
NAP4	: 4. Ulusal Denetim Projesi
TMM	: Triomental Mesafe
SMM	: Sternomental Mesafe
HZS	: Havayolu Zorluk Skoru
KBB	: Kulak-Burun-Boğaz
YSA	: Yapay Sinir Ağları
DÖ	: Derin Öğrenme
YZ	: Yapay Zeka
MÖ	: Makine Öğrenimi

ÇKA	: Çok Katmanlı Algılayıcılar
GİB	: Grafik İşlemi Birimleri
TİB	: Tensör İşleme Birimleri
TF	: Tensor Flow
UPA	: Uygulama Programlama Arayüzü
KEK	: K-Enyakın Komşu
DVM	: Destek Vektör Makinesi
ÇKAS	: Çok Katmanlı Algılayıcılar Sınıflandırması
RO	: Rastgele Orman
AUG	: Aşırı Uyum Göstermesi
EUG	: Eksik Uyum Göstermesi
AF	: Aktivasyon Fonksiyonu
KA	: Karar Ağaçları
BDB	: Birini Dışarıda Bırakma
EAA	: Eğrinin Altındaki Alan
AİK	: Alıcı İşletim Karakteristiği

ÖZET**|YÜKSEK LİSANS TEZİ|****|ENTÜBASYON İŞLEMİNİN YAPAY ZEKA ALGORİTMALARI İLE
MODELLENMESİ |****|Muhammed Emin ŞENOL|****İstanbul Üniversitesi-Cerrahpaşa****Lisansüstü Eğitim Enstitüsü****|Biyomedikal Mühendisliği Anabilim Dalı |****|Danışman : Doç. Dr. Aysel ERSOY****II. Danışman : Dr. Öğr. Üyesi Hülya YILMAZ AK |**

Entübasyon, endotrakeal tüp adı verilen ince bir tüpün ağızdan solunum yoluna doğru ilerletilmesidir. Zor entübasyon ise deneyimli bir anestezi doktorunun endotrakeal tüpün yerleştirme işleminin 10 dakikadan daha uzun sürmesi veya direkt laringoskopi yapılamaması gibi durumlara denir. Zor entübasyon, entübasyona oranla daha nadir gözükür. Zor entübasyon parametrik verilerine bakıldığında entübasyondan ayrıştırılabilir. Buradaki önemli parametreler olarak Cormack - Lehane derecelendirmesi ve Mallampati sınıflandırması gösterilebilir. Bunların sonuçları olağan dışı değerlere sahipse hastaya müdahale zorlaşmaktadır. Bu çalışmada, İstanbul Üniversitesi Cerrahpaşa Kardiyoloji Enstitüsü'ne ait olan retrospektif hasta verileri yapay zeka algoritmaları ile araştırılmıştır. Bu çalışmada 2019-2021 yılları arasında 157 kişiden toplanan veriler 10 girdi ve 1 çıktı olarak yapay zeka yöntemleri ile değerlendirilmiştir. Bu çalışmada, yazılım dili olarak Python, kütüphaneler olarak ise Keras ve ScikitLearn kullanılmıştır. Bu çalışma ile anestezi uzmanı gözetiminde olmadan veriler kullanılarak yapay zeka algoritmaları, derin öğrenme ve makine öğrenimi ile de başarılı sonuçlar elde edilebileceği gösterilmiştir. |

Ocak 2022, |105| sayfa.

Anahtar kelimeler: | Zor entübasyon ,derin öğrenme , makine öğrenmesi, yapay zeka, yapay sinir ağları |



SUMMARY**M.Sc. THESIS****MODELING THE INTUBATION PROCESS WITH ARTIFICIAL
INTELLIGENCE ALGORITHMS****Muhammed Emin ŞENOL****Istanbul University-Cerrahpasa****Institute of Graduate Studies****Department of Biomedical Engineering****Supervisor : Assoc. Prof. Dr. Aysel ERSOY****Co-Supervisor : Assist. Prof. Dr. Hülya YILMAZ AK**

Intubation is the pushing forward of a thin tube called an endotracheal tube from the mouth to the respiratory tract. Difficult intubation is the situation in which an experienced anesthesiologist takes longer than 10 minutes to insert the endotracheal tube or direct laryngoscopy cannot be performed. Difficult intubation appears to be rarer than intubation. Difficult intubation can be differentiated from intubation by looking at parametric data. Cormack-Lehane rating and Mallampati classification can be shown as important parameters here. The more unusual values these have reached, the more difficult it will be for the patient to intervene. In this study, we researched retrospective patient data belonging to İstanbul University Cerrahpaşa Cardiology Institute with artificial intelligence algorithms. In this study, the data collected from 157 people between 2019-2021 were evaluated with artificial intelligence methods as 10 inputs and 1 output. In this research, Python was used as the software language and Keras and ScikitLearn were used as the libraries. With this study, it has been shown that successful results can be obtained with artificial intelligence algorithms, deep learning and machine learning by using data without the supervision of an anesthesiologist.

January 2022, 105 pages.

Keywords: | Difficult intubation, deep learning, machine learning, artificial intelligence, artificial neural network |



1. GİRİŞ

Entübasyon, endotrakeal tüp adı verilen ince bir tüpün ağızdan solunum yoluna doğru ilerletilmesidir. Entübasyon, hava yolu kontrolünü sağlayamayan ya da yardım almadan soluk alıp verme işlemini gerçekleştiremeyen hastalara uygulanan bir işlemdir. Entübasyon sırasında oluşabilecek bazı komplikasyonlar şunlardır; Hipertansiyon ve taşikardi, Travma, Aritmiler, Aspirasyon vb.

Yapay zekâ algoritmaları genel olarak iki ana başlıkta incelenir. Bunlar, denetimli öğrenme ve denetimsiz öğrenmedir. Denetimsiz öğrenmede çıktılar belli değildir. Model kendi içinde verilere göre kümeler oluşturur. Denetimli öğrenmede farklı veriler ile oluşturulan modelin sonuçları hedef sonuçlar ile karşılaştırılır. Bu karşılaştırma sonucu oluşan hatalı sonuçlar minimize edilmeye çalışılır. Denetimsiz öğrenmenin alt başlığı olarak kümeleme algoritmaları kullanılır. Denetimli öğrenmede ise alt başlık olarak sınıflandırma ve regresyon algoritmaları bulunur.

Yapay zekâ medikal alanda oldukça kullanılır. Bunlar genel olarak; ilaç geliştirme, hastalık teşhisi, sağlık planlarının analizi, cerrahi tedavi, tıbbi verilerin yönetilmesi, tıbbi tedavi gibi. Medikal alanda yapay zekanın avantajlarını ve dezavantajlarını aşağıdaki gibi sıralanabilir.

- Avantajları;
 - Verimlilik, doğruluk ve hassasiyet.
 - Azaltılmış iş yükü, kritik durumlarda sürenin verimli kullanımı.
 - Para tasarrufu ve daha iyi izleme.
- Dezavantajları;
 - İş kaybı.
 - İnsan dokunuşu empati eksikliği, duygusal zekâ.

Veriler derin öğrenme algoritmaları ile kullanıldığında insan hatası olabilecek şeyleri göz ardı eder ve olumlu veya olumsuz olasılıkları bize sunar [23].

Derin öğrenme ve makine öğrenmesi günümüzde oldukça popüler konulardır. Doğru veriler ve anlamlı sonuçlara ulaşıldığında tahmin etmesi güç sonuçları bize kolaylıkla söyleyebilir. Zor hava yolu tahmin sürecinin riskli olduğu alanlardan biridir. Zor hava yolu doğru tespit edilip, doğru karar verilmez ise hastada kalıcı rahatsızlık ya da ölüme sebep olabilir. Bu tezin yapılmasındaki ana amaç kovid dönemi de dahil 2019-2021 yılları arasında yeni ve gerçek hasta verileri ile Keras ve Scikit-Learn kütüphanelerinin yardımıyla 6 farklı yapay zeka yöntemi kullanılarak hastanın zor entübe, entübe veya entübe edilmemesi sonuçlarının derin öğrenme yöntemleri ile hesaplanmasıdır. |



2. GENEL KISIMLAR

2.1. HASTALIĞIN TANIMI

Anestezi uygulamasının ana unsuru hava yolu yönetimidir. Hava yolu yönetiminde yaşanan problemler ciddi sonuçlara neden olabilir. Amerikan Anesteziyologlar Derneği'nin (American Society of Anesthesiologists, AAD) Zor Hava Yolu Yönetimi Uygulama Kılavuzu, zor hava yolunu “geleneksel olarak eğitilmiş bir anestezi uzmanının bir maske aracılığıyla üst hava yolunun ventilasyonunda güçlük, trakeal entübasyonda zorluk yaşadığı klinik durum” olarak tanımlamaktadır [1]. Anestezist, hastasına her koşulda yeterli gaz değişimi sağlanmalıdır. Buda üst hava yolunun açıklığı ile sürdürülebilir. Akciğer ventilasyonunda oksijen birkaç dakika bile kesilse ölüm de dahil olmak üzere kalıcı hasarlara yola açabilir. Amerika Birleşik Devletleri (United States of America, ABD) verileri anestezi sırasında meydana gelen ölümlerin %30'undan tüm bu sorunların olduğu gözlemlenmektedir [2,3]. Literatürde belirtilen zor entübasyon insidansı, %0.05 ile %18 arasında değişiklik göstermektedir [4-6]. Baş ve boyun patalojisi önemli faktörler olduğu için kulak burun boğaz cerrahisi geçiren hastalar zor hava yolu oranı daha yüksektir [7,8]. 4. Ulusal Denetim Projesi (National Audit Project, NAP4), tüm hastaların potansiyel hava yolu güçlüğüne belirlemek için hava yolu değerlendirmesinden geçmesi gerektiğini ve bu değerlendirmenin yapılmamasının kötü sonuçlara neden olabileceğini belirtmiştir [9].

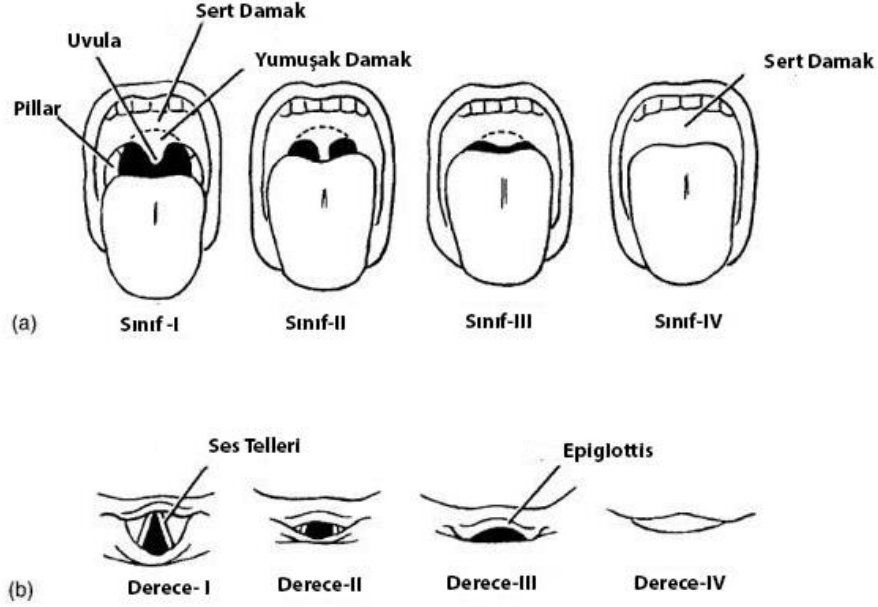
2.2. HASTALIĞIN TEŞHİSİNDE KULLANILAN PARAMETRELER

Zor entübasyon teşhisi için en sık kullanılan kriterler Cormack ve Lehane'nin sınıflandırmasıdır [10]. Entübasyonu, larengoskopide sağlanan glotisin görünümüne göre az veya çok zor olarak tanımlanır (Şekil 2.1). Diğer kriterler ise endotrakeal entübasyon sırasında karşılaşılan çeşitli sorunların insidansı ile ilgili literatürde bulunan verileri göstermektedir (Tablo 2.1) [4]. Dil ağız boşluğuna orantısız büyükse entübasyon zor olabilir. Orofaringeal göre dilin boyunu ölçmek zor olduğundan Mallampati bu faktörü değerlendirmek için kolay bir yöntem önermiştir [15]. Orofarengeal yapılarının görülemeyecek kadar büyük olması durumunda glottisin görülemeyeceği buda entübasyonu zorlaştıran bir faktör olarak karşımıza çıkar [15]. Dört kategoriye

ayırmak için bu sınıflandırma, Samssoon ve Young tarafından değiştirilmiştir (Şekil. 1a.) [16].

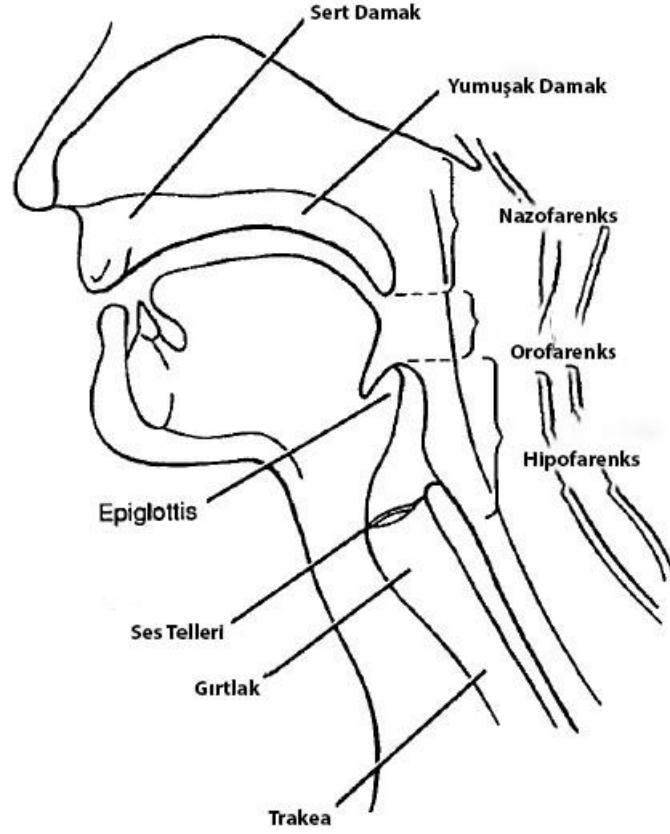
Tablo 2.1 Karşılaşılan problemlere göre zor entübasyon insidansı [12].

Problem	Oran (%)
Ciddi girişim gerektiren entübasyon veya farklı bıçaklar	1-18
Entübasyon olasılığı, Fakat Cormack ve Lehane derece IV	1-4
Entübasyon imkansızdır.	0.05-0.35
Beyin hasarına yol açabilir entübasyon ve ventilasyon imkansızdır.	0.0001-0.02



Şekil 2.1. (a) Samsoun ve Young tarafından değiştirilen Mallampati sınıflandırması [11]. Sınıf I- Tonsillar pillar kolaylıkla görselleştirilir. Sınıf II- Tüm uvula görselleştirildi. Sınıf III- Yalnızca uvula'nın tabanı görselleştirilmiştir. Sınıf IV- sadece kemikli parça görünür hale gelir. (b) Cormack ve Lehane'ye göre laringoskopik derece. Derece I- Tüm glottic orifis görülüyor. Derece II- Sadece giyottik deliğin arka yüzü görülür. Derece III- Giyottik açıklık görülmez. Derece IV- Epiglottis görülmez [12].

Anatomik olarak hava yolunu incelenecek olursa, havanın burunlardan veya ağızdan akciğerlere geçişini sağlayan tüm yapıları içerir (Şekil 2.2). Nazal ve ağız boşlukları sırasıyla nazofarinkse ve orofarinkse açılır. Ön tarafta kemikli damak, sonra yumuşak damak arkadan ayrılırlar. Ağız boşluğunun büyük bir bölümünü kaplayan dil, öncelikle hyoid kemiğe, epiglottis de tabanına tutturulur. Bu yapı glottis ve farenks (trakeaya açılan) ile özofagusu açılan hipofarenks arasında işlevsel ayrımı sağlar. Ses kablolarının bölmesi yaklaşık olarak tiroid kıkırdağıyla (Âdem Elması) aynı seviyededir [12].



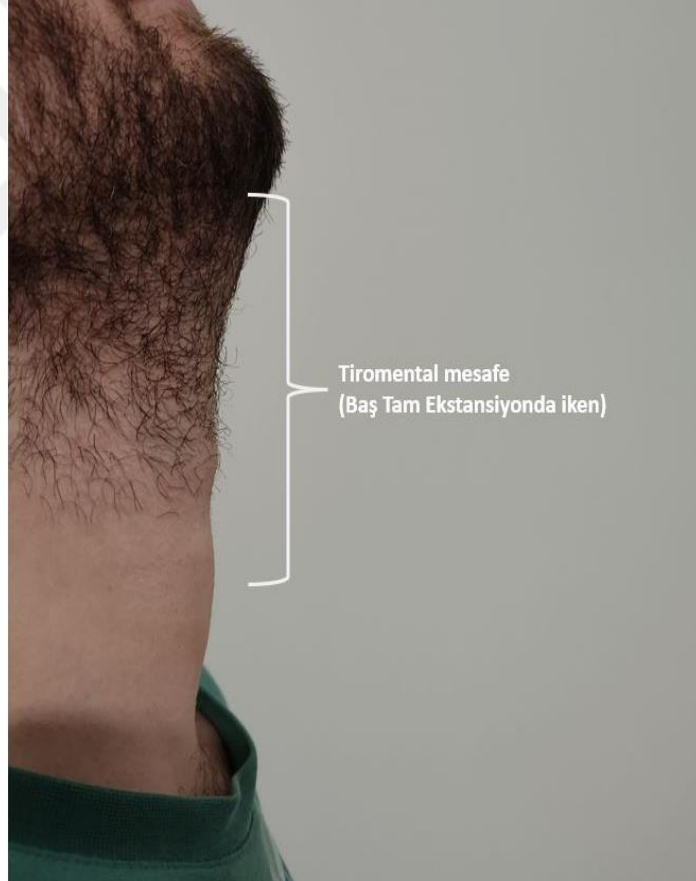
Şekil 2.2. Üst solunum yolunun anatomisi [12].

Zor hava yolu için belirli inceleme alanlarına ve parametrelerine bakılacak olursa, ilk olarak yatak başında obezite durumunu, baş ve boyun patolojisinin dışsal belirtilerini gözlemleyen genel bir muayene yapılır. Tüm bu işaretler, hava yolu yönetiminin daha zor olabileceğini göstermektedir [15].

Hastanın ağzını açtığı mesafe hayati önem taşır. Ağzını açtığı mesafe dediğimiz tabir ise hastanın kesici dişleri arasındaki mesafedir. Kesici dişler arasındaki mesafe 3 cm'den az olması işlem için endişe verici olmaktadır. Bu işlemlerde değerlendirmeler genelde göz ile yapılır. Mallampati sınıflandırması yaygın olarak kullanılır. Anatomik yapılara göre bunlara 1 ile 4 arası bir puan uygulanır (Şekil.1a). 3 ve 4'üncü skor zor ventilasyon ve zor

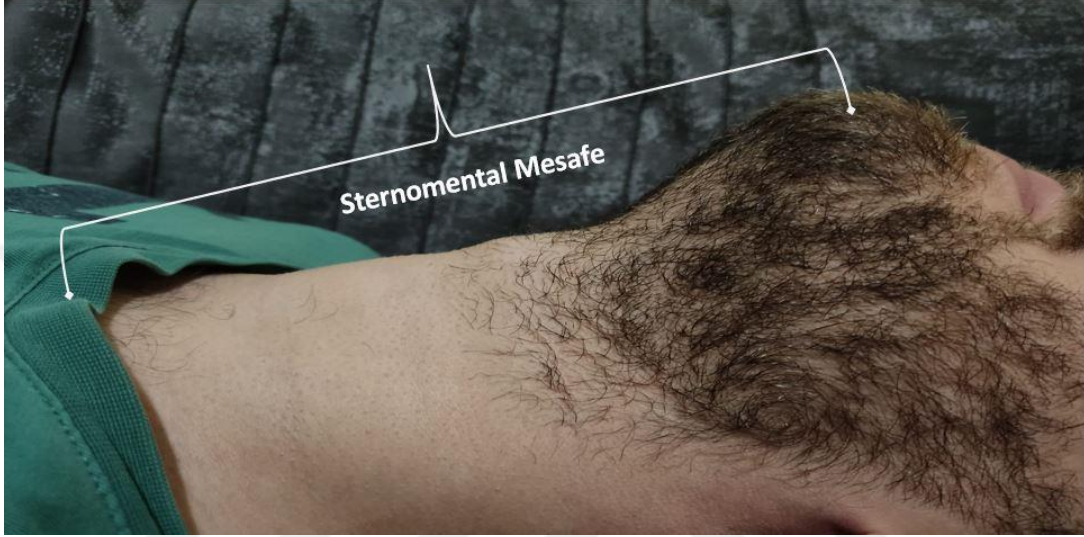
laringoskop ile ilgilidir. Mallampati gözlemcilerinin mallampatiyi doğru derecelendirme oranı %50 ve %75 arasında değişmektedir. Buda hastaların zor entübasyona dahil olmasa bile zor entübasyona dahil olarak göstermektedir [15].

Tiromental mesafe (Tyromental distance ,TMM), baş ve boyun tamamen uzatıldığında ve ağız kapalıyken tiroid çıkıntısı ile çene ucu arasındaki bir ölçümdür (Şekil 2.3). TMM 6 cm' den küçükse, laringoskop bıçağı tarafından dil sıkıştırması için daha az yer var demektir. Bu da laringeal girişin zor olabileceği anlamına gelir. Doğruluk için bu mesafe mezura veya cetvel ile ölçülür. Kesin doğruluk payı mallampatiden daha azdır [15].



Şekil 2.3. Tiromental mesafe.

Sternomental mesafe (Sternomental distance, SMM), Sternomental mesafe, baş tam ekstansiyonda ve ağız kapalı olarak çenenin üst sınırı arasındaki mesafe (cm) olarak ölçülür (Şekil 2.4) [16].



Şekil 2.4. Sternomental mesafenin ölçümü.

Diğer önemli parametreler ise boyun hareketliliği (Atlantookspital eklem ekstansiyonu) ve üst kesici dişler (Üst diş oklüzal-horizontal açısı) olarak söylenebilir (Tablo 2.2) [12]. Atlanto-okspital ve atlanto-aksiyal eklemlerin klinik olarak değerlendirilmesi çok zordur [15].

Tablo 2.2. Havayolu Zorluk Skoru (Airline Difficulty Score ,HZZS). Eğer Skor ≤ 8 olursa ventilasyon ve entübasyon muhtemelen zor olur [12].

Parametre Zorluk Dereceleri	1.	2.	3.
Tiromental Mesafe	>6 cm	5-6 cm	<5cm
Mallampati Sınıfı	Sınıf I	Sınıf II	Sınıf III&IV
Ağız Açıklığı	4 cm	2-3 cm	1 cm
Boyun Hareketliliği	Normal	Kısıtlı	Flexion düzeltildi
Üst Kesici Dişler	Yok	Normal	Çıkıntılı

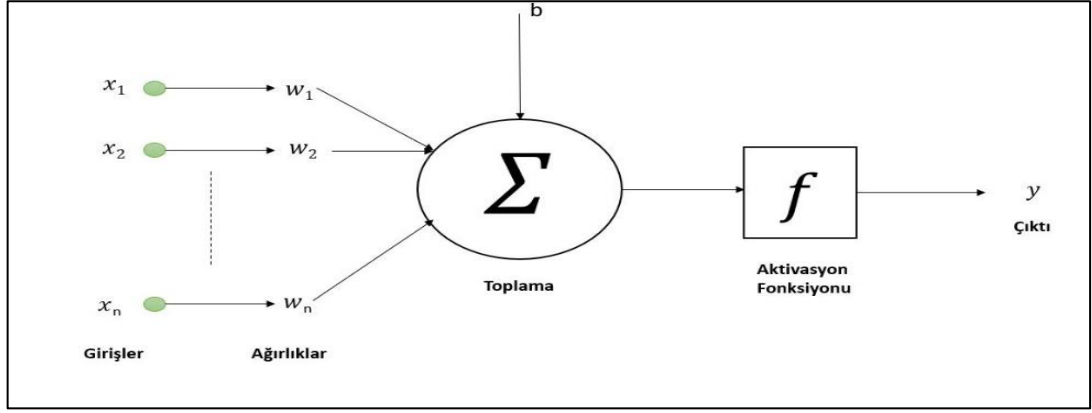
Zor hava yolu yönetimi, kulak-burun-boğaz (KBB) ameliyatı geçiren hastada tahmini bir zor hava yolu gösterdiğinde ayrıntılı bir hava yolu yöntemi geliştirmek için anestezi uzmanı ve KBB cerrahı arasında bir tartışma gereklidir. Bu konu, hava yolu yönetiminde oksijenasyon, ventilasyon ve aspirasyona karşı koruma sağlamak için tasarlanmış önceden belirlenmiş bir dizi ardışık durumu gerektirir [17]. Başarısızlıklar durumunda alternatif stratejiler önceden düşünülmelidir ve başarısızlığın zamanlamasına veya hangi aşamada bir strateji başarısızlığının ortaya çıktığına bağlı olarak farklı planlar yapılmalıdır [18]. AAD zor hava yolu yönetimine ilişkin uygulama yönergeleri potansiyel bir zor hava yolu ile karşı karşıya kalındığında rehberlik etmesini amaçlayan bir zor hava yolu algoritması içerir [19]. Bu algoritma, 3 ayrı senaryoyu ele alır;

1. Tahmin edilen zor hava yolu (Uyanık Entübasyon).
2. Zor entübasyon yeterli oksijenasyon veya ventilasyon (Acil olmayan yolu).
3. Zor entübasyon yeterli oksijenasyon veya ventilasyon olmadan (Acil durum yolu) [20].

Hava yolu değerlendirmesindeki ana zorluk kullanılabilir ideal bir hava yolu değerlendirmesi olmamasıdır. Anestezi uzmanının deneyimi, zor entübasyon olma ihtimali olan hastaların belirlenmesi ve uygun preoperatif planlama yapılması gerekir [15].

2.3 YAPAY ZEKA ALGORİTMALARININ KULLANIMI VE YAPAY ZEKANIN SAĞLIK ALANINDAKİ ÖNEMİ

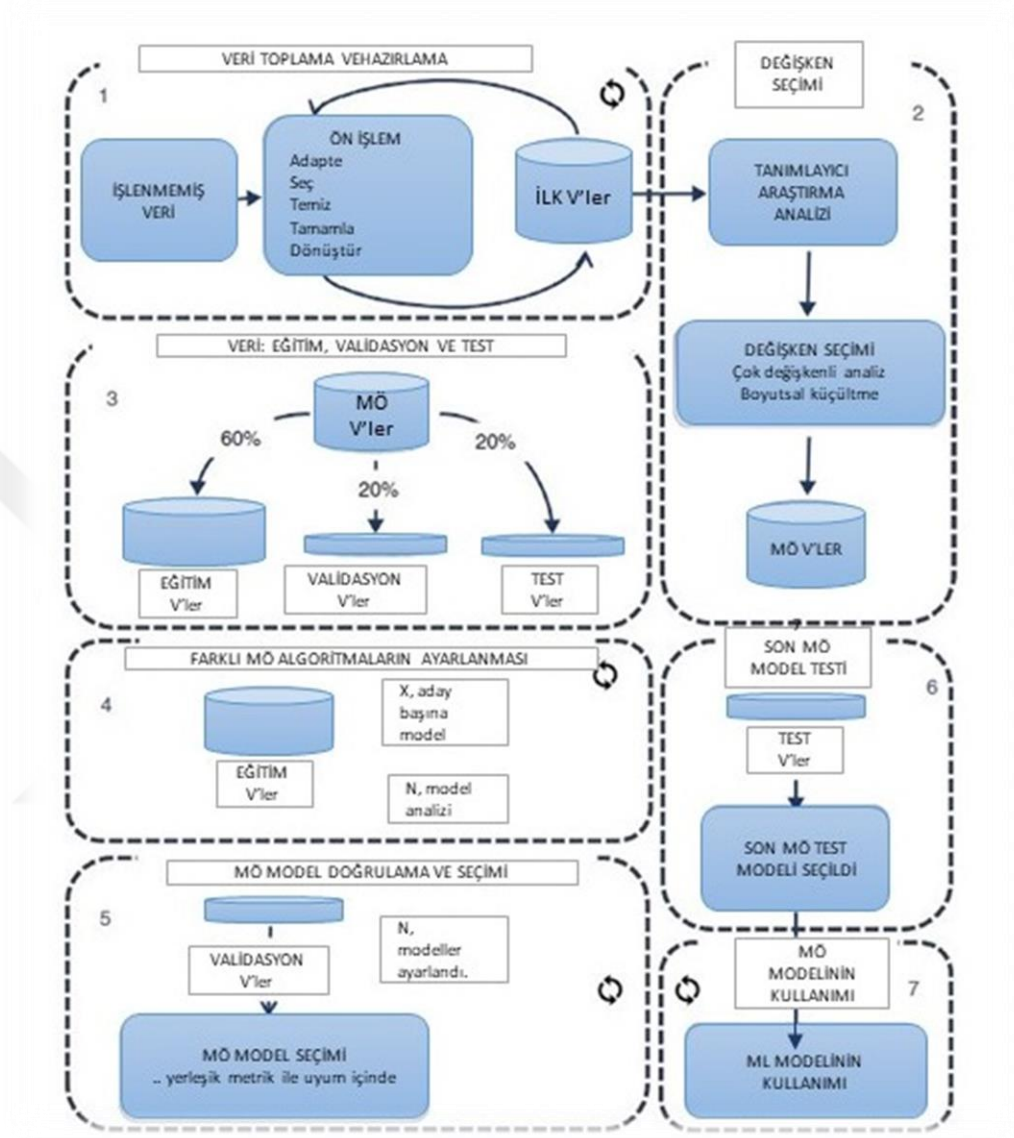
Verilerdeki gelişim, bilimin diğer alanlarında olduğu gibi biyolojik bir modelle değil, doğrudan verilerden öğrenmeye çalışan istatistiksel öğrenme yöntemlerinin benimsendiği görülür. Özellikle çok katmanlı Yapay Sinir Ağları (Artificial neural networks , YSA)'na dayalı Derin Öğrenme (Deep Learning , DÖ) yöntemlerinin popüleritesi artmaktadır [21]. YSA'ların geçmişi Rosenblatt'ın 1958'deki biyolojik olarak ilham alınan Perceptron (Şekil 2.5) modeline kadar uzanır [22]. Yeni YSA modellerinde ilk YSA modellerine göre kısıtlanmış değil, çok sayıda gizli katman içerir [21].



Şekil 2.5. Otonom algılayıcı model (Autonomous Perceptron Model,OAM) modelinin mimarisi. Sagheer 2019, makalesinden yeniden üretilmiştir [24].

Sağlık alanında yapay zeka (Artificial Intelligence,YZ) aktif kullanıma sahiptir. En çok kullanılan tekniklerden birisi gözetimli öğrenme ile tahmin etme yöntemidir. Tahmin etme Makine Öğrenimi (Machine Learning, MÖ) ve DÖ alanı içinde en gelişmiş yöntemdir. Bilgisayar görüntüleme ve görüntü işleme alanı son geliştirilen DÖ algoritmaları ile tıpta önemli bir yer edinmiştir [25]. Robotik yapay zekada ise Da Vinci cerrahi sistemi örnek gösterilebilir [26].

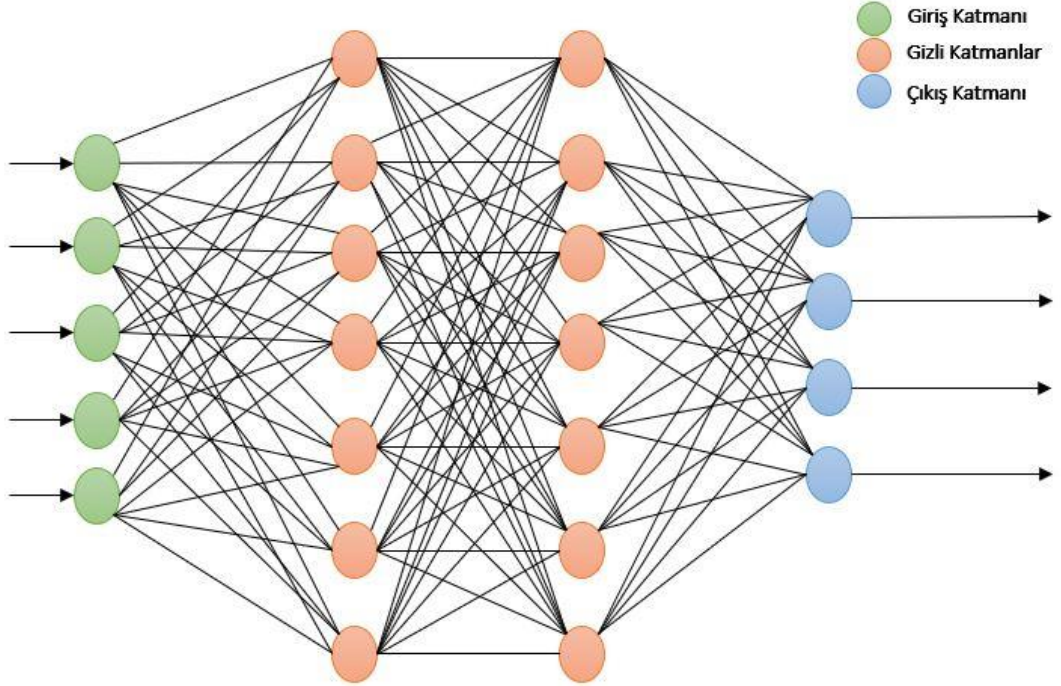
MÖ modelinin yapısı sadece öğrenme algoritmasının bir veri tabanına uygulanmasını içermez [27,28]. Genellikle Şekil 2.6'da gösterilen adımları içeren bütün süreçtir. MÖ modeline ilk bakışta istatistikle benzer olduğu gözlemlenir. Öncelikle bir veri tabanı oluşturulur. Bu veri tabanındaki en önemli değişkenler belirlenir ve keşifsel bir analiz gerçekleştirilir. Değişkenler doğrudan makine öğrenimine uygulanır. Oluşturulan değişkenler veri kümesinin üç alt grubuna parçalanır. Eğitim, validasyon ve test olarak sırasıyla %60,%20 ve %20 olarak bölünür[26]. MÖ'de genel olarak tercih edilen algoritmalar Logic Regresyon, Random Forest, XGboost vb. gibi çok çeşitli sınıflandırma ve regresyon algoritmaları mevcuttur[29].



Şekil 2.6. Bir makine öğrenimi modelinin uygulama metodolojisi. V, veritabanı; MÖ, makine öğrenimi [26].

DÖ modeli çok sayıda nöron katmanından oluşan Yapay Sinir Ağları (Artificial Neural Network, YSA) mimarilerini içerir. YSA'lar beyin nöronlarında meydana gelen öğrenme sürecine benzer. Bu algoritma daha az miktarda eğitim verisi ile çalışıp doğru sonuçlar sunabilir. YSA 'lar denetimli, denetimsiz ve pekiştirmeli öğrenme olarak kullanılabilir [25,30].

DÖ için iki temel eğitim türü vardır. Bunlar denetimli ve denetimsiz eğitimidir. Denetimli eğitim sinir ağlarının istenilen çıktıyı bildiği ve ağırlık katsayılarını ayarlaması ile hesaplanan istenilen çıktıya olabildiğince yakın tahminlerde bulunur. Çok Katmanlı Algılayıcılar (Multi-Layer Perceptrons, ÇKA) buna örnek olabilir [31]. ÇKA'lar ileri besleme ağlarının özüdür. ÇKA'lar bir veya birden fazla nörona sahip gizli katmanlardan oluşur (Şekil 2.7). ÇKA girdileri bir ağırlık ile çarpılır. Tüm ağırlıkların toplamı ve eğilim değeri hesaplanır. Buradaki toplama işleminin sonucu bir aktivasyon fonksiyonuna uygulanır ve aktivasyon çıktısının sonucu nöron çıktısı olarak kabul edilir. Her katman için bu durum tekrarlanır [32]. Aktivasyon fonksiyonları(AF) ağı doğrusal olması durumunu tanıtmak ve ağı karmaşık doğrusal olmayan işlevleri öğrenmesini mümkün kılmak için kullanılır. Bu fonksiyonlardan Sigmoid ve Tanh yaygın olarak yapay sinir ağlarında kullanılır [33,34,35].



Şekil 2.7. Çok katmanlı algılayıcı. Yeşil renk giriş katmanı, pembe renk gizli katmanı bu gizli katmanlar arttırılabilir ve mavi renk çıktı katmanını temsil eder. Her çizgi ağ için ağırlığı temsil eder ve bu ağırlıklar eğitim sürecinde belirlenir.

MÖ ve DÖ bazı kısıtlamalar veya zorluklar ile karşılaşabilir. Bunlardan ilki, doğrusal bir veri elde etmek için gereken veri miktarı oldukça önemlidir. Veri miktarının çok olması tıp ve medikal gibi alanlarda etik kriterlerden dolayı elde etmek oldukça zordur. Veri azlığından dolayı MÖ ve DÖ algoritmaları ne kadar sonucu doğru bulsa bile bazen yeni bir hasta verisini tespit etmede yetersiz olabiliyor. Bundan dolayı bir algoritmanın gerçekçi ve doğru olması için veri kaynağının güvenilir ve kalitesi esas alınır [36]. Makine öğrenimi bize tahminler biçiminde sonuçlar verir, fakat biyolojik bir açıklama sunamaz. Her ne kadar bir modelin sonuçlarını yorumlamak ve değişkenlerin belli ağırlıklar ile analizi yoluyla doğru çalıştığını ve bir tahminin en ayırt edici kısımlarını vurgulayan çeşitli metodolojiler olmasın rağmen, MÖ ve DÖ karmaşık modellerinin neden daha belirli bir karar verdiklerini şu anda tam olarak bilmek mümkün değildir [37].

MÖ ve DÖ frameworkleri ortak aktivasyon fonksiyonları, gradyan hesaplamalarını, eğitim gibi işlemleri grafik işleme birimleri (graphics processing units, GİB) veya tensör işleme birimleri (tensor processing units, TİB) ile sağlar [38]. En yaygın olarak kullanılan açık kaynak programlama dili şu anda Python'dur [25]. Python ile kullanılabilen TensorFlow (TF), bu frameworkler arasında en yaygın olarak kullanılanıdır [39]. TF, genel ve çeşitli hesaplama görevlerine uygulanabilir olacak şekilde tasarlanmış olsa da çoğunlukla makine öğrenimini, özellikle derin ağ mimarilerini hedeflemektedir [38]. TF, düşük- seviyeli framework olarak adlandırılır. Her uygulama için düşük-seviyeli framework ile kullanmak oldukça zordur, bu yüzden DÖ yöntemlerinin çoğunda olduğu gibi Keras gibi kullanıcı dostu daha yüksek seviyeli UPA'lar (uygulama programlama arayüzü) ile kullanılır [40]. TF, kadar popüler olmasa da TF gibi benzer özellikleri sunan Pytorch frameworkü de mevcuttur [41]. Daha çok MÖ için kullanılan Scikit-Learn 'üde bu frameworkler arasına katılabilir [42].

2.4. ZOR ENTÜBASYONDA YAPAY ZEKA ALGORİTMALARININ KULLANIMI

Zor entübasyon da kullanılabileceği düşünülen MÖ ve DÖ algoritmalar için araştırmalar yapılmıştır. Zor laringoskopi için Moustafa et al. (2017), çalışmasında kullanılan yapay zeka algoritması J48 Karar Ağaçları algoritması seçilmiştir. Kullandığı parametreler vücut kitle indeksi, mandibula uzunluk, interinsizör aralık, tiromental mesafe, sternomental mesafe, mallampati skoru, üst dudak ısırma testi ve atlanto-oksipital eklem ekstansiyonudur.

Buradaki veriler bilgisayarın anlayacağı sayısal değişkenlere çevrilmiştir. Kullanılan MÖ programı WEKA 'dır. Genel olarak sistemin eğitim sonrası test verilerinden bulduğu sonuç %76 olarak saptanmıştır. Bu verilerin zor laringoskopiye tarafsız bir bakış açısı getirildiği düşünülmüştür [43]. Jestin et al. (2016),_endotrakeal entübasyonun başarı oranının anestezi uzmanının deneyimine göre değiştiği belirtilmiştir. Bundan dolayı video laringoskop ile zor trakeal entübasyonun yapay zeka yöntemleriyle daha başarılı sonuç verilmesi amaçlanmıştır. Sistemde birden fazla sınıflandırma algoritması denenmiştir. Bunlar k-en yakın komşu (KEK), destek vektör makinesi(DVM), YSA ve karar ağaçları. Algoritmaların sonucunda aldığı doğruluk oranları ise KEK%80, DVM%81, karar ağaçları%74 ve YSA%75'dir [44]. Cook et al. (2019), yaptığı çalışma da ise karar ağaçları yöntemi ile önceden belirlenmiş bir algoritmayı takip ederek zor havayolunu doğrudan laringoskopinin başarısız olduğu yerde video laringoskopi ile daha başarılı olmasının amaçlamasıdır. Bilgisayar modellemesinin hangi hastaların daha kolay entübasyon olabileceğini ve video laringoskopi ile hangi hastanın daha zor entübasyon olabileceğini tahmin edilmesini sağlamıştır [45]. Zor hava yolunu pediatrik alandaki problemi için Clyde et al. (2019) yapay zekâ, makine öğrenimi ve derin öğrenmenin potansiyel kullanımını tanıtmaya çalışmışlardır [46].

Bu çalışmada DÖ algoritmalarından denetimli öğrenme yapısı altında olan yapay sinir ağları yöntemi ile mevcut verileri eğitim, validasyon ve test sonuçları işlenmiştir. Bu verilerin işlendiği framework ise TensorFlow'un Keras UPA ile yapılmıştır. Çalışmanın çözmeyi amaçladığı sorun insan faktörünü ortadan kaldırarak kararı veriler doğrultusunda en hassas ve hızlı yöntemle hastaya teşhis konulabilmesini amaçlanmıştır.

3. MALZEME VE YÖNTEM

3.1.HAM VERİLER

İstanbul Üniversitesi-Cerrahpaşa Kardiyoloji Enstitüsü'nden alınan hasta verileri ile hastaların entübe edilmeyen, entübe edilen ve zor entübasyon olmalarına göre bir veri seti oluşturuldu. Ek-1'de verilerin kullanım izninin alındığını gösterir doküman yer almaktadır. Toplam oluşturduğumuz hasta sayısı 157 kişi oluşmaktadır. Veriler toplanırken yaş, cinsiyet, boy, kilo, mallampati derecesi, atlantookipital eklem ekstansiyonu derecesi, tiromental mesafe, sternomental mesafe, üst diş oklüzal horizontal açı, Cormack ve Lehane sınıflaması parametreleri göz önüne alınmıştır. İstanbul Üniversitesi-Cerrahpaşa Kardiyoloji Enstitüsü'ne başvuran son iki (2019-2021) yılın tüm hastaları kullanılmıştır. Tablo 3.1'de gösterilen parametreler İstanbul Üniversitesi-Cerrahpaşa Kardiyoloji Enstitüsünden sağlanan 157 veriden rastgele seçilen 5 veridir. İlgili veriler talep halinde paylaşılacaktır. Burada veriler kısmında hastaların yaş, cinsiyet ve kilo bilgileri açık veril olarak bırakılmış diğer sütunlardaki veriler kapatılmıştır. Bu parametrelerin sonucunda hastada karar verilen müdahale entübasyon durumundaki sütunda gösterilmektedir. Entübasyon durumundaki değerlerin anlamları;

0: Entübe edilmeyen hasta

1: Entübe edilen hasta

2: Zor Entübe edilen hasta

Buradaki sonuçlara bakıldığında zor entübasyon oldukça zor karşılaşılan bir durum olduğu gözüküyor. Elimizde bulunan toplam hasta sayısını oranladığımızda 157 hastanın %5 kadarı zor entübasyona tabii olmuştur.

Tablo 3.1. İstanbul Üniversitesi-Cerrahpaşa Kardiyoloji Enstitüsü'nden alınan rastgele 5 hastanın parametrik tablosu.

ID	Yaş	Cinsiyet	Boy	Kilo	Mallampati	Atlantookipital Eklem Ekstansiyonu	Tiromental Mesafe	Sternomental Mesafe	Üst Diş Oklüzal Horizontal Açı	Cormack Lehanne Sınıflaması	Entübasyon Durumu
1	59	k	150	78	2	1	7	14	40	1	1
2	58	k	165	84	3	1	7	13	40	2	1
3	64	k	160	69	1	1	8	12	40	0	0
4	52	k	155	62	1	1	6	13	50	0	0
5	65	k	160	72	4	2	6	11	35	3	2

3.2.HAM VERİLERİN KULLANILDIĞI KÜTÜPHANELER VE YAPAY ZEKA ALGORİTMALARI

Tablo 3.1 'deki verileri anlamlandırıp bir yapay zeka algoritması ile kullanmak için pythonun Keras ve Scikit-learn kütüphanelerinden yararlanıldı. Keras'ın derin öğrenme alanında ve sonuçları grafiklendirme aşamasında daha avantajlı olduğundan kullanımına önem verildi. Scikit-Learn'de kullandığımız algoritmalar ise çok katmanlı algılayıcılar sınıflandırması (ÇKAS), rastgele orman (RO), karar ağaçları (KA), XGBoost, k- en yakın komşu (KEK) ve destek vektör makinesi (DVM)'dir [49]. Scikit-Learn kütüphanesini kullanma amacımız başta verileri karışık bir şekilde eğitim ve test olarak bölebiliyor olmasıdır. Daha sonra bu kütüphanenin ve Keras kütüphanesinin sonuçları gözlemlenmiştir.

Keras YSA yapısı ve Scikit-Learn YSA yapısı birbirinin benzeridir. Kullanılan diğer algoritmalar bu iki yapıdan farklılık göstermektedir.

3.2.1.Yapay Sinir Ağları

Bu veride yapay sinir ağları algoritmasına önem verilmesindeki düşünce, yapay sinir ağları sayısal verileri işlemekte daha başarılı olduğunun kabul edilmesidir. Veriler daha çok ve karmaşıklaşmış olsa bile YSA' nın özel ölçeklendirme yöntemi ile verilerin bir yöne yönelimi engellenmiş olur. Her iki kütüphanede de “MinMaxScaler()” ya da “Scaler.fit()” komutları ile veriyi 0 ile 1 arasındaki değerlere ölçeklenmiştir.

Keras ve Scikit-Learn kütüphanesinin YSA algoritmasının çalışma prensibi aslında matematiksel olarak aynı formül ile yapılır (Şekil 3.1). Matematiksel formül için Denklem 3.1'deki model kullanılır. Denklem 3.1'deki modele göre;

- X_n = Girişlerdir. Yapay sinir ağlarına dış dünyadan veya diğer bir hücreden gelen bilgilerdir.
- W_n = Ağırlıktır. Hücreler arasındaki bağlantıların sayısal değerini ifade eder. Bir hücreye gelen bilginin değerini gösterir.
- b = Bias bir ağırlıktır. Bir sinir ağında öngörülemeyen veya gözlemlenemeyen faktörleri sağlamaya çalışır.
- Σ = Toplama fonksiyonu. Hücreye gelen girdileri ağırlıklarla çarpıp toplayarak o hücrenin net girdisinin hesaplanmasını sağlar.
- F_j = Aktivasyon Fonksiyonu. Hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktının belirlenmesini sağlar.

$$NET = b + \sum_{i=1}^n x_i \cdot w_i \quad \text{ÇIKTI} = F(NET) \quad (3.1)$$

3.2.2.Yapay Sinir Ağları Aktivasyon Fonksiyonları

Yapay sinir ağlarında kullanılan aktivasyon fonksiyonları Keras ve Scikit-Learn için birbirinden farklıdır. Hatta Keras'ta kullanılan aktivasyon fonksiyonları her bir model için ayrı seçilmesine olanak sağladığından en uygun aktivasyon fonksiyonları seçilmiştir.

Kerasta “Sequential()” fonksiyonu ile 3 farklı model belirlenmiştir. “Sequential()” fonksiyonu istenilen sayıda istenildiği kadar nöronu bir araya getirip işlem yapılmasına olanak sağlar. Eklenen modellerin AF’ları sırası ile relu, linear ve softmax ‘tir. Bu modellerin nöron sayısı sırası ile 60, 100 ve 12’dir. Kullanılan AF’lerin matematiksel tanımı aşağıda tanımlanmıştır.

Relu fonksiyonu, çıktının sıfırın veya negatifin altına düşmemesini sağlar. Gelen girdi sıfırdan büyük ise değişmeden çıktı olarak verilir. Eğer sıfırdan küçük veya eşit ise çıktıyı sıfır olarak verir (Denklem 3.2). Bu fonksiyon Sckit-Learn’de de aktivasyon fonksiyonu olarak kullanılmıştır.

$$F(\text{NET}) = \text{Maks}(0, \text{NET}) \quad (3.2)$$

Linear fonksiyonu, doğrusal aktivasyon fonksiyonları girdi değerlerinin değiştirmeden çıktı olarak verir (Denklem 3.3).

$$F(\text{NET}) = \text{NET} \quad (3.3)$$

Softmax fonksiyonu, bir girdi değerini, toplamı 1’e kadar olan olasılık dağılımını izleyen değerler vektörünü normalleştiren bir lojistik regresyon biçimidir (Denklem 3.4).

- $\vec{z} = (z_0, \dots, z_K)$ 'den oluşan softmax işlevine giriş vektörü.
- $z_i =$ Tüm z_i değerleri, softmax işlevine giriş vektörünün öğeleridir ve pozitif, sıfır veya negatif herhangi bir gerçekteği değeri alabilirler.
- $e^{z_i} =$ Giriş vektörünün her elemanına standart üstel fonksiyonu uygulanır. Bu, girdi negatifse çok küçük ve girdi büyükse çok büyük olacak olan 0'ın üzerinde pozitif bir değeri verir. Ancak yine de bir olasılık için gerekli olan (0, 1) aralığında sabit değildir.
- $\sum_{j=1}^K z_j =$ Formülün altındaki terim normalizasyon terimidir. Fonksiyonun tüm çıktı değerlerinin toplamının 1 olmasını ve her birinin (0, 1) aralığında olmasını sağlar, böylece geçerli bir olasılık dağılımı oluşturur.
- $K =$ Çok sınıflı sınıflandırıcıdaki sınıf sayısı.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K z_j} \quad (3.4)$$

3.2.3. Karar Ağaçları Ve Rastgele Orman Algoritması

Rastgele orman, her adımda bir miktar rastgeleleştirme sunmaya çalışır. Rastgele orman algoritmasının her adımda sunduğu önerileri son tahmini yapabilmek için bir araya getirir [47]. Yapay sinir ağlarından farklı olarak bu tip verilerde genel olarak RO'da kullanıldığını yapılan literatür araştırmalarında gözlemlenmiştir. Bu sayede verinin grafikler dışında algoritmaların öğrendiklerini ya da ezberlediklerinin ufak bir göstergesi olarak görülebilir. RO'nun bu tezdeki ikincil durumu ise hangi parametrenin ne kadar kullanıldığını gösteriyor olmasıdır.

Rastgele orman algoritmasında aslında ana çıkış noktası olan karar ağaçlarına (KA) da değinilmesi gerekir. KA, hedef değeri tahmin etmek için veri kümesini daha küçük ve daha küçük alt kümelere nasıl ayıracağını öğrenir. Durum yaprak ya da düğüm olarak, olası sonuçlar ise dallar ya da kenar olarak tanımlanır. Karar ağaçları maksimum derinliğe ulaştığında ya da belirlenen koşulu sağlayana kadar tekrarlama işlemine devam eder. Daha önceden bahsedilen rastgele orman ise bünyesinde birden fazla KA barındırır. Rastgele ormanın son tahmini için daha önceden bahsedilen tanımını açıklayıcı hale getirmek gerekirse bünyesinde bulundurduğu tüm ağaçlardan gelen sonuçlara bakarak nihai sonuca ulaşır.

Scikit-Learn bu işlemi matematiksel olarak önce iki çocuk düğüme bakarak hesaplar [49] (Denklem 3.5).

$$n_{ij} = w_j C_j - w_{sol(j)} C_{sol(j)} - w_{sağ(j)} C_{sağ(j)} \quad [49] \quad (3.5)$$

- n_{ij} = j düğümünün önemi
- w_j = j düğümüne ulaşan ağırlık sayısı
- C_j = j düğümünün saflık derecesi
- $sol(j)$ = j düğümünde sola bölünmüş alt bölüm

- sağ(j) = j düğümünde sağa bölünmüş alt bölüm

Bir Karar ağacındaki her bir özellik öneminin hesaplanması aşağıdaki gibi yapılır [49] (Denklem 3.6).

$$f_i = \frac{\sum_{j:\text{düğüm } j \text{ özellik üzerine bölünmüş}} n_{ij}}{\sum_{k \in \text{tüm düğümler}} n_k} \quad [49] \quad (3.6)$$

- f_i = i özelliğinin önemi
- n_{ij} = j düğümünün önemi

Bu sonuçlar daha sonra 0-1 aralığına normalize edilir [49] (Denklem 3.7).

$$\text{norm}f_i = \frac{f_i}{\sum_{j \in \text{tüm özellikler}} f_j} \quad [49] \quad (3.7)$$

Rastgele orman düzeyindeki son özellik önemi, tüm ağaçların ortalamasıdır. Her ağaçtaki özelliğin önem değerinin toplamı hesaplanır ve toplam ağaç sayısına bölünür [49] (Denklem 3.8).

$$ROf_i = \frac{\sum_{j \in \text{tüm ağaçlar}} \text{norm}f_{ij}}{T} \quad [49] \quad (3.8)$$

- ROf_i = Rastgele Orman modelinde tüm ağaçlardan hesapladığımız özelliğin önemi.
- $\text{norm}f_{ij}$ = j ağacında i için normalleştirilmiş özellik önemi.
- T = Toplam ağaç sayısı

3.2.4.XGBOOST Algoritması

XGBoost, hız ve performans için tasarlanmış gradyan yükseltme karar ağaçlarının bir uygulamasıdır. XGBoost aslında rastgele orman algoritması gibi temeli karar ağaçlarına dayanmaktadır. Gradyan yükseltme algoritma ise karar ağaçlarının aksine ilk kök değil öncelikli olarak ilk yaprak oluşturur. Tahmin hatalara bakarak yeni ağaçlar oluşturur. Bu yöntem elindeki karar ağaçları bitene kadar ya da daha fazla gelişme olmayıncaya kadar devam eder.

XGBoostun en önemli özellikleri arasında yüksek tahmin gücü, boş verileri yönetebilme, aşırı uyumluluğun önüne geçme ve bunları oldukça hızlı çözümlenmeyi bize sunar. Daha az kaynak kullanarak üstün sonuçlar elde edilmesini sağlar. Bu durum medikal alanda algoritmalar arasında ön plana çıkmasını sağlar.

XGBoost'un performansının iyi olmasının sebepleri; regülarizasyon, budama, boş değerler ile çalışabilmesi ve sistem optimizasyonudur. XGBoost'u formüle etmek için önce AdaBoost'u açıklamak gerekir.

AdaBoost kök, bir düğüm ve iki yapraktan oluşan derinliği bir olan karar ağaçlarını kullanır. Kümülatif bir yapı olduğundan sıralama çok önemlidir. Yani ardışık bir hesaplama yapmaktadır. İlk karar ağacının yaptığı hata, ikinci ağacın ağırlıklarını etkiler. AdaBoost sonucu Denklem 3.9 'da ki gibi hesaplanmaktadır.

$$H_{final}(x) = \text{sign}(\sum_t \alpha_t h_t(x)) \quad (3.9)$$

- $h_t(x)$ = Köklerden birinin oyudur.
- α_t = O köklerin oyunun ağırlığı.

İki sınıflı bir sınıflandırmada ağırlığı Denklem 3.10, çok sınıflı bir sınıflandırmada ise ağırlığı Denklem 3.11 ile hesaplanır.

$$\alpha_t = \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \quad (3.10)$$

- $\alpha_t = \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) + \ln C - 1$ (3.11)
- ε_t = Kökün hata payı.
- C = Sınıf sayısı.

AdaBoost ile hesapladığımız tüm örneklerin kendi ağırlığını hesaplayıp (Denklem 3.12), yanlış tahmin edilen örneklere yeni örneklerin ağırlıklarını atanmaktadır.

$$W_{new} = S_w x e^{K_w} \quad (3.12)$$

- W_{new} = Yeni örnek ağırlıkları.

- S_w = Örnek ağırlıkları.
- K_w = Kökün ağırlığı.

AdaBoost'u matematiksel olarak tanımladıktan sonra gradyan yükseltme algoritması ile XGBoost algoritmasını açıklamak gerekmektedir.

Gradyan yükseltme aynı AdaBoost gibi tahmin edicileri ekleyerek çalışır. Fakat AdaBoost da olduğu gibi her yinelemede örnek ağırlıkları değiştirmek yerine bu yöntemde tahminci önceki tahmin edici tarafından yapılan hatalara uydurmaya çalışır [42].

3.2.5.K-En Yakın Komşu Algoritması

K-en yakın komşu algoritması sınıfları belli olan bir örnek veri setine katılacak olan yeni verinin mevcut verilere olan uzaklığı hesaplanır. Hesaplanan verinin k sayıda yakın komşularına bakılır.

KEK'i matematiksel olarak anlamak için kullandığımız uzaklık hesaplama yöntemlerini incelemek gerekir. Bu çalışmada 3 tip uzaklık türü sırası ile Öklid (Denklem 3.13), Manhattan (Denklem 3.14) ve Minkowski (Denklem 3.15)'dir.

Öklid en basit olarak iki nokta arasındaki uzaklığa bakar.

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3.13)$$

- x_1 = A noktasındaki x mesafesi
- x_2 = B noktasındaki x mesafesi
- y_1 = A noktasındaki y mesafesi
- y_2 = B noktasındaki y mesafesi

Manhattan noktalar arasındaki mutlak uzaklıkların toplamını hesaplar.

$$\sum_{i=1}^k |x_i - y_i| \quad (3.14)$$

- k = Komşu sayısı.
- $x_i = i$ kadar x noktaları.
- $y_i = i$ kadar y noktaları.

Minkowski q sayıda değişkene bağlı bir uzaklık hesabı yapar. Q değerinin yerine 2 rakamı yazıldığında Öklid formülünü elde ettiğimiz görülecektir.

$$\left[\sum_{i=1}^k (|x_i - y_i|)^q \right]^{1/q} \quad (3.15)$$

- q = Kaç sayıda değişkene bağlı olduğunu gösterir.

Uzaklık hesaplamaları yapıldıktan sonra yaklaşık değerlere bakılarak uzaklıklar matrisi oluşturulur. Matris hesaplandıktan sonra uzaklığı en az olan iki nokta arasını alarak gruplama işlemini yapmaktadır. K -en yakın komşunun zaten amaçladığı noktaları en yakında bulunan noktalar ile gruplayarak yeni gelen değer için hangi grup içinde yer alacağını belirlemektir.

3.2.6. Destek Vektör Makineleri Algoritması

Destek vektör makinesi, her bir verinin belirli bir koordinatının değeri olan her özelliğin değeri ile birlikte n özellikli bir nokta çizilir. Bu işlemden sonra iki sınıfı çok iyi ayırdığı için hiper bir düzlem bularak sınıflandırma işlemini gerçekleştirir.

DVM’de varsayalım ki iki farklı noktamız bulunsun. Bu noktalardan biri mavi diğeri kırmızı renkte olsun. Burada noktaları iki sınıfa ayırmış olduk. Sınıflandırma yapmak için elimizde bulunan iki sınıfı ayıran bir doğru çizilir. Doğrunun ± 1 arasında kalan bölgesine “Margin” olarak adlandırılır. Margin’in geniş olması iki veya daha fazla sınıfın ne kadar iyi ayrıştıracağı anlamına gelir. Mavi noktaların bu çizilen Margin’in alt bölgesinde, kırmızı noktaların Margin’in üst bölgesine yerleştiğini varsayalım.

$$\hat{y} = \begin{cases} 0 & \text{ise } w \cdot x + b < 0, \\ 1 & \text{ise } w \cdot x + b \geq 0 \end{cases} \quad (3.16)$$

- w = Ağırlık.
- x = Girdi vektörü.
- b = Sapma.

Denklem 3.16 'ya göre yeni gelen değer için sonucumuz sıfırdan küçükse mavi noktalara yakın konumlanacaktır. Eğer yeni gelen değer için sonucumuz sıfıra eşit veya sıfırdan büyükse yeni gelen değerimiz kırmızı noktalara yakın konumlanacaktır.

3.3.VERİLERİ İŞLEMEDE KULLANILAN YARDIMCI KÜTÜPHANELER VE FONKSİYONLARI

Ham veriyi işlemek için önce Excel dosyasını Python'un Pandas kütüphanesi ile içeri aktarılır. Pandas ile buradaki ham veriyi DataFrame olarak ayrı bir değişken haline çevirip kopyalanır. Bunu yapılmasındaki amaç ham verinin bozulmasını engellemektir. DataFrame olarak kopyalanan verideki eksik verileri incelenir. Tablo 3.1'deki veri düzenlenen veridir. Hastalık geçirmiş olanlar nümerik olarak anlamlandırılmadığı için ilgili parametreyi bu veriden kaldırmak durumunda kalınmıştır. Bu durumun sebebi ise bir hasta birden fazla hastalık geçirdiyse tek parametrede birden fazla hastalığı sayısal olarak incelenememesidir. Hastalıkların önem sırası tanımlanamadığı için bu parametre göz ardı edilmiştir. Toplam 4 hasta verisinde bazı parametreler eksik olduğu tespit edilmiştir. Veri azlığından dolayı eksik parametrelili hastaları kaybetmek veri için oldukça zararlı olacağından, ilgili parametredeki verilerin ortalama değeri eksik parametrelere atanmıştır. Cinsiyet parametresinin nümerik olarak anlamlandırmak için Sklearn kütüphanesinin LabelEncoder fonksiyonundan yararlanıldı, kızları "1" erkekleri "0" olarak tanımlandı.

3.4.EĞİTİM, VALİDASYON VE TEST VERİ SETİ ANALİZİ VE VERİ SETİNİ AYRIŞTIRMA YÖNTEMLERİ

Verinin algoritmalarda kullanılabilir hale getirildikten sonra ikinci aşama olan derin öğrenme ve makine öğrenmesi algoritmaları kullanıldı. Eğer veri son haline getirilmeseydi anlamsız veriler ile elde edilen sonuç oldukça kötü sonuç verecektir. Veri, oluşturduktan sonra üç ana parçaya ayrılmıştır. Bunlar; Eğitim Veri Seti, Validasyon Veri Seti ve Test Veri Setidir.

- Eğitim veri seti temel modellemeleri yapılarak üzerinde en doğru makine öğrenmesi algoritması seçmeyi sağlar. Verinin büyük bir kısmını kullanır. Buda verinin %75'ini kapsamaktadır.
- Validasyon veri seti ise eğitim veri seti içinden seçilir. Eğitim veri setinde uygulanan modelin iyileştirilmeye çalışıldığı veri setidir. Eğitim veri setinin %25'ini oluşturur.
- Test veri seti, eğitim veri setini ayırırken kalan %25'lik veri setidir. Bu veri seti asla eğitim ve validasyon verisine karışmaz.

Bu verileri en optimum şekilde ayırmak için sklearn kütüphanesinden yararlanıldı. Sklearn “train_test_split” fonksiyonu sayesinde veriyi rastgele bir şekilde istenilen yüzdesel rakamlarla eğitim veri seti ve test veri seti olarak bölünmesine olanak sağlar. Eğer bu fonksiyon kullanılsaydı, eğitim ve test veri setini el ile seçerek ayırmak gerekecektir. Buda işlemin güvenilirliğini azaltır. Validasyon veri setini ise Keras kütüphanesinin “Sequential().fit” fonksiyonu ile belirlenmiştir.

3.5.KERAS VE SCİKİT-LEARN KÜTÜPHANESİNDE KULLANILAN PARAMETRELER

Keras ile yapılan işlem Scikit-Learn kütüphanesinin çok katmanlı algılayıcılar sınıflandırma algoritması ile de gerçekleştirildi. Burada yapılan değişiklik ise ÇKAS optimize ederek işlemi gerçekleştirmektir. Optimize etme işlemi çok katmanlı algılayıcılar sınıflandırma algoritmasının kullanacağı aktivasyon(activation), alfa(alpha), gizli katman sayısı(hidden_layer_sizes), öğrenme oranı(learning_rate), çözücü(solver) gibi parametrelerini varsayılan değerleri ile işleme sokmak yerine içlerinden en iyi değerlere sahip parametreleri bulunmasını sağlar. Bu parametrelerin karşılıklarına gelecek birkaç farklı kombinasyon yazılır ve bunun sonucunda en iyi bulduğu değerler ile verilerin sonucuna bakılır. Aynı işlem RO algoritmasını kullanırken de gerçekleştirilmiştir.

Keras ile oluşturulan modelde kayıp fonksiyonu olarak seyrek kategorik çapraz entropisi(sparse categorical crossentropy) kullanıldı. Keras dokümanının da bu fonksiyonu etiketler ve tahminler arasındaki çapraz entropi kaybını hesaplar. İki veya daha fazla etiket sınıfı kullanıldığında bu çapraz entropi kaybının kullanılması gerekir [48]. Bu çalışmada

kullanılan veride birden fazla etiket sınıfı olduğundan bu yapıyı kullanıp sonuçları incelenmiştir.

Kategorik çapraz entropinin bir alt dalı olan seyrek kategorik çapraz entropi aslında aynı formül ile kullanılmaktadır (Denklem 3.17). Bu iki çapraz entropinin en büyük farkı gerçek etiketleri bir ve sıfırdan oluşuyorsa üç sınıflı bir sınıflandırmada kategorik çapraz entropi kullanılmaktadır. Eğer gerçek etiketler tamsayı ifadeleri ile ifade edilmişse seyrek kategorik çapraz entropi kullanılır.

$$w = \frac{1}{n} \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \quad (3.17)$$

- y_i = Gerçek etiket değerleri.
- \hat{y}_i = Tahminin etiket değerleri

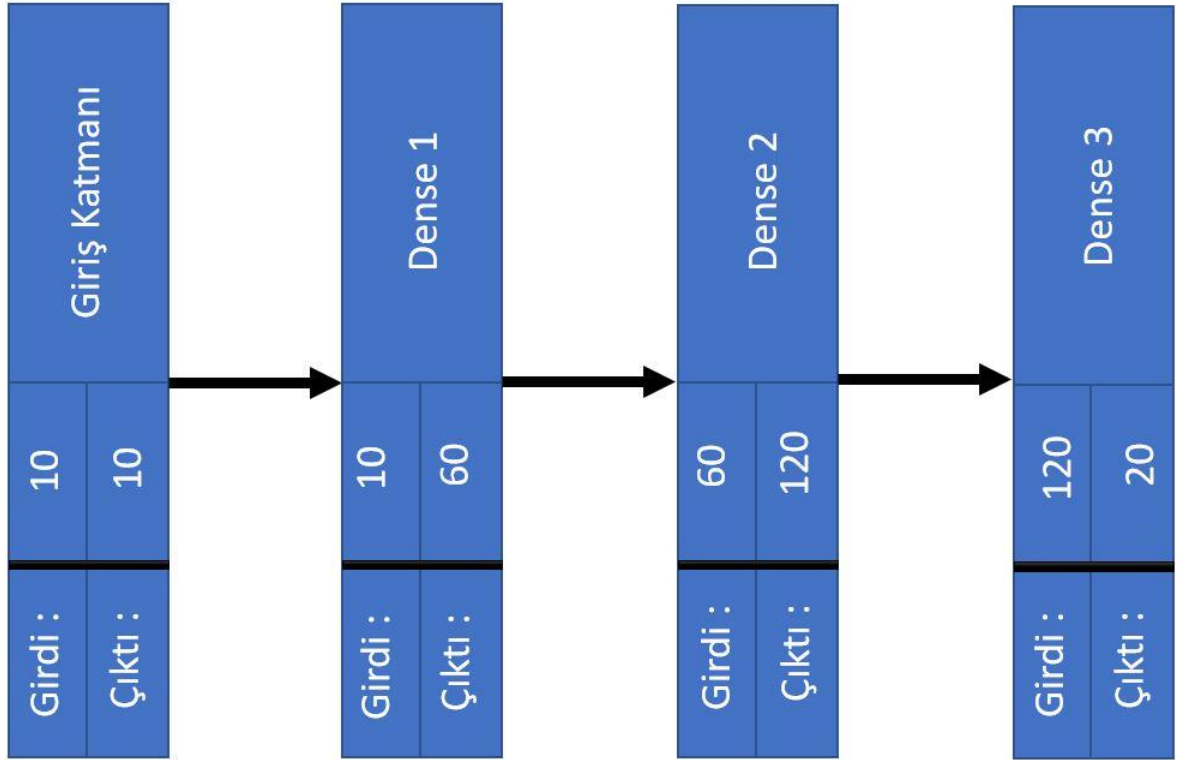
3.5.1.Keras Yapay Sinir Ağlarında Kullanılan Hiperparametreler

Keras yapay sinir ağları ile oluşturulan modelde 3 farklı gizli katman sayısı kullanılmıştır. Her gizli katman sayısı için ayrı aktivasyon fonksiyonlarına yer verilmiştir. Algoritmada kullanılan modelde ilki girdi sayısıdır. Bu tezde kullanılan model yapısı ise 10 girdi daha sonrasında 60 nöronlu ilk gizli katman sayısı ve kullanılan aktivasyon fonksiyonu Relu, ikinci gizli katmanda nöron sayısı 120 kullanılan aktivasyon fonksiyonu Linear ve son gizli katmanda nöron sayısı 20 kullanılan aktivasyon fonksiyonu ise Softmax olarak seçilmiştir. Keras model oluşturma işleminde bu esnekliği sağlayan fonksiyon “Dense” fonksiyonudur. Kısaca Dense Denklem 3.18 deki gibi Keras dokümanında tanımlanır [48]. Tüm modelin özetini Şekil 3.1 ‘de görülmektedir.

$$\text{Çıktı} = \text{aktivasyon (nokta (girdi, çekirdek) + bias)} \quad (3.18)$$

- Girdi = Girdi verisini tanımlar.
- Çekirdek = Ağırlık verisini tanımlar.
- Nokta = Tüm girdilerin ve buna karşılık gelen ağırlıkların numpy nokta ürününü temsil eder.

- Bias = Modeli optimize etmek için makine öğreniminde kullanılan taraflı bir değeri temsil eder.
- Aktivasyon = Aktivasyon fonksiyonunu tanımlar.



Şekil 3.1. Keras yapay sinir ağlarında oluşturulan modelin özeti.

Oluşturulan modeli derlerken kullanılan optimize edici “Adam” olarak, kayıp fonksiyon bir önceki başlıkta bahsedilen kategorik çapraz entropi ve metrik birimi olarak doğruluk tanımlanmıştır. Adam optimizasyonu, birinci dereceden ve ikinci dereceden momentlerin uyarlanabilir tahminine dayanan stokastik bir gradyan iniş yöntemidir. Doğruluk ise tahminlerin etiketlere ne sıklıkla eşit olduğunu hesaplamaktadır [48].

Modeli iterasyona almak için kullanılan gradyan başına güncelleme sayısı 20 ve epoch sayısı 200 olarak belirlenmiştir. Model bu parametreler ile eğitim ve validasyon işlemlerine tabii tutulmuştur.

3.5.2.SCİKİT-LEARN Çok Katmanlı Algılayıcılarda Kullanılan Hiperparametreler

Scikit-Learn çok katmanlı algılayıcılar ile oluşturulan modelde kullanılan hiperparametreler;

- Aktivasyon fonksiyonu: Relu.
- Alfa değeri: 0.2.
- Gizli katman sayısı: 1.
- Nöron sayısı: 100.
- Öğrenme oranı: Sabit
- Çözücü: lbfgs.

Bu parametreler bir optimizasyon işlemi sonucunda seçilmiştir. Optimizasyon işlemi gerçekleştiren Scikit-Learn kütüphanesinin sağladığı “GridSearchCV()” sayesinde. Bu parametreler dışındakiler varsayılan olarak bırakılmıştır. Bu şekilde birçok model kombinasyonunu denenmesi sonucunda en optimal sonuç bulunur. Kullanılan hiperparametrelerin sağladığı özellikleri açıklamak gerekirse;

- Aktivasyon fonksiyonu = Gizli katman için aktivasyon fonksiyonun.
- Alfa değeri = Regresyon terimi parametresi.
- Gizli katman sayısı = Gizli katman sayısını temsil eder.
- Nöron sayısı = Gizli katmandaki nöron sayısı.
- Öğrenme oranı = Ağırlıkların güncellenmesinde adım boyutunu kontrol eder.
- Çözücü = Ağırlık optimizasyonu için çözücü.

3.5.3.Rastgele Ormanda Kullanılan Hiperparametreler

Scikit-Learn rastgele orman ile oluşturulan modelde kullanılan hiperparametreler;

- Maksimum derinlik: 5.
- Maksimum özellikleri: $\log 2$.
- Bölünme için gereken örnek sayısı: 2.
- N sayıdaki tahminçiler: 100.

Bu parametrelerde aynı çok katmanlı algılayıcılar sınıflandırmasında olduğu gibi “GridSearchCV()” fonksiyonu sayesinde birden çok parametrelerin farklı kombinasyonları sonucunda optimize edilip seçilmiştir. Bu parametreler dışındaki parametreler varsayılan olarak bırakılmıştır. Kullanılan hiperparametrelerin sağladığı özellikleri açıklamak gerekirse;

- Maksimum derinlik = Ağacın maksimum derinliğini gösterir.
- Maksimum özellikleri = En iyi bölünme işlemi gerçekleştirirken kullanılması belirlenen özelliklerin sayısıdır.
- Bölünme için gereken örnek sayısı= Bir dahili düğümü bölmek için gereken örnek sayısıdır.
- N sayıdaki tahminçiler = Ormandaki ağaç sayısını temsil eder.

3.5.4.Karar Ağaçlarında Kullanılan Hiperparametreler

Scikit-Learn karar ağaçları ile oluşturulan modelde kullanılan hiperparametreler;

- Maksimum derinlik = 4.
- Bölünme için gereken örnek sayısı= 5.

Bu parametrelerde aynı şekilde Scikit-Learn’ün tüm algoritmalarında uygulanan “GridSearchCV()” fonksiyonu sayesinde birden çok parametrelerin farklı kombinasyonları sonucunda optimize edilip seçilmiştir. Bu parametreler dışındaki parametreler varsayılan olarak bırakılmıştır.

3.5.5.XGBOOST'a Kullanılan Hiperparametreler

Scikit-Learn XGBoost ile oluşturulan modelde kullanılan hiperparametreler;

- Öğrenme oranı: 0.1.
- Maksimum derinlik: 3.
- Bölünme için gereken örnek sayısı: 2.
- N sayıdaki tahminçiler: 100.
- Alt örnek sayısı: 0.6.

Bu parametrelerde aynı şekilde Scikit-Learn'ün tüm algoritmalarında uygulanan "GridSearchCV()" fonksiyonu sayesinde birden çok parametrelerin farklı kombinasyonları sonucunda optimize edilip seçilmiştir. Bu parametreler dışındaki parametreler varsayılan olarak bırakılmıştır. Daha önce açıklanmayan hiperparametreleri açıklamak gerekirse;

- Öğrenme oranı = Öğrenme oranı, öğrenme hızı ile her ağacın katkısını küçültür. Buda algoritmayı hızlanmasını sağlamaktadır.
- Alt örnek sayısı = Bireysel temel öğrenenlere uydurmak için kullanılacak örneklerin oranı.

XGBoost, rastgele orman ve karar ağaçlarının parametrelerin birbirine benzemesi aslında hepsinin temel olarak aynı algoritma ile çalışmasından kaynaklanmaktadır. Bu farklılıkların veri sonuçlarına etkisi kabul edilerek bu çalışmada kullanılmıştır.

3.5.6.K-En Yakın Komşuda Kullanılan Hiperparametreler

Scikit-Learn k-en yakın komşu ile oluşturulan modelde kullanılan hiperparametreler;

- Algoritma: Otomatik.
- Yaprak boyutu: 30.
- Komşu sayısı: 9.
- Ağırlık: Uniform.

Bu parametrelerde aynı şekilde Scikit-Learn'ün tüm algoritmalarında uygulanan “GridSearchCV()” fonksiyonu sayesinde birden çok parametrelerin farklı kombinasyonları sonucunda optimize edilip seçilmiştir. Bu parametreler dışındaki parametreler varsayılan olarak bırakılmıştır. Kullanılan hiperparametrelerin sağladığı özellikleri açıklamak gerekirse;

- Algoritma = En yakın komşuları hesaplamak için kullanılan algoritmalarıdır. Otomatik; “fit” yöntemine sunulan değerlere göre en uygun algoritmayı seçer.
- Yaprak boyutu = BallTree veya KDTree algoritmalarının yaprak boyutunu temsil eder.
 - BallTree = Algoritma bir metrik ağaç olarak düşünülebilir. Metrik ağaçları, noktaların bulunduğu metrik uzayı dikkate alarak veri noktalarını düzenler ve yapılandırır.
 - KDTree = Algoritma veri noktalarını her düğümde iki kümeye böler.
- Komşu sayısı = K-en yakın komşu için varsayılan olarak kullanılacak komşu sayısıdır.
- Ağırlık = Tahminde kullanılan ağırlık fonksiyonu.

3.5.7. Destek Vektör Makinesinde Kullanılan Hiperparametreler

Scikit-Learn destek vektör makinesi ile oluşturulan modelde kullanılan hiperparametreler;

- Kernel = Doğrusal.
- C = 2.

Bu parametrelerde aynı şekilde Scikit-Learn'ün tüm algoritmalarında uygulanan “GridSearchCV()” fonksiyonu sayesinde birden çok parametrelerin farklı kombinasyonları sonucunda optimize edilip seçilmiştir. Bu parametreler dışındaki parametreler varsayılan olarak bırakılmıştır. Kullanılan hiperparametrelerin sağladığı özellikleri açıklamak gerekirse;

- Kernel = Algoritmada kullanılacak çekirdek türünü belirler.
- C = Regresyon parametresidir. Regresyonun gücü ile C ters orantılıdır. Kesinlikle pozitif olmalıdır.

3.6. ALGORİTMALARDA KULLANILAN VALİDASYON YÖTEMLERİ

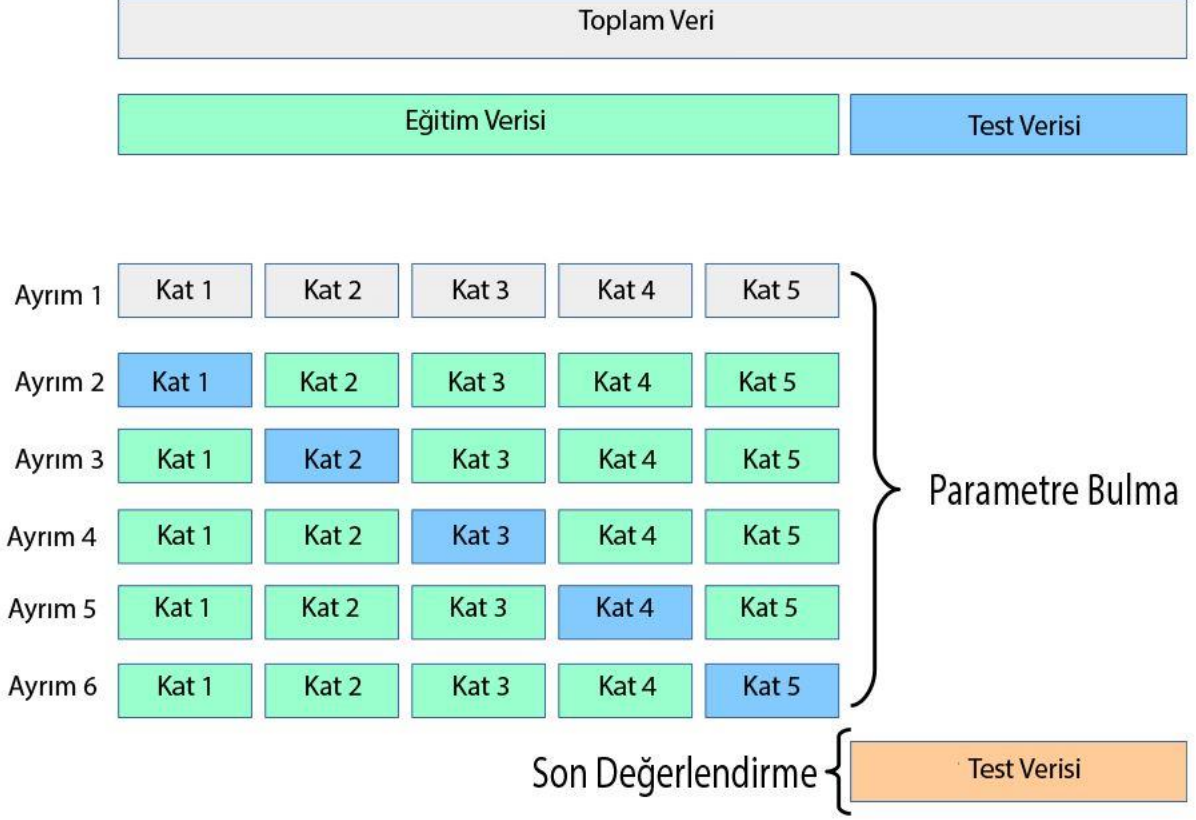
Yapay zekâ algoritmalarında validasyon yöntemi olarak üç farklı yöntem denenmiştir. Bu yöntemler tüm algoritmalara uygulanmamıştır. Yalnızca yapay sinir ağları algoritmasında farklarına bakılmıştır. Bu çalışmada farklı validasyon yöntemlerinin denenme sebebi sonuçlara olan etkisinin incelenmesidir.

3.6.1. Hold-Out Validasyon Yöntemi

Hold-Out validasyon yöntemi verileri eğitim ve test olarak veri setini ikiye ayırır. Hold-Out nispeten basit bir çapraz validasyon yöntemidir. Sistem yalnızca eğitim veri setinin vermiş olduğu sonuçlarla uyum sağlar. Sonraki aşamada daha önce hiç karşılaşmadığı test veri seti ile işleme konulu veri sonuçları değerlendirilir. Bu yöntemin avantajı, genellikle artık yönteminin tercih edilmesi ve artık hesaplanmasının gerekmemesidir. Değerlendirme, büyük ölçüde eğitim kümesinde hangi veri noktalarının ve hangi verilerin test kümesinde sona erdiğine bağlı olabilir ve bu nedenle değerlendirme, bölmenin nasıl yapıldığına bağlı olarak önemli ölçüde farklı olabilir. Bu yöntem Keras yapay sinir ağları modelinde tercih edilmiştir. Validasyon veri seti de eğitim setinin içinden seçilmiştir.

3.6.2. K-Kat Çapraz Validasyon Yöntemi

K-kat çapraz validasyon yöntemi, veri rastgele k alt kümeye bölünür ve Hold-Out yöntemini k kez tekrarlama işlemi gerçekleştirir. Her seferinde, k alt kümelerinden biri test kümesi olur ve diğer k-1 alt kümeleri eğitim kümesi olarak işlem görür. Sonuç olarak tüm k denemedeki ortalama hata hesaplanır. Bu yöntemdeki avantaj her verinin tam olarak bir kez test setinde yer alması ve k-1 kez eğitim setinde yer almasıdır (Şekil 3.2). Bu yöntemin dezavantajı ise sistemde ve işlemde yavaş çalışmasıdır. Bunun ana sebebi ise eğitim veri setini k kez yeniden çalıştırıp ve k kez farklı hesaplama işlemi yapılmasından kaynaklanmaktadır.



Şekil 3.2. K-kat çapraz validasyon yönteminin şematik gösterimi [49].

Bu yöntem Sckit-Learn'de gerçekleştirilen tüm algoritmalar için denenmiştir. K-kat çapraz entropi sayısı tüm hesaplamalarda 10 seçilmiştir. Bunun seçilme nedeni literatürde 5 ile 10 değerlerinin genel olarak seçilmesidir.

3.6.3. Birini Dışarıda Bırakma Yöntemi

Birini dışarıda bırakma (LeaveOneOut) yöntemi, eğitim setindeki biri hariç tüm örnekler alınarak oluşturulur. Test seti bu yöntemde dışarıda kalan örnek olmaktadır. Bu nedenle, örnekler için farklı eğitim setleri ve test setleri oluşturulmuştur. Bu çapraz validasyonda eğitim setinden bir örnek çıkarılması verinin bozulmasına sebep olmamaktadır.

K-katlı apraz validasyon ile karřılařtırıldıđında, $n > k$ olduđu k model yerine n rnekten n model oluřturulur. Ayrıca, her biri $(k-1)n/k$ yerine $n-1$ rnekler zerinde eđitilir. Her iki Őekilde de k'nin ok byk olmadıđı ve $k < n$ olduđu varsayıldıđında, birini dıřarıda bırakma, k-katlı apraz validasyona gre hesaplama aısından daha uzun srelidir. Fakat dođruluk aısından, birini dıřarıda bırakma genellikle test hatası iin bir tahmin edici olarak yksek varyansla sonulanmaktadır.

Bu tezde birini dıřarıda bırakma yntemi Scikit-Learn ok katmanlı algılayıcılar sınıflandırma modelinde kullanılmıřtır. Kullanılma sebebi ise zor entbe hastalarının diđer sonulara gre az olduđundan veri bozukluđuna neden olmaktadır. Bu nedenle k-apraz validasyon yerine birini dıřarıda bırakma yntemi denenerek sonular incelenmiřtir. Ancak belirtmek gerekir ki bu yntem olduka maliyetli ve sonu bulması uzun sren bir yntemdir.

4. BULGULAR

Eđitim veri setindeki sonuçlar Keras yapay sinir ađları sınıflandırması, Scikit-Learn çok katmanlı algılayıcılar sınıflandırma, rastgele orman algoritması, karar ađaçları, XGBoost ve destek vektör makinesinde oldukça benzer sonuçlar ile karşılaşılmıştır. Bu sonuçlar bize verinin anlamlı olduğunu söylemektedir. Her birinden birbirinden farklı deđerler hesaplanırsa idi, verinin anlamsız ve kullanılamaz olduđu sonucuna varılabilirdi. Bu durum işlemin dođru olmadığını ve algoritmaların öğrenmemiş ezberlemiş olduğunu gösterirdi. Bunu gözlemek için işlemin aşırı uyum veya eksik uyum yapıp yapmadığına bakmak gerekir. Ancak k-en yakın komşu algoritması yukarıdaki 6 farklı yöntemle göre oldukça kötü sonuçlar ortaya çıkardığı gözlemlenmiştir.

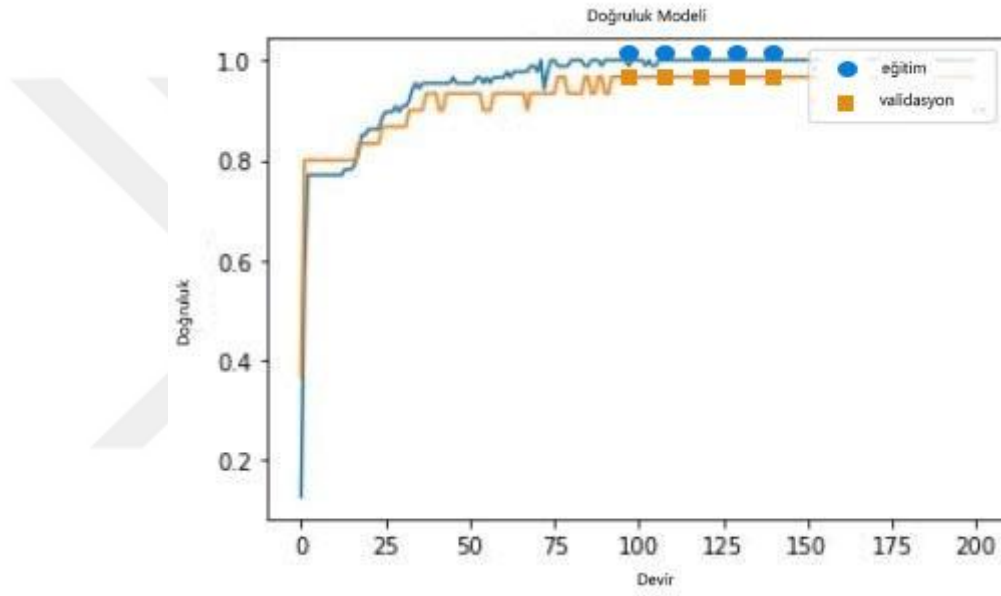
Zor entübasyon sonucuna ulaşmaya çalışılan veri setindeki zor entübe sayısının diđer entübe ve entübe olmayan verilere göre azlığından dolayı test sonuçlarında karışıklığa neden olduğu fark edilmiştir. Entübe ve entübe olmayan hasta verilerinin incelendiđi veri setinde ise algoritmaların test sonuçlarının oldukça başarılı olduğu gözlemlenmiştir.

4.1. KERAS YAPAY SİNİR AĐLARI ÖĐRENME SONUÇLARI

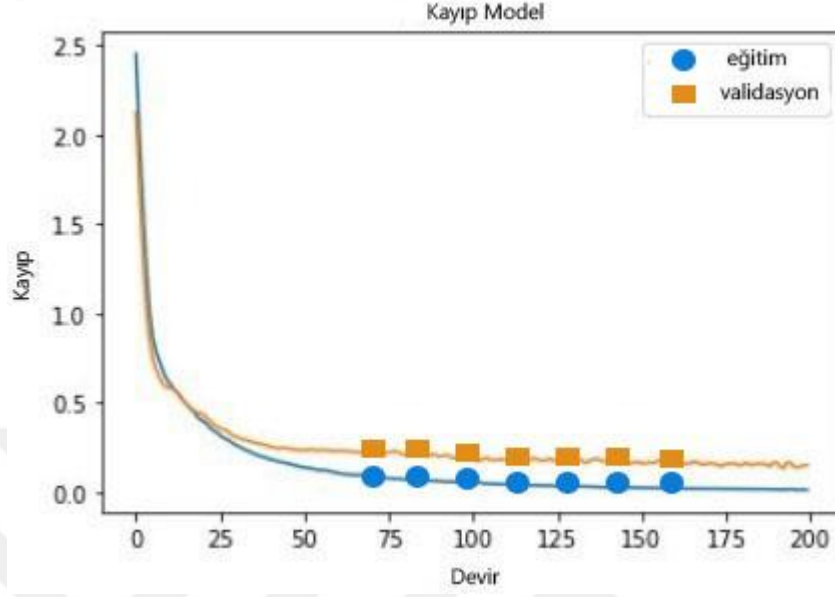
Aşırı uyum göstermesi (AUG) ve eksik uyum göstermesi (EUG) nedir ve neden bu verilere bakılması gereklidir? Bu iki soru, yapay zeka işlemlerinde oldukça önemli bir yere sahiptir. Verinin AUG veya EUG davranışı göstermesi kullanılan algoritmanın başarısız olduğunu gösterir. Makine öğreniminde aşırı uyum göstermesi; bir modelin eğitim verilerinde oldukça iyi performans gösterdiği, ancak iyi genelleme yapılamadığı anlamına gelir. Eğitim hatası düşükse ve test verilerinde hata yüksek ise modelin aşırı uyum gösterdiği anlamına gelir. Eksik uyum göstermesi ise aşırı uyum göstermenin tam tersi olarak karşımıza çıkar. Modelin verilerinin altında yatan yapıyı öğrenemeyecek kadar basit olduğu durumlarda ortaya çıkar. [42]. Modelin aşırı uyum veya eksik uyum göstermesinin olup olmadığını anlamak için grafiksel olarak eğitim veri setinin doğruluđunu validasyon veri seti ile karşılaştırıp grafik haline getirilir. Aynı işlemi eğitim veri seti ile validasyon veri setinin kayıp deđerlerini de karşılaştıran bir grafik çizip aradaki farklarına bakarak karar verilebilir (Şekil 4.1, Şekil 4.2,

Şekil 4.3 ve Şekil 4.4). Eğitim veri setinin aşırı uyum göstermesi daha çok karşılaşılan bir problemdir.

4.1.1.Zor Entübe Veri Setinin Öğrenme Sonuçları



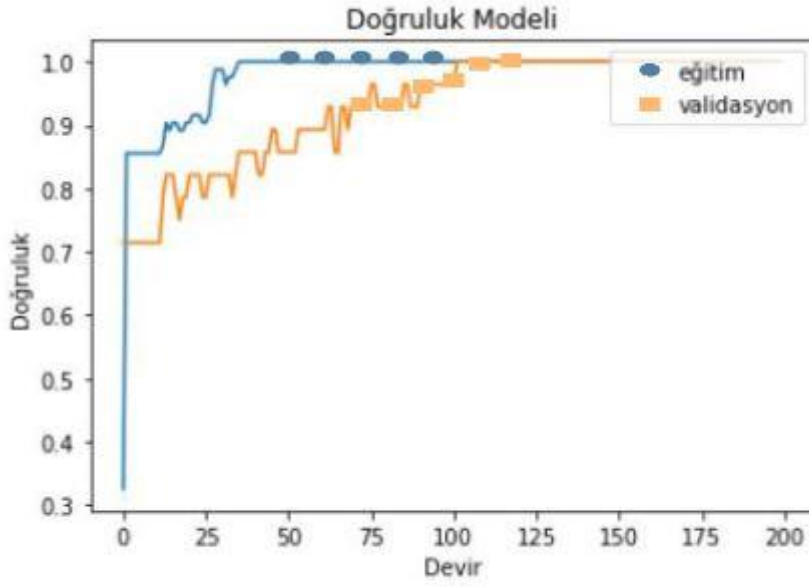
Şekil 4.1. Keras yapay sinir ağları ile oluşturulan zor entübe veri seti modelinin eğitim ve validasyon doğruluk karşılaştırma grafiği.



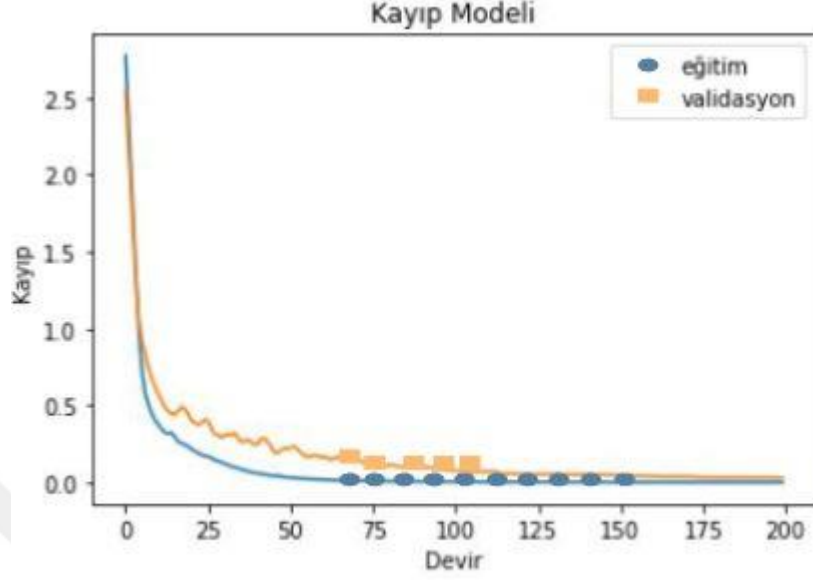
Şekil 4.2. Keras yapay sinir ağları ile oluşturulan zor entübe veri seti modelin eğitim ve validasyon kayıp grafiği.

Şekil 4.1 ve Şekil 4.2 'deki sonuçlara bakıldığında modelin ezberlemediğinin ve öğrendiğinin söylenmesi mümkündür. Buradaki kayıp grafiğine bakıldığında modelin AUG veya EUG göstermediği görülür. Model AUG göstermiş olsa idi bu verinin model doğruluğu ve model kayıp grafiğindeki eğitim ve validasyon göstergelerinin birbirinden ilgisiz bir doğrultuda ilerleyecekti. Mesela model doğruluk grafiği eğitim 0.6 civarlarında iken validasyonun 1.00 göstermesi gibi veya kayıp grafiğinin validasyon çizgisinin eğitim çizgisinden 1.0'larda ayrıştığını ve eşit gitmediğini gösterir. Veri setinde daha çok veri olsa idi bu grafiklerin daha az pik yaptığı gözlenebilirdi.

4.1.2.Entübe Veri Setinin Sonuçları



Şekil 4.3. Keras yapay sinir ağları ile oluşturulan entübe veri seti modelinin eğitim ve validasyon doğruluk karşılaştırma grafiği.



Şekil 4.4. Keras yapay sinir ağları ile oluşturulan entübe veri seti modelin eğitim ve validasyon kayıp grafiği.

Şekil 4.3 ve Şekil 4.4 'deki sonuçlara bakıldığında modelin ezberlemediğinin ve öğrendiğinin söylenmesi mümkündür. Buradaki kayıp grafiğine bakıldığında modelin AUG veya EUG göstermediği görülür. Model AUG göstermiş olsa idi bu verinin model doğruluğu ve model kayıp grafiğindeki eğitim ve validasyon göstergelerinin birbirinden bağımsız bir doğrultuda ilerleyecektir. Ama validasyonun oldukça pik yapan bir grafik çizdiği görülmüştür. Bunu nedeni her ne kadar zor entübe kadar az olmasa da entübe olmayan çıktı verisinin azlığından kaynaklıdır. Veri entübe ve entübe olmayan verilerin yakın oranlarda olması bu problemi çözecektir. Fakat medikal alanda bu kadar çok sayıda veriyi toplamak oldukça güçtür.

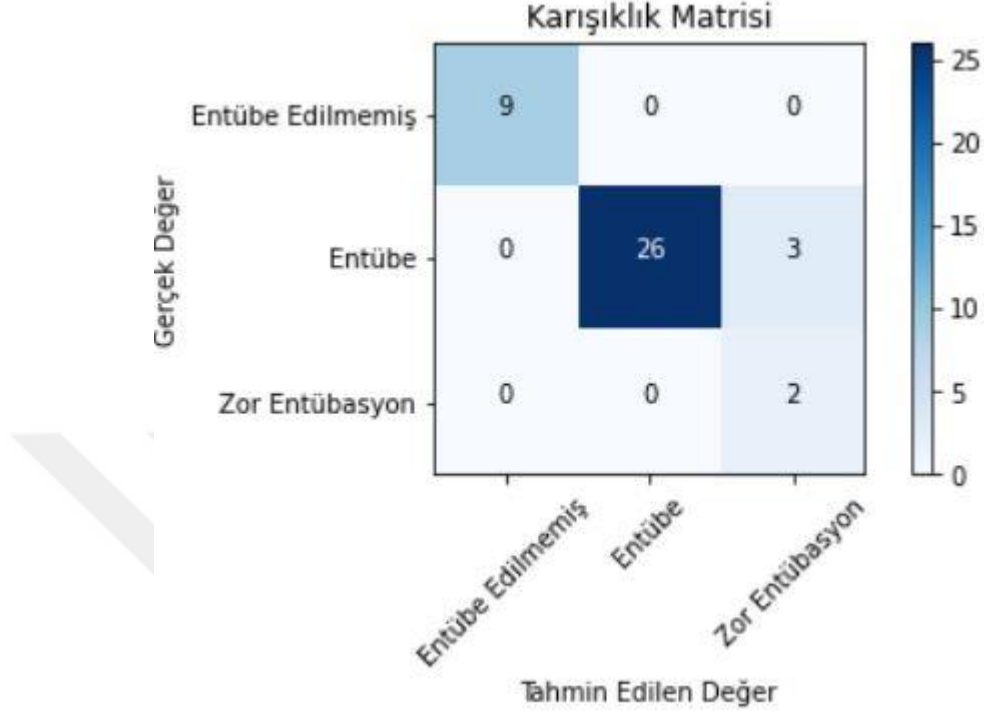
4.2. VERİ SETİNİN EĞİTİM, VALİDASYON VE TEST SONUÇLARI

4.2.1.Zor Entübe Veri Seti Eğitim Validasyon Ve Test Sonuçları

Tablo 4.1. Keras YSA modelinin zor entübe veri setine göre eğitim ve validasyon işlem sonuçları.

KERAS YSA MODELİ YÜZDELİK OLARAK EĞİTİM VE VALİDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 95.87
Validasyon veri seti doğruluk oranı	% 93.1
Eğitim veri seti kayıp oranı	% 0.15
Validasyon veri seti kayıp oranı	% 0.25

Tablo 4.1'deki sonuçları test veri setinin sonuçları ile karşılaştırıldığında oluşturulan modelin başarılı olduğu söylenebilir. Eğitim ve validasyon sonuçlarının aralarında oldukça düşük bir fark görülmektedir. Bu sonuca göre modelin ezberlemediği öğrendiği anlamını çıkartabiliriz.



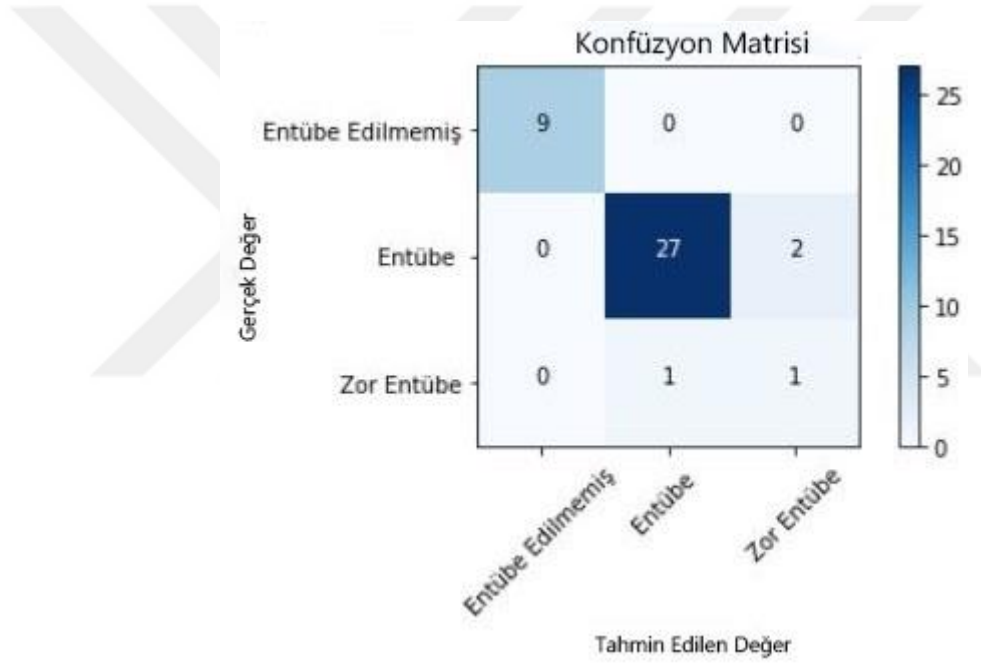
Şekil 4.5. Keras yapay sinir ağları ile oluşturulan modelinin zor entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.

Şekil 4.5’ deki Keras YSA modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça yakın ve başarılı olduğu gözlemlenir. Keras ile yapılan test sonucunda toplam 40 kişiden 3 kişiyi hatalı bulmuştur. Test sonucunun %92.5 oranında doğru bulduğu hesaplanmıştır. Ancak Keras yapay sinir ağları modelinde zor entübe verilerindeki hata payına bakıldığında sonuçlar başarısız olduğu söylenebilir. Buradaki durum zor entübe hastalarının azlığından kaynaklanmaktadır.

Tablo 4.2. Scikit-Learn ÇKAS modelinin zor entübe veri setine göre eğitim ve validasyon işlem sonuçları.

SCİKİT-LEARN ÇKAS MODELİ YÜZDELİK OLARAK EĞİTİM VE VALIDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 98
Validasyon veri seti doğruluk oranı	% 80

Tablo 4.2'deki sonuçları test veri setinin sonuçları ile karşılaştırıldığında oluşturulan modelin başarısı, validasyon kaybı eğitim veri setine göre oldukça fazladır. Bu farkın verideki zor entübe verisinin azlığından kaynaklı olduğu düşünülmektedir.



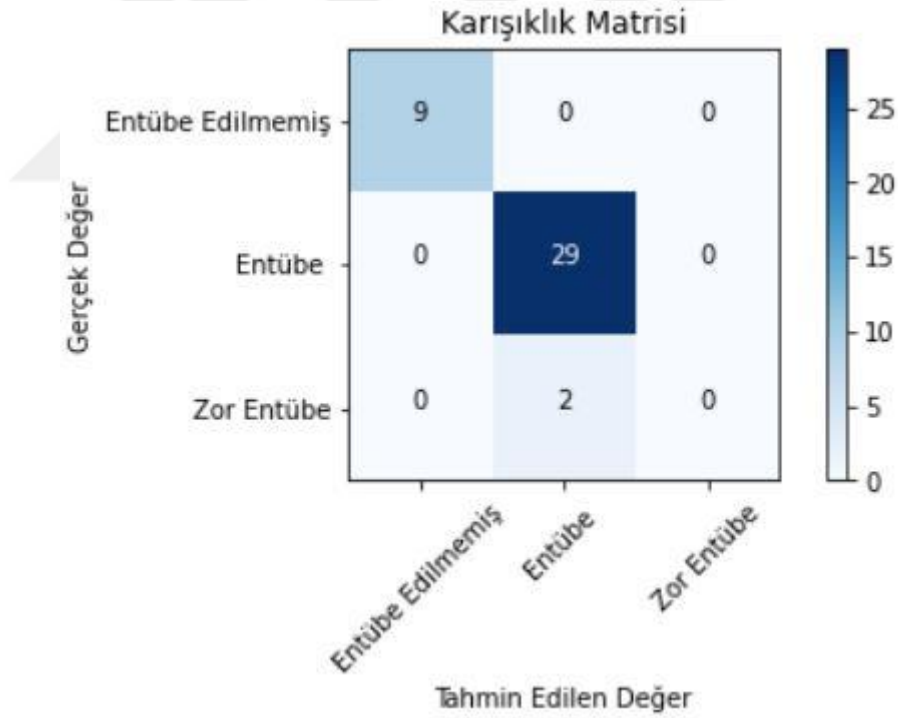
Şekil 4.6. Scikit-Learn ile oluşturulan ÇKAS modelinin zor entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.

Şekil 4.6' deki Scikit-Learn ÇKAS modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça yakın ve başarılı olduğu gözlemlenir. ScikitLearn ÇKAS algoritması ile test sonuçlarında 40 kişiden 3 kişiyi hatalı bulmuştur. Şekil 4.6 test sonucunun %92.5 oranında doğru olduğunu göstermektedir.

Tablo 4.3. Scikit-Learn RA modelinin zor entübe veri setine göre eğitim ve validasyon işlem sonuçları.

SCİKİT-LEARN RA MODELİ YÜZDELİK OLARAK EĞİTİM VE VALİDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 99
Validasyon veri seti doğruluk oranı	% 95

Tablo 4.3'deki sonuçları test veri setinin sonuçları ile karşılaştırıldığında oluşturulan modelin başarısı, validasyon kaybı eğitim veri setine göre oldukça yakındır. Bu, algoritmanın iyi öğrendiğinin göstergesidir.

**Şekil 4.7.** Scikit-Learn ile oluşturulan RA modelinin zor entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.

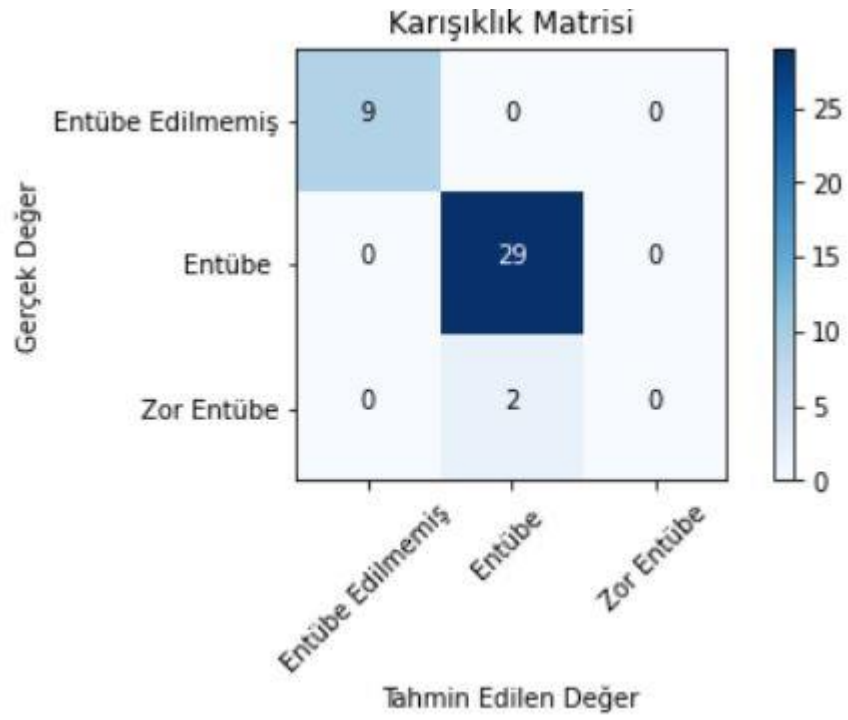
Şekil 4.7' deki Scikit-Learn RA modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça yakın ve başarılı olduğu gözlemlenir. Fakat zor entübe hastalarında oldukça

başarısız bir sonuç ile karşılaşmıştır. Bu sonuç yine veri sayısının azlığı nedeni ile olduğu söylenebilir. Scikit-Learn rastgele ağaç algoritması ile test sonuçlarında 40 kişiden 2 kişiyi hatalı bulmuştur. Bir diğer ifade ile %95 oranında doğru olduğunu göstermektedir.

Tablo 4.4. Scikit-Learn KA modelinin zor entübe veri setine göre eğitim ve validasyon işlem sonuçları.

SCİKİT-LEARN KA MODELİ YÜZDELİK OLARAK EĞİTİM VE VALİDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 99
Validasyon veri seti doğruluk oranı	% 95

Tablo 4.4'deki sonuçları test veri setinin sonuçları ile karşılaştırıldığında oluşturulan modelin başarısı, validasyon kaybı eğitim veri setine göre oldukça yakındır. Burada da algoritmanın iyi öğrendiğini göstermektedir.



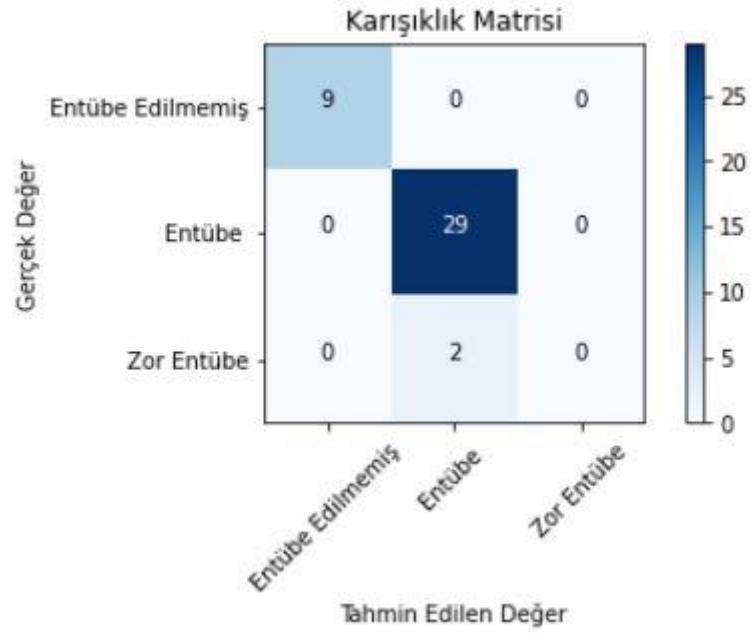
Şekil 4.8. Scikit-Learn ile oluşturulan KA modelinin zor entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.

Şekil 4.8’ deki Scikit-Learn KA modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça yakın ve başarılı olduğu gözlemlenir. Fakat zor entübe hastalarında oldukça başarısız bir sonuç ile karşılaşmıştır. Bu sonuç RA algoritmasındaki gibi yine veri sayısının azlığının nedeni olarak gösterilebilir. Bu iki algoritmanın benzer sonuç vermesi aslında temelde aynı algoritma yapısını kullanmasından kaynaklanmaktadır. Scikit-Learn karar ağaçları algoritması ile test sonuçlarında 40 kişiden 2 kişiyi hatalı bulmuştur. Bir diğer ifade sonucunun %95 oranında doğru olduğunu göstermektedir.

Tablo 4.5. Scikit-Learn XGBoost modelinin zor entübe veri setine göre eğitim ve validasyon işlem sonuçları.

SCİKİT-LEARN XGBOOST MODELİ YÜZDELİK OLARAK EĞİTİM VE VALIDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 100
Validasyon veri seti doğruluk oranı	% 95

Tablo 4.5’deki sonuçları test veri setinin sonuçları ile karşılaştırıldığında oluşturulan modelin başarısı, validasyon kaybı eğitim veri setine göre oldukça yakındır. Bu sonuçlar algoritmanın iyi öğrendiğini göstermektedir.



Şekil 4.9. Scikit-Learn ile oluşturulan XGBoost modelinin zor entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.

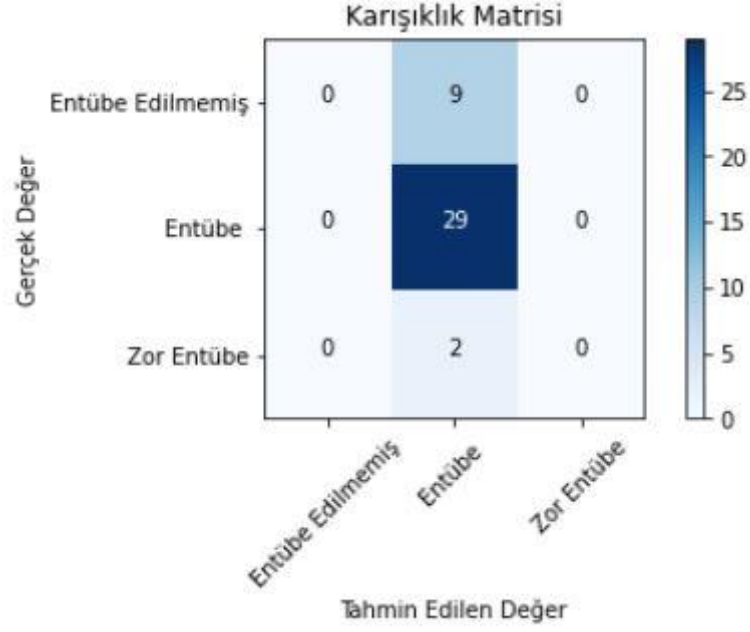
Şekil 4.9’ daki Scikit-Learn XGBoost modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça yakın ve başarılı olduğu gözlemlenir. Fakat zor entübe

hastalarında oldukça başarısız bir sonuç ile karşılaşmıştır. Bu sonuç RA ve KA algoritmasındaki gibi veri sayısının azlığının nedeni olarak gösterilebilir. Bu algoritmaların benzer sonuç vermesi aslında temelde aynı algoritma yapısını kullanmasından kaynaklanmaktadır. Scikit-Learn XGBoost algoritması ile test sonuçlarında 40 kişiden 2 kişiyi hatalı bulmuştur. Bir diğer ifade ile sonucunun %95 oranında doğru olduğunu göstermektedir.

Tablo 4.6. Scikit-Learn KEK modelinin zor entübe veri setine göre eğitim ve validasyon işlem sonuçları.

SCİKİT-LEARN KEK MODELİ YÜZDELİK OLARAK EĞİTİM VE VALİDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 77
Validasyon veri seti doğruluk oranı	% 70

Tablo 4.6'deki sonuçları test veri setinin sonuçları ile karşılaştırıldığında oluşturulan modelin başarısı, validasyon kaybı eğitim veri setine göre oldukça yakındır. Bu sonuçlara bakıldığında k-en yakın komşunun zor entübe veri setinde başarısız olduğu gözlemlenmiştir.



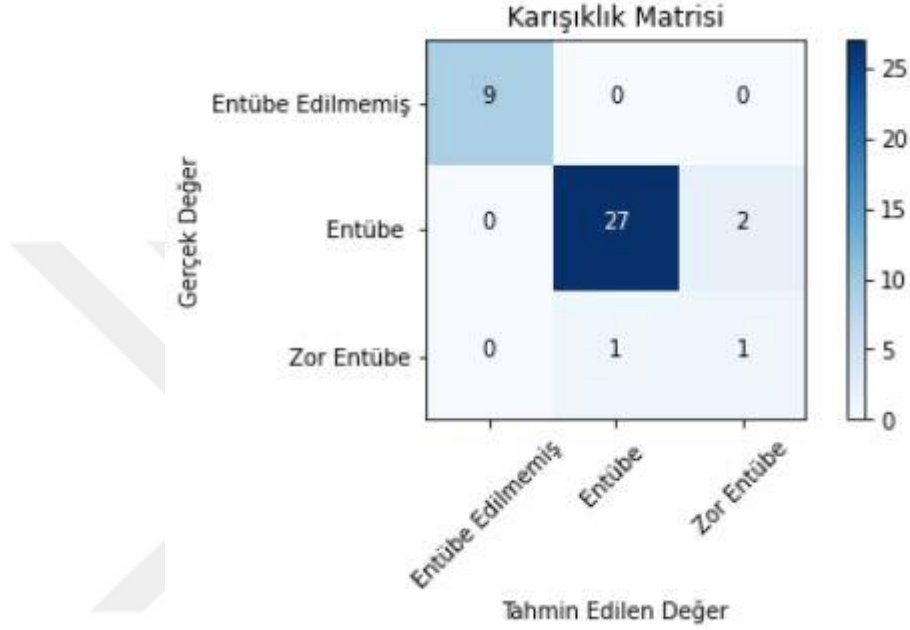
Şekil 4.10. Scikit-Learn ile oluşturulan KEK modelinin zor entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.

Şekil 4.10' daki Scikit-Learn KEK modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça uzak ve başarısız olduğu gözlemlenir. Scikit-Learn karar ağaçları algoritması ile test sonuçlarında 40 kişiden 11 kişiyi hatalı bulmuştur. Bir diğer ifade ile sonucunun %72.5 oranında doğru olduğunu göstermektedir.

Tablo 4.7. Scikit-Learn DVM modelinin zor entübe veri setine göre eğitim ve validasyon işlem sonuçları.

SCİKİT-LEARN DVM MODELİ YÜZDELİK OLARAK EĞİTİM VE VALİDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 99
Validasyon veri seti doğruluk oranı	% 95

Tablo 4.7'deki sonuçları test veri setinin sonuçları ile karşılaştırıldığında oluşturulan modelin başarısı, validasyon kaybı eğitim veri setine göre oldukça yakındır. Buda algoritmanın iyi öğrendiğini göstermektedir.

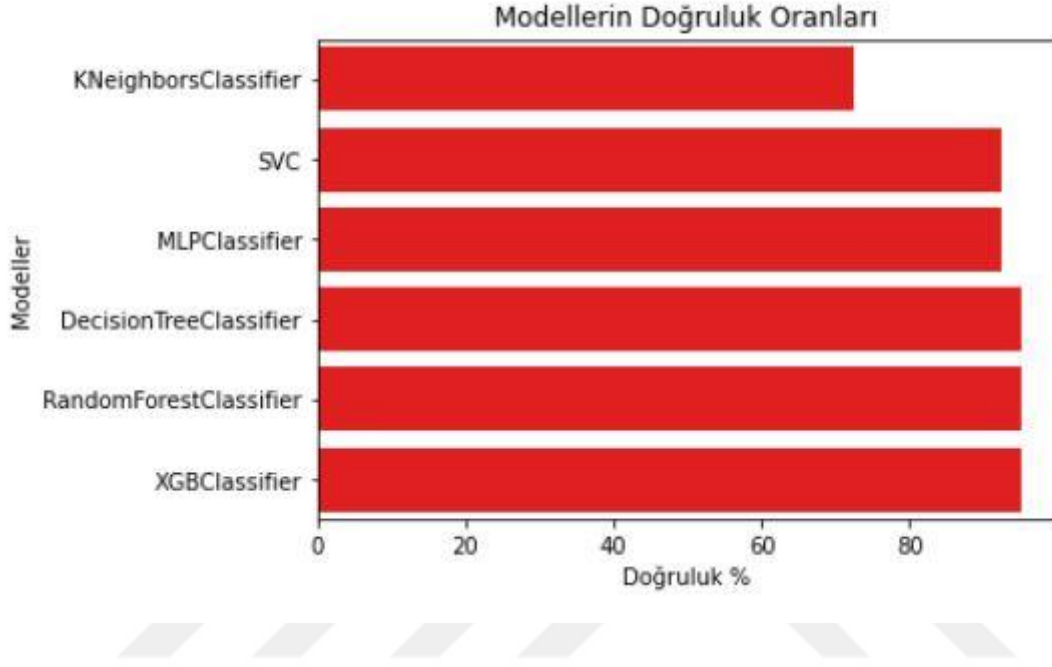


Şekil 4.11. Scikit-Learn ile oluşturulan DVM modelinin zor entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.

Şekil 4.11' deki Scikit-Learn DVM modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça yakın ve başarılı olduğu gözlemlenir. Fakat zor entübe hastalarında yapay sinir ağlarındaki modellere benzer bir sonuç ile karşılaşmıştır. Scikit-Learn destek vektör makinesi algoritması ile test sonuçlarında 40 kişiden 2 kişiyi hatalı bulmuştur. Şekil 4.11'de sonucun %92.5 oranında doğru olduğunu göstermektedir.

Keras ve ScikitLearn'de kullanılan verilerin birbirinden farklı olmasının sebebi bu verilerin rastgele olarak sistem tarafından her seferinde değiştirilmesinden ve farklı validasyon yöntemlerinin kullanılmasından kaynaklıdır. Bu yapıdan da gördüğümüz gibi hem Keras hem de Scikit-Learn'de ki modellerin birbirinden farklı olmalarına rağmen modelin doğru öğrenip sonuçları oldukça anlamlı çıkarmıştır. Ancak zor entübe anlamında bakıldığında verinin başarılı

bir sonuç vermediğini görmek mümkündür. Scikit-Learn zor entübe veri setinin test sonuçlarının tüm algoritalarda ki modellerin sonuçları Şekil 4.12’de gösterilmiştir.



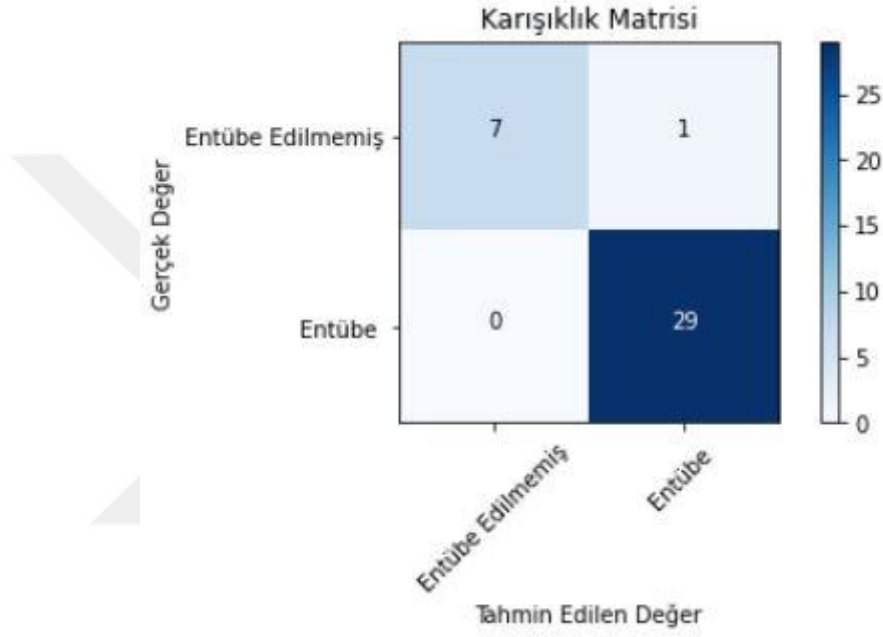
Şekil 4.12. Scikit-Learn zor entübe verisine göre modellerin doğruluk oran grafiği.

4.2.2. Entübe Veri Setinin Eğitim, Validasyon Ve Test Sonuçları

Tablo 4.8. Keras YSA modelinin entübe veri setine göre eğitim ve validasyon işlem sonuçları.

KERAS YSA MODELİ YÜZDELİK OLARAK EĞİTİM VE VALİDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 98
Validasyon veri seti doğruluk oranı	% 93
Eğitim veri seti kayıp oranı	%0.09
Validasyon veri seti kayıp oranı	%0.18

Tablo 4.8'deki sonuçları test veri setinin sonuçları ile karşılaştırıldığında oluşturulan modelin başarılı olduğu söylenebilir. Eğitim ve validasyon sonuçlarının aralarında oldukça düşük bir fark görülmektedir. Bu sonuca göre modelin ezberlemediği öğrendiği anlamı çıkartılabilir.



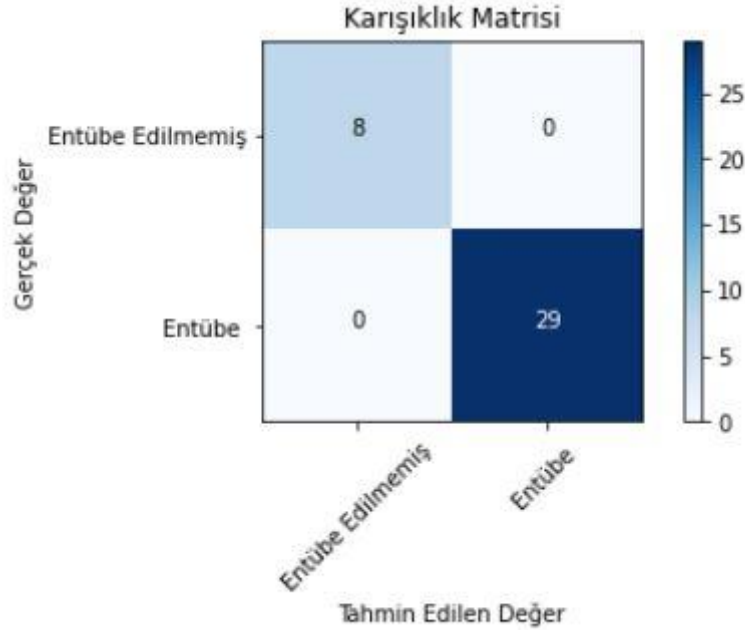
Şekil 4.13. Keras yapay sinir ağı ile oluşturulan modelinin entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.

Şekil 4.13' deki Keras YSA modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça yakın ve başarılı olduğu gözlemlenir. Keras ile yapılan test sonucunda toplam 37 kişiden 1 kişiyi hatalı bulmuştur. Test sonucunun %97.3 oranında doğru bulduğu hesaplanmıştır.

Tablo 4.9. Scikit-Learn ÇKAS modelinin entübe veri setine göre eğitim ve validasyon işlem sonuçları.

SCİKİT-LEARN ÇKAS MODELİ YÜZDELİK OLARAK EĞİTİM VE VALİDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 100
Validasyon veri seti doğruluk oranı	% 97.5

Tablo 4.9'daki sonuçları test veri setinin sonuçları ile karşılaştırıldığında oluşturulan modelin başarılı olduğu söylenebilir. Eğitim ve validasyon sonuçlarının aralarında oldukça düşük bir fark görülmektedir. Bu sonuca göre modelin ezberlemediği öğrendiği anlamı çıkartılabilir.

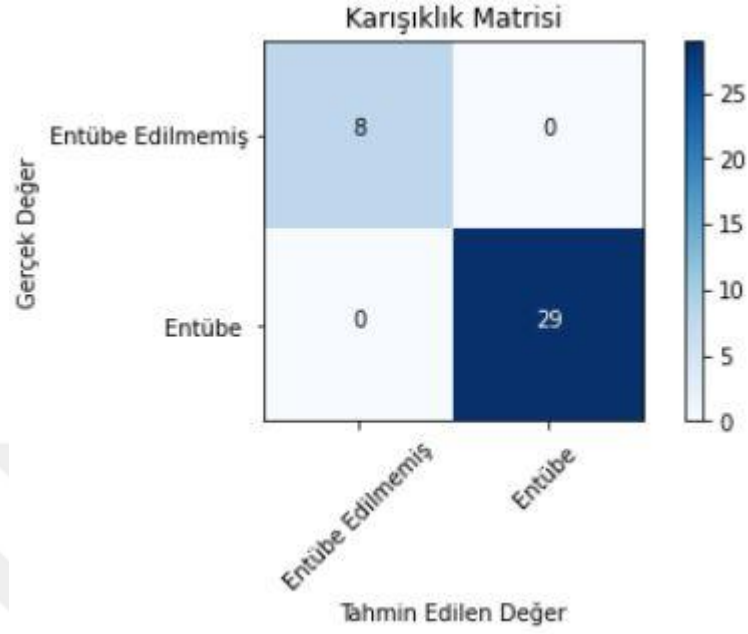
**Şekil 4.14.** Scikit-Learn ile oluşturulan ÇKAS modelinin entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.

Şekil 4.14’ deki Scikit-Learn ÇKAS modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça yakın ve başarılı olduğu gözlemlenir. Scikit-Learn ile yapılan test sonucunda toplam 37 kişiden 0 kişiyi hatalı bulmuştur. Test sonucunun %100 oranında doğru bulduğu hesaplanmıştır.

Tablo 4.10. Scikit-Learn RA modelinin entübe veri setine göre eğitim ve validasyon işlem sonuçları.

SCİKİT-LEARN RA MODELİ YÜZDELİK OLARAK EĞİTİM VE VALİDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 100
Validasyon veri seti doğruluk oranı	% 100

Tablo 4.10’deki sonuçları test veri setinin sonuçları ile karşılaştırıldığında oluşturulan modelin başarılı olduğu söylenebilir. Eğitim ve validasyon sonuçlarının aralarında oldukça düşük bir fark görülmektedir. Bu sonuca göre modelin ezberlemediği öğrendiği anlamı çıkartılabilir.



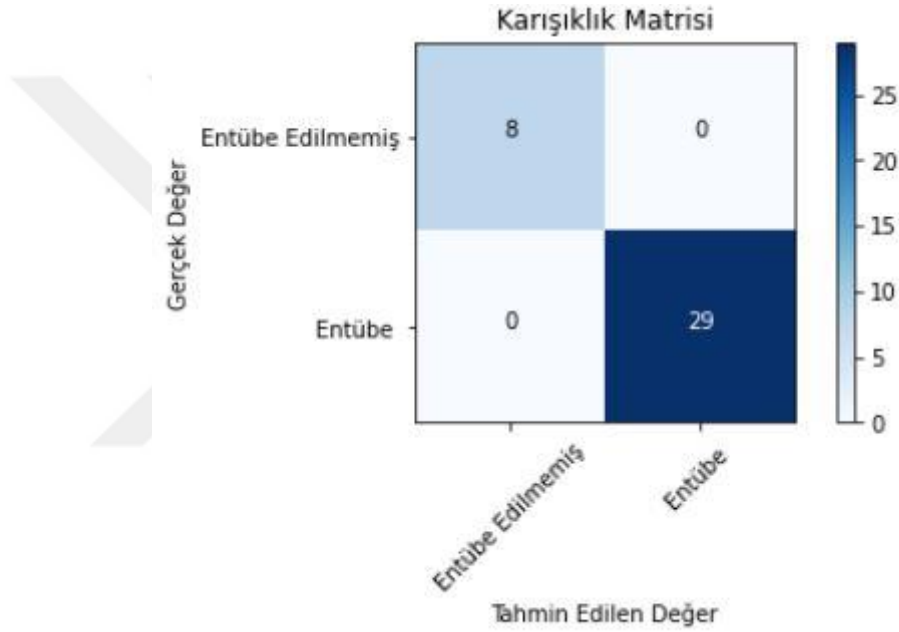
Şekil 4.15. Scikit-Learn ile oluşturulan RA modelinin entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.

Şekil 4.15’ deki Scikit-Learn RA modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça yakın ve başarılı olduğu gözlemlenir. Scikit-Learn ile yapılan test sonucunda toplam 37 kişiden 0 kişiyi hatalı bulmuştur. Test sonucunun %100 oranında doğru bulunduğu hesaplanmıştır.

Tablo 4.11. Scikit-Learn KA modelinin entübe veri setine göre eğitim ve validasyon işlem sonuçları.

SCİKİT-LEARN KA MODELİ YÜZDELİK OLARAK EĞİTİM VE VALİDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 100
Validasyon veri seti doğruluk oranı	% 97.5

Tablo 4.11'deki sonuçları test veri setinin sonuçları ile karşılaştırıldığında oluşturulan modelin başarılı olduğu söylenebilir. Eğitim ve validasyon sonuçlarının aralarında oldukça düşük bir fark görülmektedir. Bu sonuca göre modelin ezberlemediği öğrendiği anlamı çıkartılabilir.



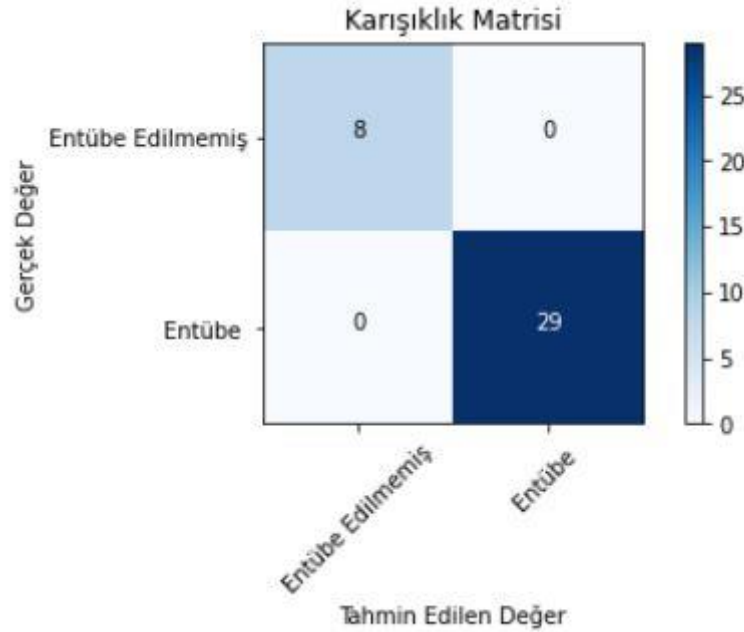
Şekil 4.16. Scikit-Learn ile oluşturulan KA modelinin entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.

Şekil 4.16' deki Scikit-Learn KA modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça yakın ve başarılı olduğu gözlemlenir. Scikit-Learn ile yapılan test sonucunda toplam 37 kişiden 0 kişiyi hatalı bulmuştur. Test sonucunun %100 oranında doğru bulduğu hesaplanmıştır.

Tablo 4.12. Scikit-Learn XGBoost modelinin entübe veri setine göre eğitim ve validasyon işlem sonuçları.

SCİKİT-LEARN XGBOOST MODELİ YÜZDELİK OLARAK EĞİTİM VE VALIDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 100
Validasyon veri seti doğruluk oranı	% 97.5

Tablo 4.12'deki sonuçları test veri setinin sonuçları ile karşılaştırıldığında oluşturulan modelin başarılı olduğu söylenebilir. Eğitim ve validasyon sonuçlarının aralarında oldukça düşük bir fark görülmektedir. Bu sonuca göre modelin ezberlemediği öğrendiği anlamı çıkartılabilir.



Şekil 4.17. Scikit-Learn ile oluşturulan XGBoost modelinin entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.

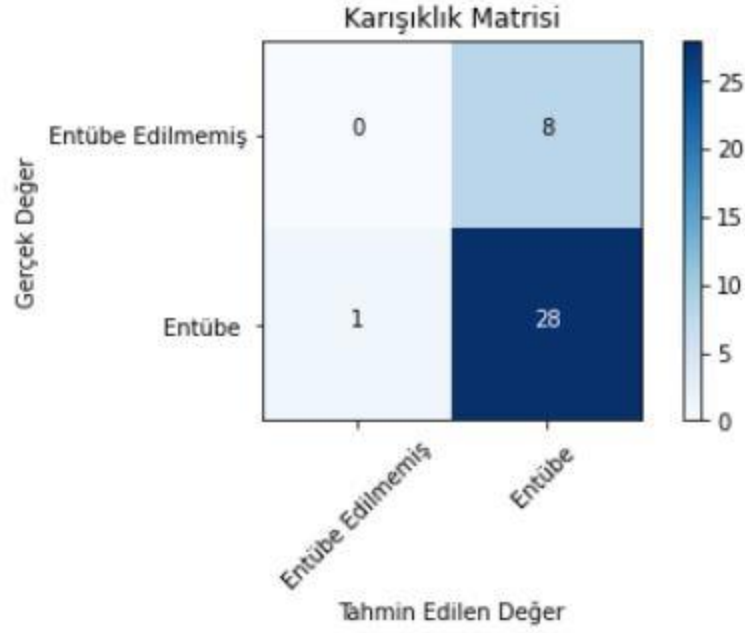
Şekil 4.17' deki Scikit-Learn XGBoost modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça yakın ve başarılı olduğu gözlemlenir. Scikit-Learn ile yapılan

test sonucunda toplam 37 kişiden 0 kişiyi hatalı bulmuştur. Test sonucunun %100 oranında doğru bulduğu hesaplanmıştır.

Tablo 4.13. Scikit-Learn KEK modelinin entübe veri setine göre eğitim ve validasyon işlem sonuçları.

SCİKİT-LEARN KEK MODELİ YÜZDELİK OLARAK EĞİTİM VE VALİDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 100
Validasyon veri seti doğruluk oranı	% 74.1

Tablo 4.13'deki sonuçları test veri setinin sonuçları ile karşılaştırıldığında oluşturulan modelin başarısız olduğu görülmüştür. Eğitim ve validasyon sonuçlarının aralarında oldukça yüksek bir fark görülmektedir. Bu sonuca göre modelin öğrenemediğini anlamı çıkartılabilir. Eğitim veri seti doğruluk oranına bakılırsa modelin ezberlediği görülmüştür.



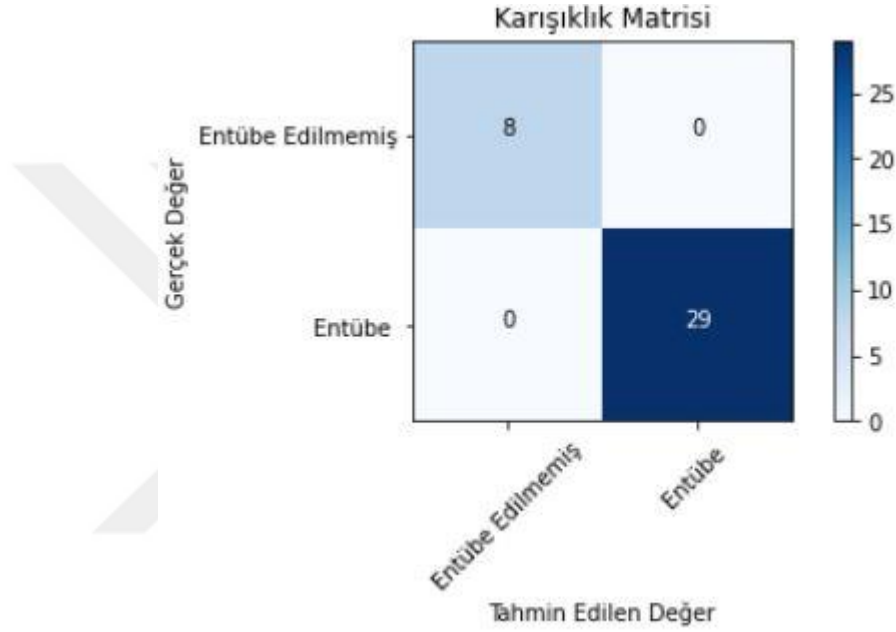
Şekil 4.18. Scikit-Learn ile oluşturulan KEK modelinin entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.

Şekil 4.18’deki Scikit-Learn KEK modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça uzak ve başarısız olduğu gözlemlenir. Scikit-Learn ile yapılan test sonucunda toplam 37 kişiden 9 kişiyi hatalı bulmuştur. Test sonucunun %75.6 oranında doğru bulunduğu hesaplanmıştır. Bu oran eğitim veri setindeki doğruluk oranı ile karşılaştırıldığında öğrenmek yerine ezberleyen bir modelin sistemi ne kadar yanıltacağı gösterilmiştir.

Tablo 4.14. Scikit-Learn DVM modelinin entübe veri setine göre eğitim ve validasyon işlem sonuçları.

SCİKİT-LEARN DVM MODELİ YÜZDELİK OLARAK EĞİTİM VE VALİDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 100
Validasyon veri seti doğruluk oranı	% 100

Tablo 4.14'deki sonuçları test veri setinin sonuçları ile karşılaştırdığımızda oluşturulan modelin başarılı olduğu söylenebilir. Eğitim ve validasyon sonuçlarının aralarında oldukça düşük bir fark görülmektedir. Bu sonuca göre modelin ezberlemediği öğrendiği anlamı çıkartılabilir.

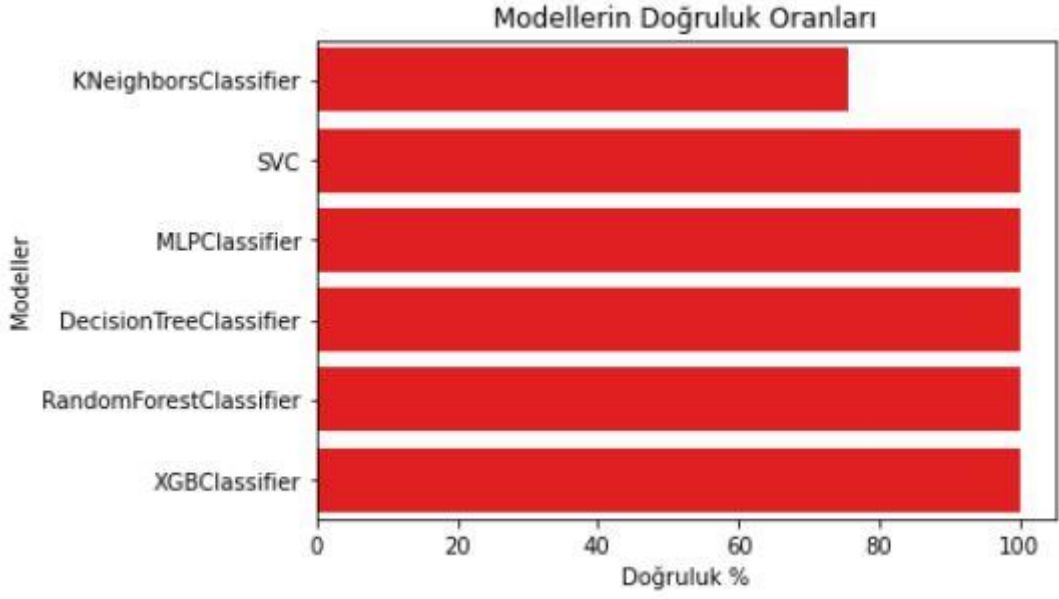


Şekil 4.19. Scikit-Learn ile oluşturulan DVM modelinin entübe veri setine göre test sonuçlarının karışıklık matrisi sonuçları.

Şekil 4.19' daki Scikit-Learn DVM modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça yakın ve başarılı olduğu gözlemlenir. Scikit-Learn ile yapılan test sonucunda toplam 37 kişiden 0 kişiyi hatalı bulmuştur. Test sonucunun %100 oranında doğru bulduğu hesaplanmıştır.

Keras ve ScikitLearn'de kullanılan verilerin birbirinden farklı olmasının nedeni bu verilerin rastgele olarak sistem tarafından her seferinde değiştirilmesinden ve farklı validasyon yöntemlerinin kullanılmasından kaynaklıdır. Bu yapıdan da gördüğümüz gibi hem Keras hem de Scikit-Learn'de ki modellerin birbirinden farklı olmalarına rağmen modelin doğru öğrenip sonuçları oldukça anlamlı çıkarmıştır. Entübe veri setinin diğer zor entübe veri setine göre daha

anlamalı olduğunu kullanılan yapay zeka algoritma sonuçlarına bakarak söyleyebiliriz. Oluşturulan veri setindeki sonuçların birbirine yüzde olarak orantısız olmaması gerekir. Bu orantısızlık veri setinde dengesizliği artırır ve bulunan sonuçların doğruluk oranının düşmesine sebep olur. Scikit-Learn entübe veri setinin test sonuçlarının tüm algoritmalarda ki modellerin sonuçları Şekil 4.20’de gösterilmiştir.



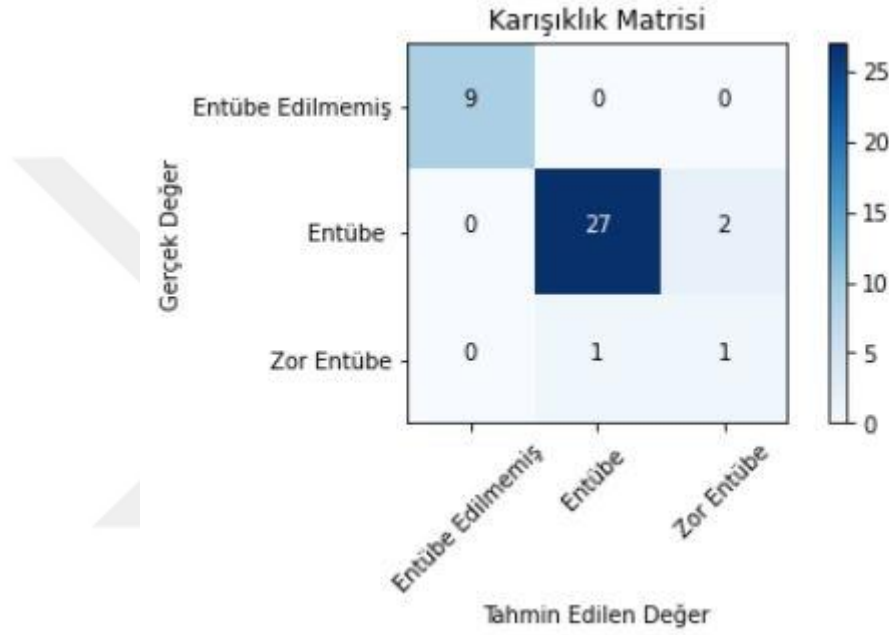
Şekil 4.20. Scikit-Learn entübe verisine göre modellerin doğruluk oran grafiği.

4.2.3. Birini Dışarıda Bırakma Çapraz Validasyon Yöntemi İle ÇKAS Modeli

Tablo 4.15. Scikit-Learn ÇKAS modelinin zor entübe veri setine göre eğitim ve birini dışarıda bırakma validasyon işlem sonuçları.

SCİKİT-LEARN ÇKAS MODELİ YÜZDELİK OLARAK EĞİTİM VE VALİDASYON SONUÇLARI	
Eğitim veri seti doğruluk oranı	% 77
Validasyon veri seti doğruluk oranı	% 62.5

Tablo 4.15'deki sonuçları test veri setinin sonuçları ile karşılaştırdığımızda oluşturulan modelin başarısız olduğu görülmüştür. Eğitim ve validasyon sonuçlarının aralarında oldukça düşük bir fark görülmektedir.

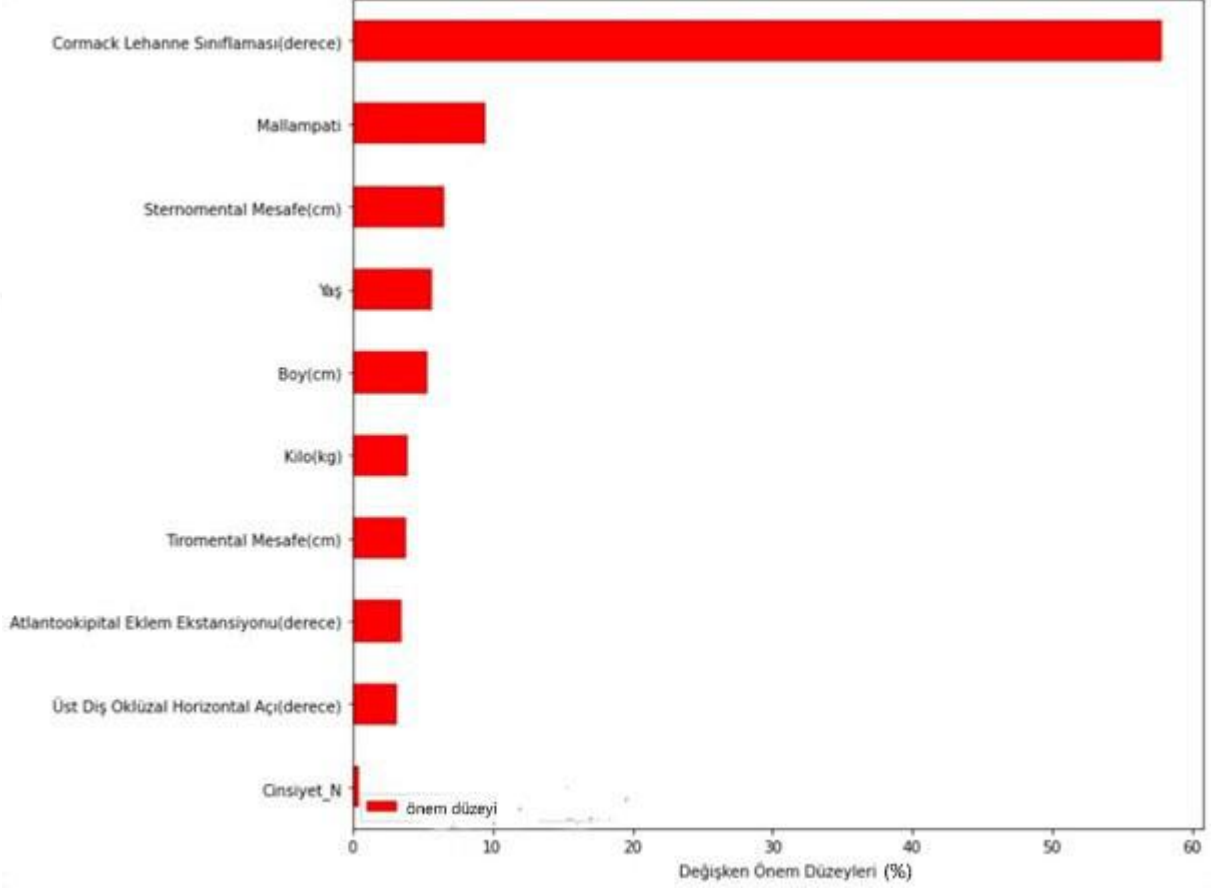


Şekil 4.21. Scikit-Learn ile oluşturulan ÇKAS modelinin zor entübe veri setinin birini dışarıda bırakma çapraz validasyon yöntemine göre test sonuçlarının karışıklık matrisi sonuçları.

Şekil 4.19' daki Scikit-Learn ÇKAS modelinin test sonuçlarının karışıklık matrisine bakıldığında sonuçların oldukça yakın ve başarılı olduğu gözlemlenir. Scikit-Learn ile yapılan test sonucunda toplam 40 kişiden 3 kişiyi hatalı bulmuştur. Test sonucunun %92.5 oranında doğru bulduğu hesaplanmıştır.

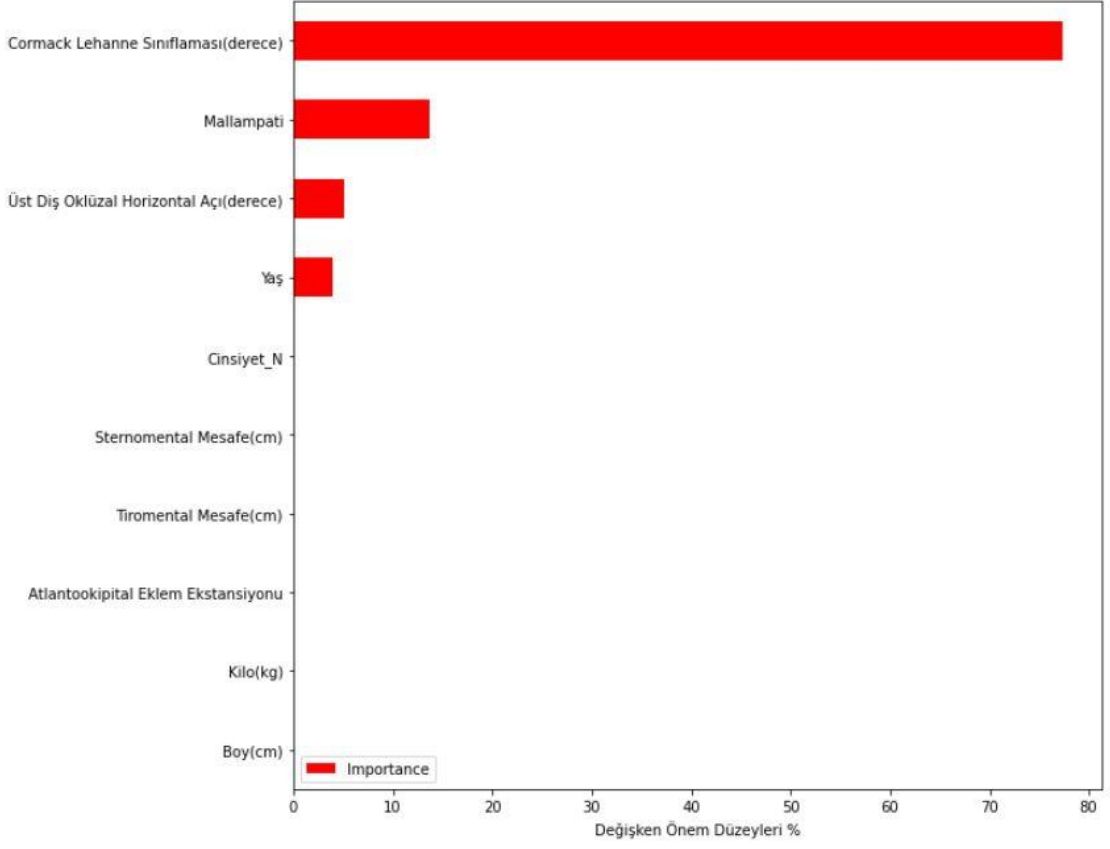
Birini dışarıda bırakma (BDB) çapraz validasyonun zor entübe veri setinde algoritmanın oldukça karıştığı görülmüştür. Test sonuçları her ne kadar bir önceki validasyon yöntemleri ile aynı çıkmış olsa da eğitim ve validasyon doğruluk oranları oldukça düşüktür. Bu yöntemin denenmesindeki amaç bir önceki validasyon yöntemlerinin zor entübe veri setinden zor entübasyon hastalarını bulmada başarısız olmasındandır.

4.3. KARAR AĞAÇLARI TABANLI ALGORİTMALARDA KULLANILAN PARAMETRELERİN ÖNEM DÜZEYLERİ



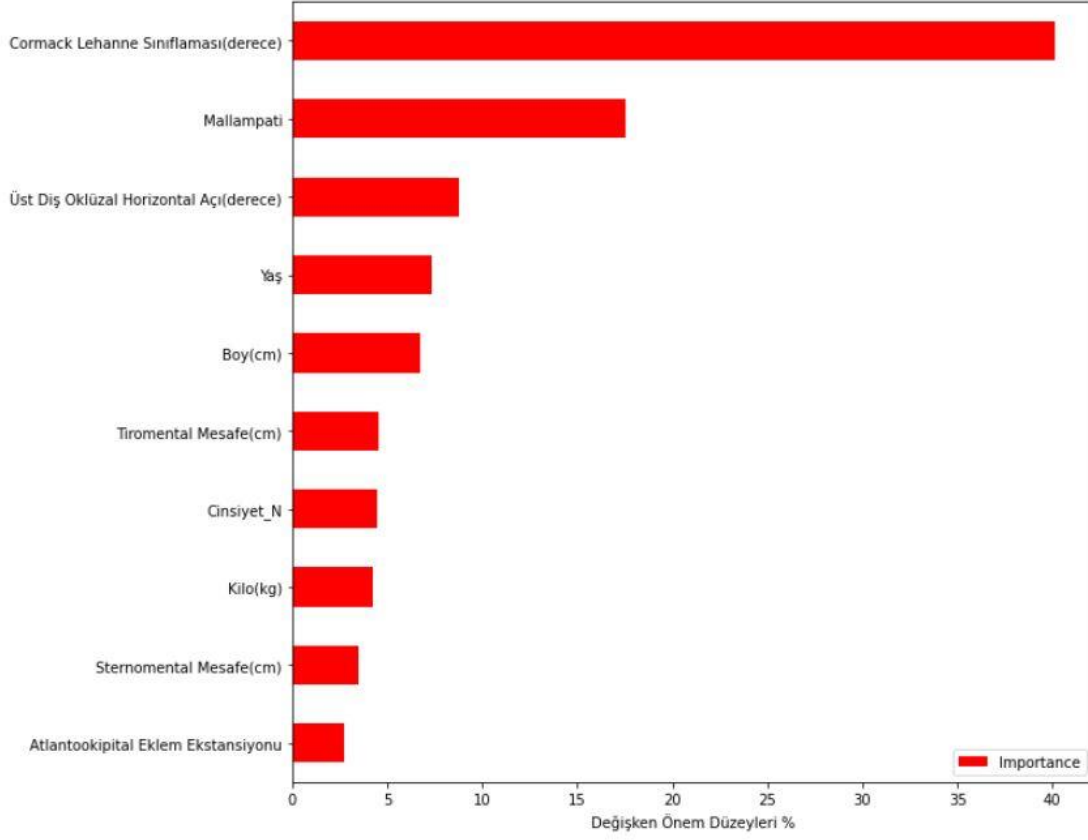
Şekil 4.22. Oluşturulan Scikit-Learn rastgele orman modelinde verideki değişkenleri kullanmadaki önem düzeyi.

Şekil 4.22 'de bulunan sonuçlar bize rastgele orman modelinin hangi değişkene göre daha çok sonuçları ayırttığını göstermektedir. Bu grafik normalde beklenen sonuçları vermiştir. Normalde anestezi uzmanları da Cormack ve Lehanne sınıflandırmasını entübe edilecek hastada en çok bakılan parametre olarak gösterilir.



Şekil 4.23. Oluşturulan Scikit-Learn karar ağaçları modelinde verideki değişkenleri kullanmadaki önem düzeyi.

Şekil 4.23 'de bulunan sonuçlar bize karar ağaçlarının modelinin hangi değişkene göre daha çok sonuçları ayırttığını göstermektedir. Bu grafikte rastgele ormandaki gibi beklenen sonuçları verip en çok Cormack- Lehanne sınıflandırmasına önem vermiştir.



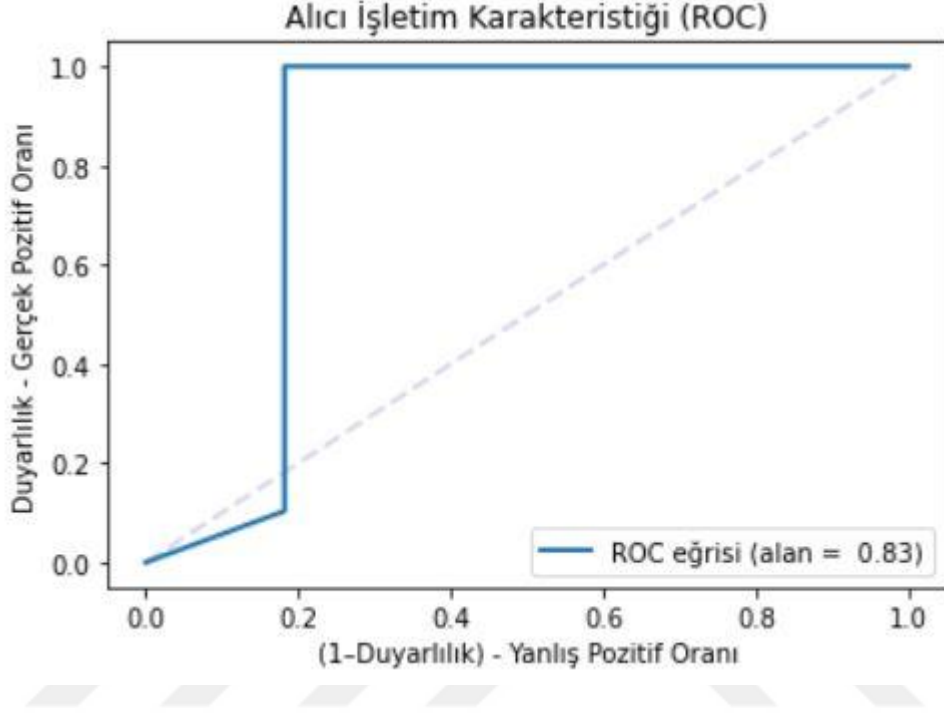
Şekil 4.24. Oluşturulan Scikit-Learn XGBoost modelinde verideki değişkenleri kullanmadaki önem düzeyi.

Şekil 4.24 'de bulunan sonuçlar bize XGBoost modelinin hangi değişkene göre daha çok sonuçları ayırttığını göstermektedir. Bu grafikte diğer iki algoritmada olduğu gibi Cormack-Lehanne sınıflamasını önem verdiği gözlemlenmiştir.

4.4. YAPAY ZEKA ALGORİTMALARIN AİK-EAA GRAFİKLERİ

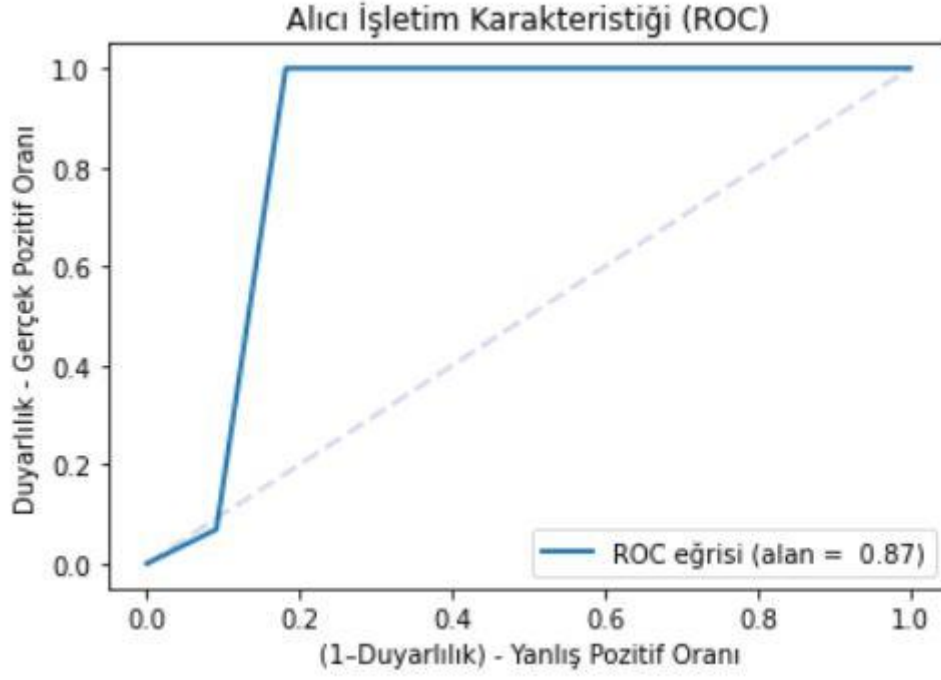
Sınıflandırma probleminin performans ölçümü Eğrinin Altındaki Alan (EAA, AUC) ve Alıcı İşletim Karakteristiği (AİK, ROC) eğrisine bakılmaktadır. AİK bir olasılık eğrisidir ve EAA modelin sınıflarının ne kadar ayırt edileceğini göstermektedir. EAA ne kadar yüksek ise tahmini de o kadar iyidir. Bu çalışma bağlamında EAA'nin yüksek çıkması modelin entübe olan veya entübe olmayan hastaları ayırt etmekte o kadar iyi olduğu göstermektedir.

4.4.1. Zor Entübe Veri Setinin ROC Eğrisi



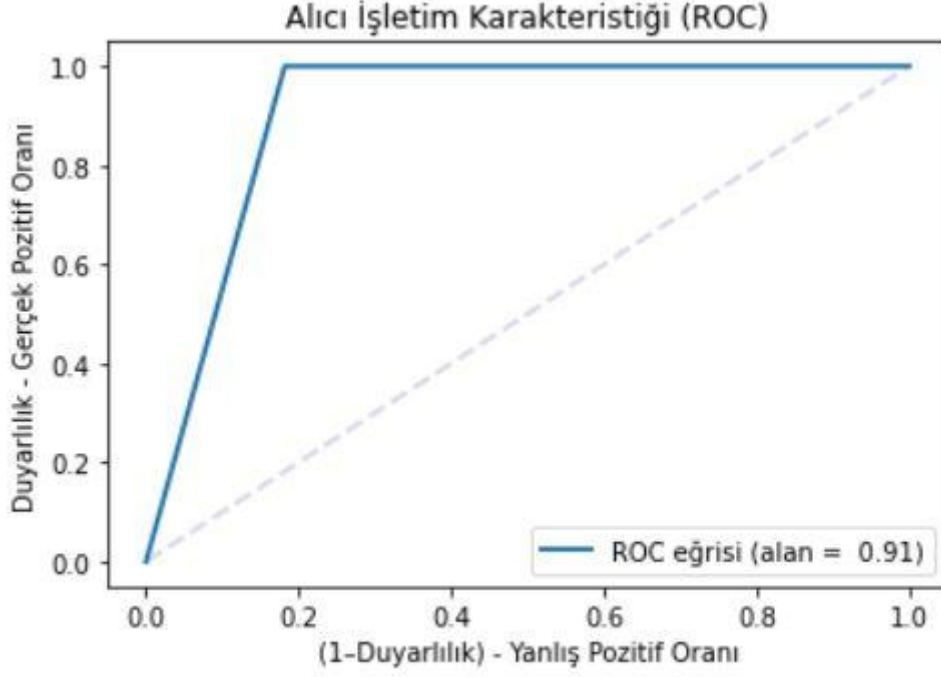
Şekil 4.25. Keras yapay sinir ağı sınıflandırma modelinin zor entübe veri setinin sonuçlarının AİK eğrisi.

Şekil 4.25'te görüldüğü gibi verinin dengesizliğinden kaynaklı AİK eğrisinde bir düşüş yaşayıp daha sonra düzelmiştir. Dengesiz dağılım olan veri setlerinde bu sonuç ile karşılaşmak oldukça normaldir.



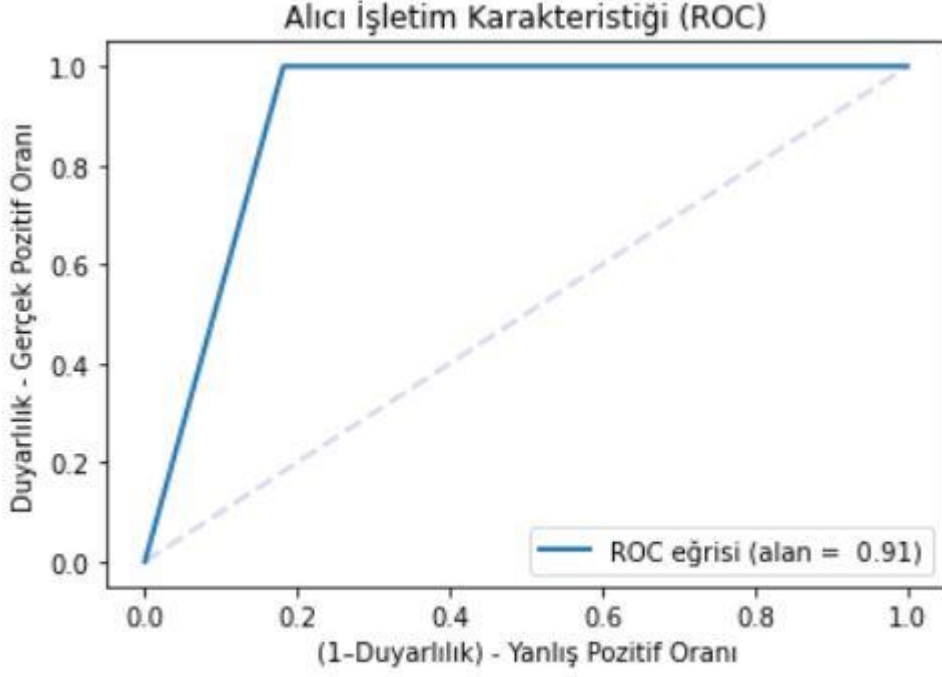
Şekil 4.26. Scikit-Learn çok katmanlı algılayıcılar sınıflandırma modelinin zor entübe veri setinin sonuçlarının AİK eğrisi.

Şekil 4.26’da görüldüğü gibi verinin dengesizliğinden kaynaklı AİK eğrisinde bir düşüş yaşayıp daha sonra düzelmiştir. Keras yapay sinir ağlarında olduğu gibi aynı şekilde bir durum gözlemlenmiştir.



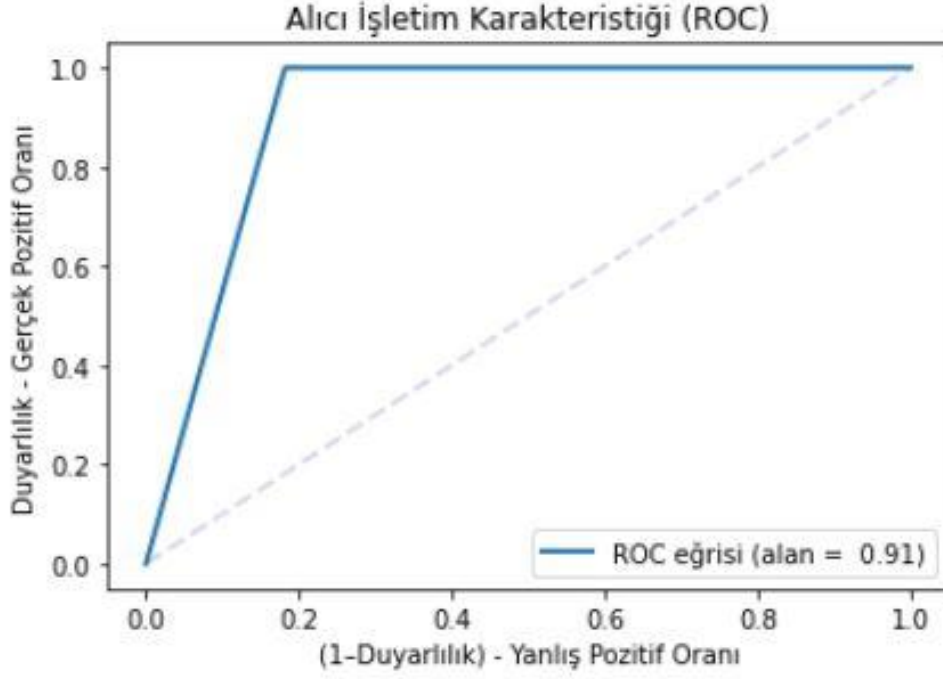
Şekil 4.27. Scikit-Learn rastgele orman sınıflandırma modelinin zor entübe veri setinin sonuçlarının AİK eğrisi.

Şekil 4.27’de görüldüğü gibi verinin dengesizliğinden kaynaklı AİK eğrisinde bir düşüş yaşamadığı görülmüştür. Rastgele orman yapısının bir karar ağaç tabanlı algoritması olmasından kaynaklı etiketleme ile çalışmaktadır. Etiket mantığında düzensiz veriler elenip veri setinin dışındaymış gibi görünür. Bu neden AİK Eğrisi beklenen bir seyir izlemiştir.



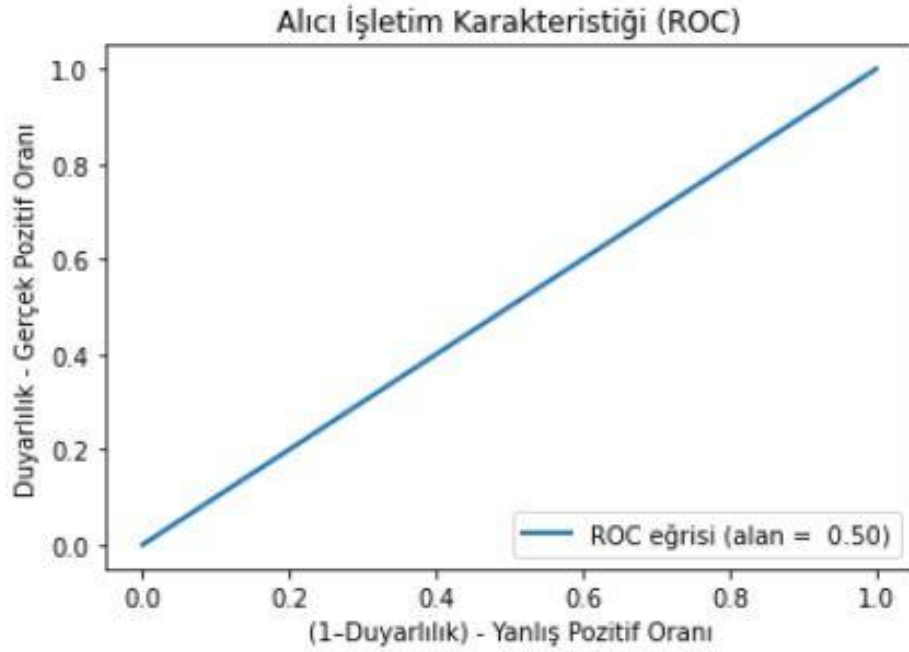
Şekil 4.28. Scikit-Learn karar ağaçları sınıflandırma modelinin zor entübe veri setinin sonuçlarının AİK eğrisi.

Şekil 4.28’de görüldüğü gibi verinin dengesizliğinden kaynaklı AİK eğrisinde bir düşüş yaşamadığı görülmüştür. Karar ağaçlarındaki sonucun rastgele ormandaki sonuçla aynı olmasının nedeni yine etiket mantığında çalışmasından kaynaklanmaktadır.



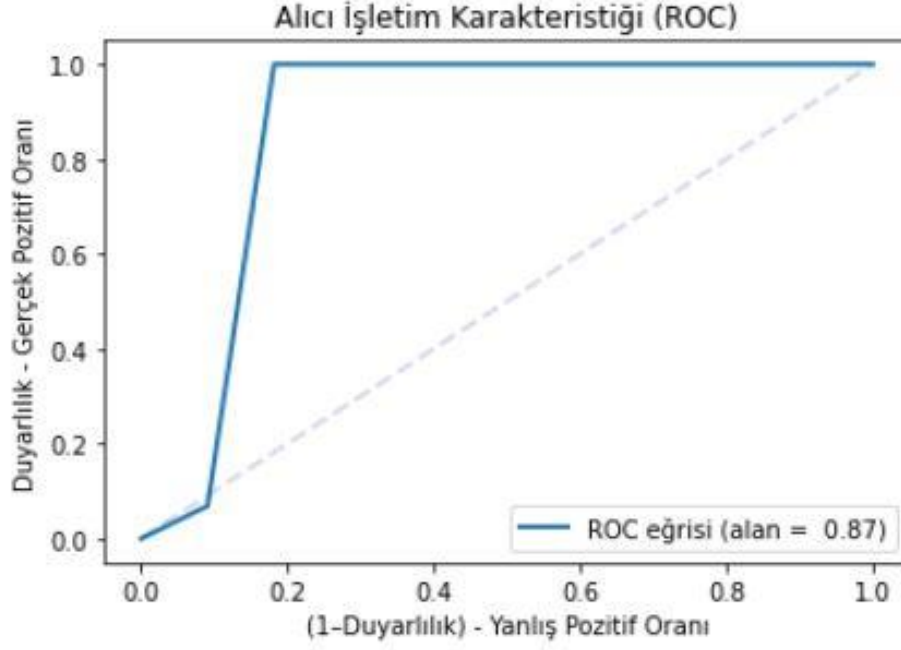
Şekil 4.29. Scikit-Learn XGBoost sınıflandırma modelinin zor entübe veri setinin sonuçlarının AİK eğrisi.

Şekil 4.29’da görüldüğü gibi verinin dengesizliğinden kaynaklı AİK eğrisinde bir düşüş yaşamadığı görülmüştür. XGBoost temelde karar ağaçları gibi çalıştığından sonucun karar ağaçlarındaki sonuçla aynı olmasının nedeni yine etiket mantığında çalışmasından kaynaklanmaktadır.



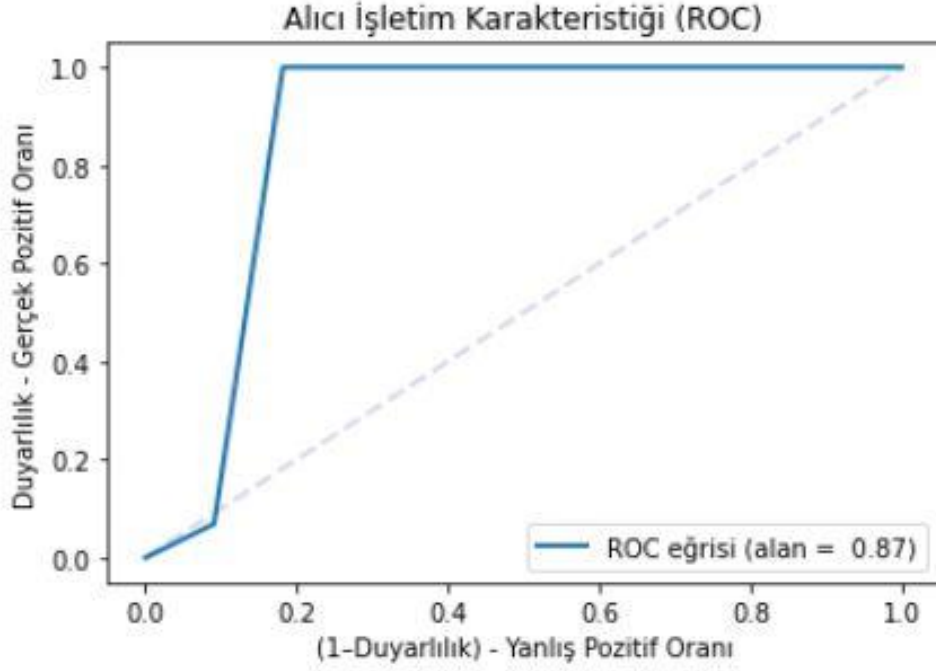
Şekil 4.30. Scikit-Learn k-en yakın komşu sınıflandırma modelinin zor entübe veri setinin sonuçlarının AİK eğrisi.

Şekil 4.30'da gözlemlenen AİK eğrisinde oldukça başarısızdır. Eğrinin bize söylediği girdinin sonucunu %50 olasılıkla bulabileceğidir. Bu algoritmanın bu tip bir model veri setine uygun olmadığı bu eğri ile anlaşılmaktadır.



Şekil 4.31. Scikit-Learn destek vektör makinesi sınıflandırma modelinin zor entübe veri setinin sonuçlarının AİK eğrisi.

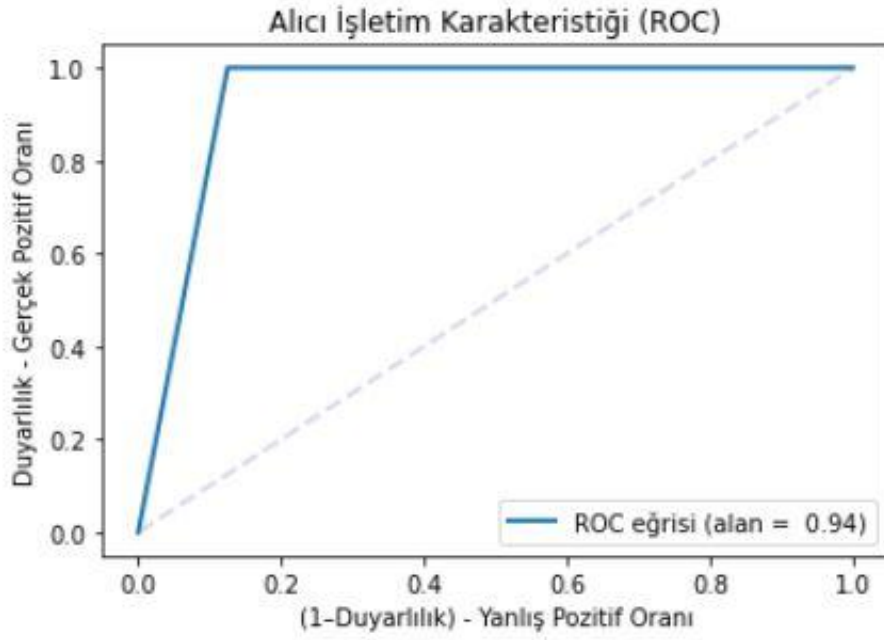
Şekil 4.31’de görüldüğü gibi verinin dengesizliğinden kaynaklı AİK eğrisinde bir düşüş yaşayıp daha sonra düzelmiştir. Keras yapay sinir ağlarında ve Scikit-Learn çok katmanlı algılayıcılar olduğu gibi destek vektör makinesinde de benzer bir durum gözlemlenmiştir.



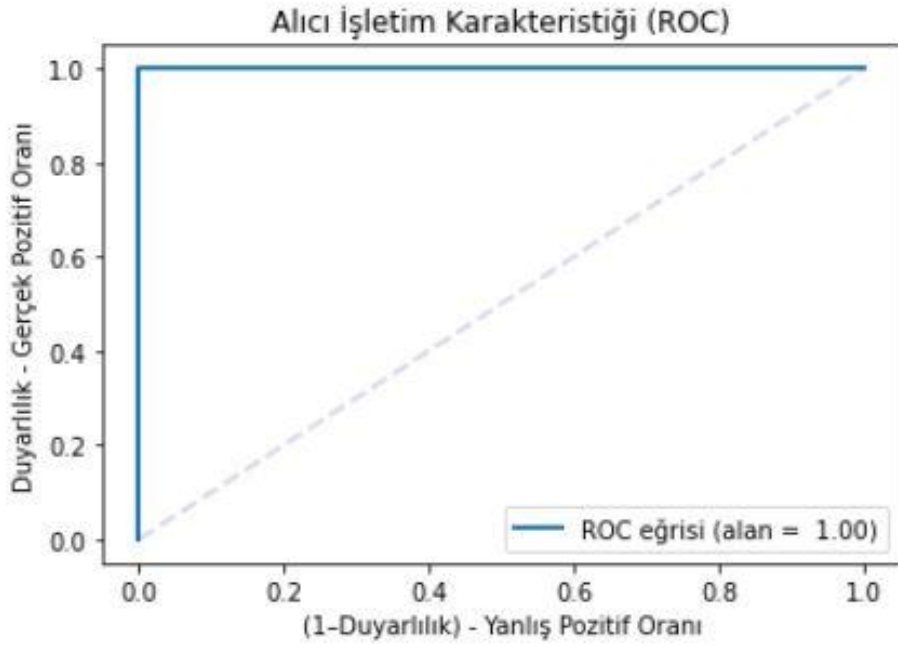
Şekil 4.32. Scikit-Learn çok katmanlı sınıflandırma modelinin BDB çapraz validasyon yöntemi ile zor entübe veri setinin sonuçlarının AİK eğrisi.

Şekil 4.32’de görüldüğü gibi verinin dengesizliğinden kaynaklı AİK eğrisinde bir düşüş yaşayıp daha sonra düzelmiştir. Scikit-Learn k-kat çapraz validasyon yöntemi ile yapılan çok katmanlı algılayıcılardan farklı bir yapı görülmemiştir.

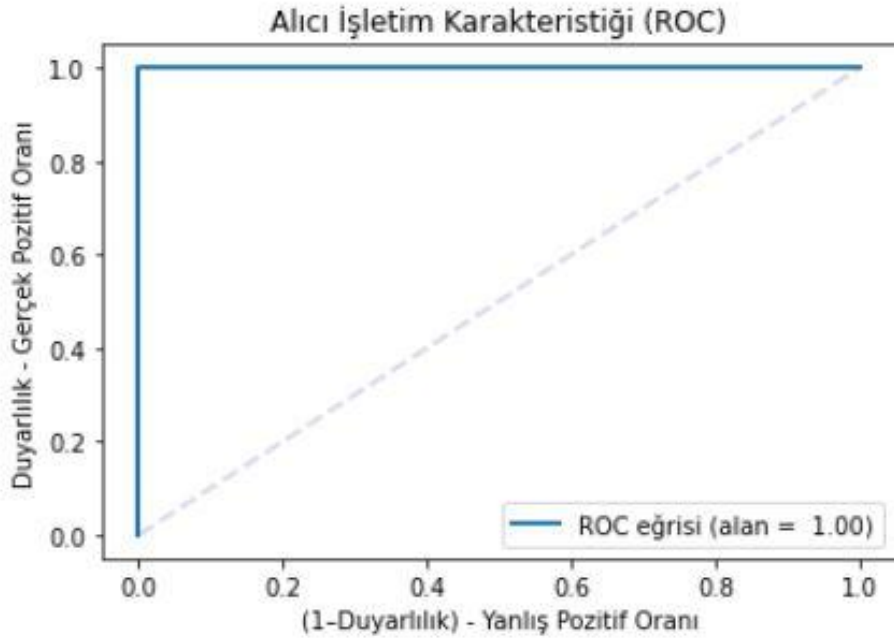
4.4.2. Entübe Veri Setinin ROC Eğrisi



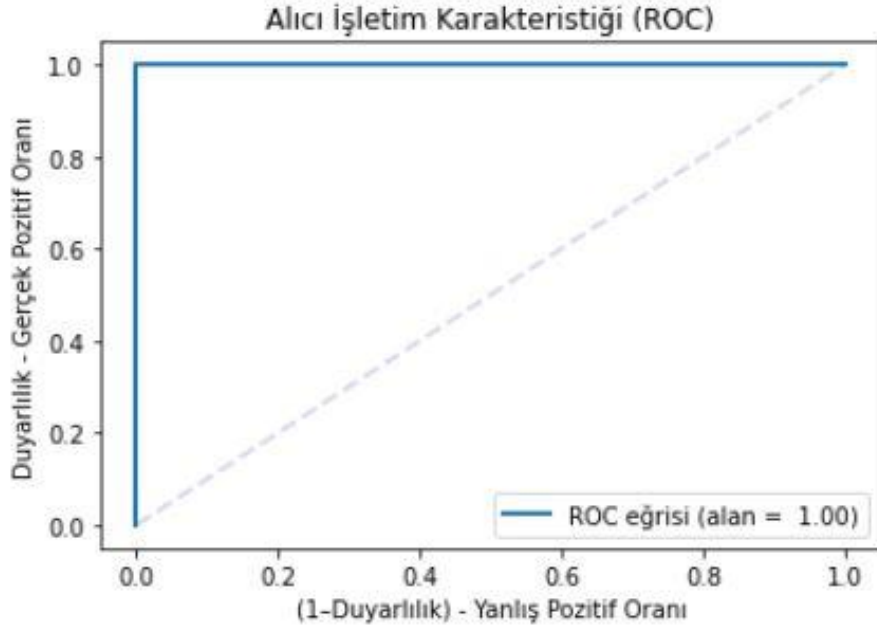
Şekil 4.33. Keras yapay sinir ağları sınıflandırma modelinin entübe veri setinin sonuçlarının AİK eğrisi.



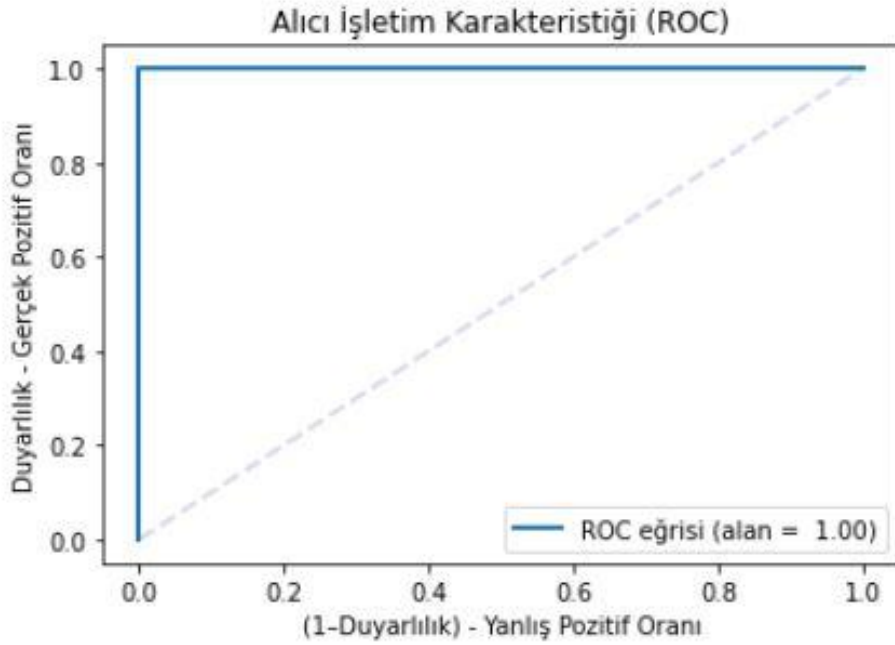
Şekil 4.34. Scikit-Learn çok katmanlı algılayıcılar sınıflandırma modelinin entübe veri setinin sonuçlarının AİK eğrisi.



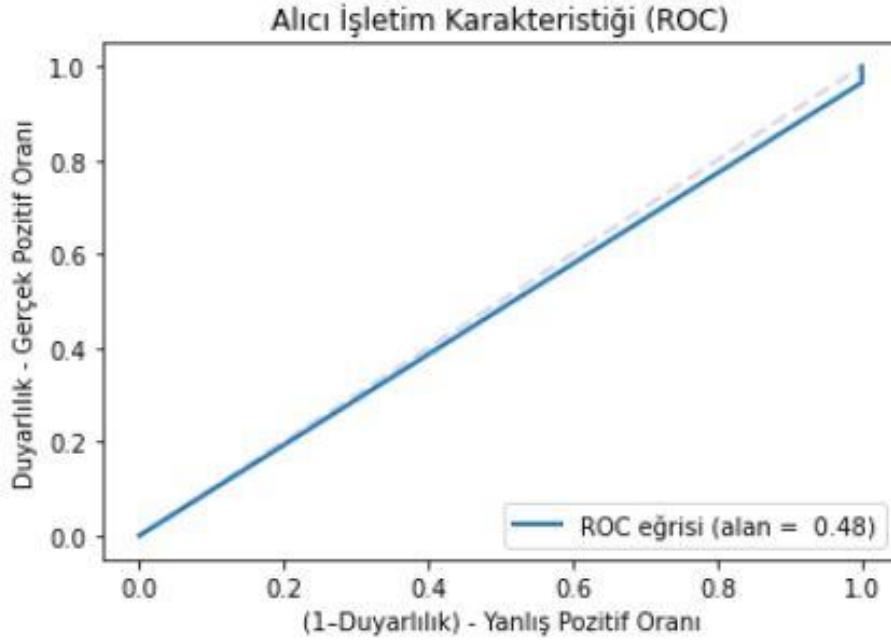
Şekil 4.35. Scikit-Learn rastgele orman sınıflandırma modelinin entübe veri setinin sonuçlarının AİK eğrisi.



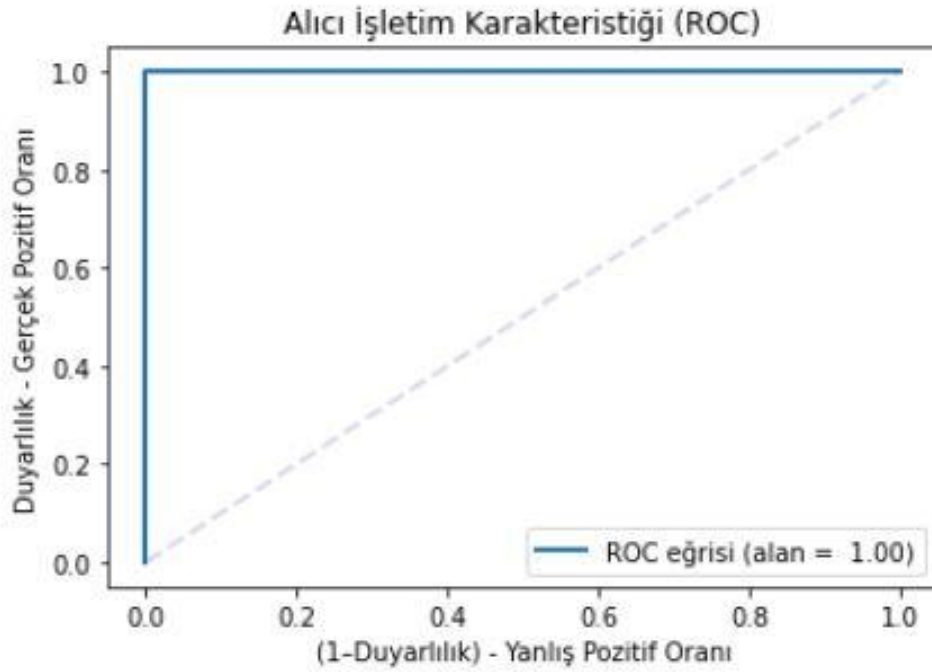
Şekil 4.36. Scikit-Learn karar ağaçları sınıflandırma modelinin entübe veri setinin sonuçlarının AİK eğrisi.



Şekil 4.37. Scikit-Learn XGBoost sınıflandırma modelinin entübe veri setinin sonuçlarının AİK eğrisi.



Şekil 4.38. Scikit-Learn k-en yakın komşu sınıflandırma modelinin entübe veri setinin sonuçlarının AİK eğrisi.



Şekil 4.39. Scikit-Learn destek vektör makinesi sınıflandırma modelinin entübe veri setinin sonuçlarının AİK eğrisi.

Şekil 4.34, Şekil 4.35, Şekil 4.36, Şekil 4.37 ve Şekil 4.39 %100 oranında bir doğruluk olduğu gözlemlenmiştir. Entübe veri setindeki k-kat çapraz validasyon ile kullanılan bu yöntemlerin oldukça başarılı olmuştur. Şekil 4.33'deki Keras yapay sinir ağları sınıflandırma modelindeki AİK eğrisi %94 oranında bir doğruluk oranı vermiştir. Bunun sebebi Hold-Out çapraz validasyon yönteminden kaynaklandığı söylenebilir. Şekil 4.38 'dek KEK algoritması oldukça kötü sonuç vermiştir. Bu çalışmada zor entübe ve entübe veri setinde KEK'in en başarısız algoritma olduğu gözlemlenmiştir.



5. TARTIŞMA VE SONUÇ

İstanbul Üniversitesi-Cerrahpaşa Kardiyoloji Enstitüsü'ne ait olan retrospektif hasta verileri kullanılmıştır. Farklı yapay zekâ algoritmaları ile zor entübasyon olacak hastaların bir hekim yardımı olmaksızın hastaların verileri üzerinden entübasyon kararının verilmesi amaçlamıştır. Çalışmada kullanılan zor entübe hasta sayısının az olmasından kaynaklı yapay sinir ağlarında iki modelde karar vermenin karıştırma olasılığının arttığı gözlemlenmiştir. Birbirine yakın olan bu iki durum hastanın entübe veya zor entübe olması, zor entübe hasta sayısının çok olması durumunda daha kesin (%80 ve üzeri) sonuçlarla karşılaşılabileceği mümkündür. Kovid rahatsızlığı döneminde toplanan bu veriler değerlendirildiğinde kovid gibi göğüs hastalıklarının daha az rastlandığı dönemlerde dahi zor entübe verisi toplamak oldukça güç ve uzun bir zaman aralığı gerektirmektedir. Yapılan uygulamalar değerlendirildiğinde test sonuçlarının oldukça başarılı olduğu bulgular bölümündeki grafik ve tablolar üzerinden gösterilmiştir. Bu önerilen modellerin doğru, anlamlı veriler içerdiğini gösterir.

Bu çalışma entübe olabilecek hastaları veya zor entübe olabilecek hastaları ayırmak amacı ile tasarlanmıştır. Modelleme sırasında entübe hastalarına oldukça benzeyen ve zor entübe denilen hasta verileri incelenerek karar verme süreçlerine katkı sağlanması amaçlanmıştır. Bu nedenle her iki şekilde de hem entübe ve entübe olmayan veri seti ile, entübe, entübe olmayan ve zor entübe olan veri seti tüm algoritmalar ile çalışılmıştır.

Bu çalışmada kullanılan verilerin 80 tanesi kovid-19 hastalarından oluşmaktadır. Hasta verilerinin az olmasından dolayı zor entübe hasta test sonuçlarının verdiği doğru bulma sonucu karışıklık matrislerinde görüldüğü üzere ortalama %50 civarındadır. Fakat bu bize daha çok zor entübe hasta verisi olması durumunda sonucun pozitif yönde artacağını de göstermiştir.

Karışıklık matrisleri ve AİK eğrilerine bakıldığında zor entübe veri setindeki verilerin dengesizliğinden kaynaklı sonuçların anlamlılığını yitirdiği görülmüştür. Entübe veri setine bakıldığında ise verilerin çok dengeli olmasa da nasıl başarılı olduğu karışıklık matrisleri ve AİK eğrilerinden anlaşılmıştır. Bu iki veri setinde de çoğu algoritma genel anlamda başarılı olmuştur. Bu veri setlerinde başarılı olmayan tek algoritma k-en yakın komşu olduğu gözlemlenmiştir.

Çalışmadaki izlenen yollar derin öğrenme ve makine öğrenmesi yapılarıdır. Keras UPA daha ön planda tutulmuş olmasının nedeni ise hem literatürde hem de genelde derin öğrenme alanında oldukça popüler ve arkasında TensorFlow çalışmasıdır. Scikit-Learn ise derin öğrenme alanından daha çok makine öğrenimi alanında çalışır. Bu çalışmada Scikit-Learn'ün veriyi eğitim ve test veri setlerine parçalama konusunda daha başarılı olduğu görülmüştür. Bunun teze katkısı ise veriyi el ile seçerek ayırmayıp rastgele bir şekilde eğitim ve test veri setini oluşturmasıdır.

Bu çalışma ile doğru ve yeterli sayıda hasta verisi olması durumunda zor entübasyon gibi anestezi uzmanı olmadan verilemeyen kararların bir bilgisayarın hızlı ve doğru karar vermesi ile başarılabileceği gösterilmiştir. Böylece hastaların teşhisinin kolaylaşması mümkün olabilecektir. Yapay zekânın hastanelerde bu tip teşhis koyma konularında da oldukça başarılı ve kullanışlı yöntem olduğu gösterilmiştir.

KAYNAKLAR

- [1]. Benumof JL. The ASA difficult airway algorithm: new thoughts/considerations. Annual Refresher Course Lectures. Park Ridge, IL, USA: Am Soc Anesthesiologists 1997; 241: 1–7.
- [2]. Bellhouse CP, Dore C. Criteria for estimating likelihood of difficulty of endotracheal intubation with the Macintosh laryngoscope. *Anaesth Intens Care* 1988; 18: 329–337.
- [3]. Caplan RA, Posner KL, Ward RJ, Cheney FW. Adverse respiratory events in anesthesia: a closed claims analysis. *Anesthesiology* 1990; 72: 828–833.
- [4]. Benumof JL. Definition and incidence of the difficult airway. In: Benumof JL, ed. *Airway Management: Principles and Practice*. Philadelphia, USA: Mosby, 1995: 121–125.
- [5]. Rose DK, Cohen MM. The airway: problems and predictions in 18,500 patients. *Can J Anaesth* 1994; 41: 372–383.
- [6]. Schwartz DE, Matthay MA, Cohen NH. Death and other complications of emergency airway management in critically ill adults. A prospective investigation of 297 tracheal intubations. *Anesthesiology* 1995; 82: 367–376.
- [7]. Nekhendzy V, Biro P. Airway management in head and neck surgery. In: Hagberg CA, Artim C, Aziz M, editors. *Hagberg and Benumof's airway management*. 3rd edition. Philadelphia: Elsevier; 2018. p. 668–91.
- [8]. Arne J, Descoins P, Fusciardi J, et al. Preoperative assessment for difficult intubation in general and ENT surgery: predictive value of a clinical multivariate risk index. *Br J Anaesth* 1998;80(2):140–6.
- [9]. Cook TM, Woodall N, Frerk C. NAP 4. Major complications of airway management in the United Kingdom. *Br J Anaesth* 2011;106: 617e31.
- [10]. Cormack RS, Lehane J. Difficult tracheal intubation in obstetrics. *Anaesthesia* 1984; 3e: 1105–1111.

- [11]. Benumof JL. Recognition of the difficult airway. In: Benumof JL, ed. *Airway Management: Principles and Practice*. Philadelphia, USA: Mosby, 1995: 126–142.
- [12]. Management of difficult intubation M. Janssens and G. Hartstein Department of Anaesthesiology, University Hospital of Liege, Belgium *European Journal of Anaesthesiology* 2001, 18, 3–12.
- [13]. Mallampati SR, Gatt SP, Gugino LD et al. A clinical sign to predict difficult tracheal intubation: a prospective study. *Can Anaesth Soc J* 1985; 32: 429–434.
- [14]. Samssoon GLT, Young JRB. Difficult tracheal intubation: a retrospective study. *Anaesthesia* 1987; 42: 487–490.
- [15]. Sumit Gajree and Kevin J O’Hare Identification of the difficult airway, Royal College of Anaesthetists CPD Matrix: 1C01, 1C02, 2A01, 3A01.
- [16]. G. Johom, M.Ronyne, A.J. Cunningham, Prediction of difficult tracheal intubation Beaumont Hospital, Department of Anaesthesia and Intensive Care, Dublin, Ireland, *European Journal of Anaesthesiology* 2003; 20: 31-36
- [17]. Woodall N, Frerk C, Cook TM. Can we make airway management (even) safer? lessons from national audit. *Anaesthesia* 2011;66(Suppl 2):27–33.
- [18]. Hohn A, Kaulins T, Hinkelbein J, et al. Awake tracheotomy in a patient with stridor and dyspnoea caused by a sizeable malignant thyroid tumor: a case report and short review of the literature. *Clin Case Rep* 2017;5(11):1891–5.
- [19]. Apfelbaum JL, Hagberg CA, Caplan RA, et al. Practice guidelines for management of the difficult airway: an updated report by the American Society of Anesthesiologists Task Force on Management of the Difficult Airway. *Anesthesiology* 2013;118(2):251–70.
- [20]. Artime C, Hagberg C. Airway management. In: Miller RD, editor. *Miller’s anesthesia*. 8th edition. Philadelphia: Churchill Livingstone/Elsevier; 2014. p. 1647–83.

- [21]. Bertil Schmidt Q3 and Andreas Hildebrandt, Deep learning in next-generation sequencing, Institut für Informatik, Johannes Gutenberg University Mainz, Germany, Drug Discovery Today _ Volume 00, Number 00 _ October 2020
- [22]. Rosenblatt, F. (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* 65, 386
- [23]. Amisha, Paras Malik, Monika Pathania, and Vyas Kumar Rathaur. "Overview of artificial intelligence in medicine." *Journal of family medicine and primary care* 8.7 (2019): 2328.
- [24]. Sagheer, Alaa, Mohammed Zidan, and Mohammed M. Abdelsamea. "A novel autonomous perceptron model for pattern classification applications." *Entropy* 21.8 (2019): 763.
- [25]. Dorado-Díaz, P. Ignacio, et al. "Applications of artificial intelligence in cardiology. The future is already here." *Revista Española de Cardiología (English Edition)* 72.12 (2019): 1065-1075.
- [26]. Yoo JS, Kim JB, Jung SH, et al. Mitral durability after robotic mitral valve repair: analysis of 200 consecutive mitral regurgitation repairs. *J Thorac Cardiovasc Surg.* 2014; 148:2773–2779.
- [27]. Melero-Alegria JI, Cascon M, Romero A, et al. SALMANTICOR study. Rationale and design of a population-based study to identify structural heart disease abnormalities: a spatial and machine learning analysis. *BMJ Open.* 2019;9: e024605.
- [28]. Rajkomar A, Dean J, Kohane I. Machine Learning in Medicine. *N Engl J Med.* 2019; 380:1347–1358.
- [29]. Murphy, Kevin P. *Machine learning: a probabilistic perspective.* MIT press, 2012.
- [30]. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature.* 2015; 521:436–444.

- [31]. Svozil, Daniel, Vladimir Kvasnicka, and Jiri Pospichal. "Introduction to multi-layer feed-forward neural networks." *Chemometrics and intelligent laboratory systems* 39.1 (1997): 43-62.
- [32]. Le, William Trung, et al. "Overview of machine learning: part 2: deep learning for medical image analysis." *Neuroimaging Clinics* 30.4 (2020): 417-431.
- [33]. Goodfellow I, Bengio Y, Courville A. *Deep learning*. Cambridge: MIT Press; 2016.
- [34]. Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. *Proceedings of the 30th Annual International Conference on Machine Learning*, Vol. 30. Atlanta (GA); 2013. p. 3.
- [35]. Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Fort Lauderdale (FL); 2011. p. 315–
- [36]. Char DS, Shah NH, Magnus D. Implementing machine learning in health care — addressing ethical challenges. *N Engl J Med*. 2018; 378:981–983.
- [37]. Sturm I, Lapuschkin S, Samek W, Muller KR. Interpretable deep neural networks for single-trial EEG classification. *J Neurosci Methods*. 2016; 274:141–145.
- [38]. Schmidt, Bertil, and Andreas Hildebrandt. "Deep learning in next-generation sequencing." *Drug Discovery Today* (2020).
- [39]. Abadi, M. et al. (2016) Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv arXiv:1603, 04467
- [40]. Chollet, F. (2018) Keras: The Python Deep Learning Library. Astrophysics Source Code Library
- [41]. Paszke, A. et al. (2019) Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* 32, 8026–8037

- [42]. Géron, Aurélien. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, 2019.
- [43]. Moustafa, Moustafa Abdelaziz, et al. "Defining difficult laryngoscopy findings by using multiple parameters: A machine learning approach." *Egyptian Journal of Anaesthesia* 33.2 (2017): 153-158.
- [44]. Carlson, Jestin N., et al. "A novel artificial intelligence system for endotracheal intubation." *Prehospital Emergency Care* 20.5 (2016): 667-671.
- [45]. Cook, Fabrice, et al. "Prospective validation of a new airway management algorithm and predictive features of intubation difficulty." *British journal of anaesthesia* 122.2 (2019): 245-254.
- [46]. Matava, Clyde, et al. "Artificial intelligence, machine learning and the pediatric airway." *Pediatric Anesthesia* 30.3 (2020): 264-268.
- [47]. Garreta, Raul, and Guillermo Moncecchi. *Learning scikit-learn: machine learning in python*. Packt Publishing Ltd, 2013.
- [48]. <https://keras.io/> Erişim Tarihi:03/10/2021
- [49]. <https://scikit-learn.org/stable/index.html> Erişim Tarihi:19/12/2021

EKLER**EK-1**

Retrospektif hasta verilerinin kullanılmasına dair etik kurulu izin belgesi

Tarih: 03.06.2021

İstanbul Üniversitesi-Cerrahpaşa, Kardiyoloji Enstitüsü, Anesteziyoloji ve Reanimasyon
Anabilim Dalı'na

İstanbul Üniversitesi-Cerrahpaşa, Lisansüstü Eğitim Enstitüsü Biyomedikal Mühendisliği bölümünde Doçent Doktor Aysel Ersoy Yılmaz ile eş danışmanı olduğum 4801191000 numaralı Muhammed Emin Şenol isimli öğrencinin "**Entübasyon İşleminin Yapay Zeka Algoritmaları ile Modellenmesi**" isimli tezinde klinik arşiv verilerimizde yer alan rastgele seçilmiş klinik verilerden 80 erkek ve 80 kadın olacak şekilde 160 hastanın verilerinin paylaşımına izin verilmesi hususunda gereğini arz ederim.

Dr. Öğr. Üyesi Hülya YILMAZ AK

İmzası

BAŞ HEKİM ONAYLIDIR