



**T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**



Doktora Tezi

**DENGESİZ KLİNİK VERİLER İÇİN KARAR DESTEK ÖNERİSİ:
AKUT APANDİSİT ÖRNEĞİ**

Kevser ŞAHİNBAŞ

Enformatik Anabilim Dalı

Enformatik Programı

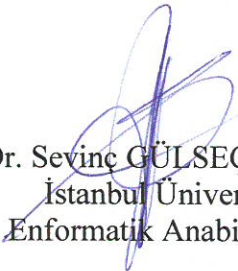
**DANIŞMAN
Prof. Dr. Sevinç GÜLSEÇEN**

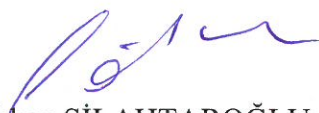
Ağustos, 2019

İSTANBUL

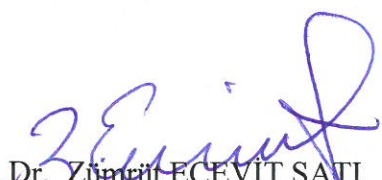
Bu çalışma, 20.08.2019 tarihinde aşağıdaki jüri tarafından Enformatik Anabilim Dalı, Enformatik Programında Doktora tezi olarak kabul edilmiştir.

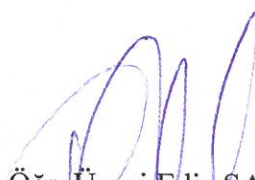
Tez Jürisi


Prof. Dr. Sevinç GÜLSEÇEN(Danışman)
İstanbul Üniversitesi
Enformatik Anabilim Dalı


Prof. Dr. Gökhan SİLAHTAROĞLU
Medipol Üniversitesi
İşletme ve Yönetim Bilimleri Fakültesi


Doç. Dr. Seda TOLUN
İstanbul Üniversitesi
İşletme Fakültesi


Doç. Dr. Zümrüt ECEVİT SATI
İstanbul Üniversitesi
Siyasal Bilgiler Fakültesi


Dr. Öğr. Üyesi Ediz ŞAYKOL
Beykent Üniversitesi
Mühendislik-Mimarlık Fakültesi



20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi’nin aboneli olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü’nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

Bu çalışmada kullanılan veriler T.C. Medipol Üniversitesi Sosyal Bilimler Bilimsel Araştırmalar Etik Kurulu tarafından sağlanmıştır.

ÖNSÖZ

Ülkemiz çapında sık rastlanılan acil durumlardan biri olan akut apandisit klinik tanı koymasını hekim açısından kolaylaştırmak amacıyla uzman hekimlerle görüşülmüş ve onların önerileri doğrultusunda bu çalışma ortaya çıkmıştır. Akut batın, akut karın ağrısı, bağırsak tıkanıkları, gastroenterit vb ile akut apandisit belirtileri çok fazla karıştırıldığı için bu çalışma genel cerrahide hekimlerin karar vermesinde kolaylık sağlayacaktır.

Bu tezin konusu olan akut apandisit klinik tanı koymasında en iyi performansı gösteren algoritma belirlenmiş olup akut apandisit teşhisinde kullanılan Alvarado skorlarına yeni bir bakış açısı geliştirilerek hekim için Karar Destek Sistemi oluşturulmuştur.

Tez çalışmamın yürütülmesinde desteğini esirgemeyen, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sevgili danışmanım Prof. Dr. Sevinç GÜLSEÇEN'e değerli yönlendirmeleri ve her türlü desteği için en kalbi duygularım ile sonsuz teşekkürlerimi sunarım.

Tez izleme komitesinde beni yönlendirmeleriyle büyük katkılar sağlayan Prof.Dr. Gökhan SİLAHTAROĞLU ve Doç.Dr. Seda TOLUN'a gönülden teşekkürlerimi sunarım.

Tezde ihtiyaç duyulan verinin temin edilebilmesini sağlayan Medipol Mega Üniversite Hastanesi Genel Cerrahi Kliniği'ne, uzmanlık tezindeki verileri sağlayan Doç.Dr. İlker Murat ARER'e, İstanbul Esenyurt Necmi Kadioğlu Devlet Hastanesi eski Başhekim Dr.Öğr. Üyesi Ahmet Atilla YILMAZ'a değerli katkıları için teşekkürlerimi sunarım.

Tez konumun ortaya çıkmasına vesile olan, tıbbi olarak beni bilgilendiren ve yönlendiren Prof.Dr. Ayşegül KAPTANOĞLU'na teşekkürlerimi sunarım.

Hem iyi gün hem kötü gün dostum Nergiz BAŞEĞMEZ'e, makine öğrenmesi ve tezimin ilerleme aşamalarında kıymetli yönlendirmeleri için Doç.Dr. Ferhat Özgür ÇATAK'a, Arş. Gör. Dr.Elif KARTAL'a ve Enformatik Bölümü değerli hocalarına teşekkürlerimi sunarım.

Tezimi, annem Satı ŞAHİNBAŞ, babam Osman ŞAHİNBAŞ başta olmak üzere canım aileme ithaf ederim.

Ağustos 2019

Kevser ŞAHİNBAŞ

İÇİNDEKİLER

Sayfa No

ÖNSÖZ	iv
İÇİNDEKİLER.....	v
ŞEKİL LİSTESİ	viii
TABLO LİSTESİ.....	x
SİMGE VE KISALTMA LİSTESİ	xiii
ÖZET	xiv
SUMMARY	xv
1. GİRİŞ	1
2. GENEL KISIMLAR.....	6
2.1. DENGESİZ VERİ SETLERİ.....	6
2.1.1. Sınıf Dengesizliği Yaklaşımları	6
2.1.1.1. <i>Rasgele Aşırı Örnekleme (ROS)</i>	7
2.1.1.2. <i>Rasgele Az Örnekleme(Undersampling)</i>	8
2.1.1.3. <i>SMOTE</i>	9
2.2. AKUT APANDİSİT	11
2.3. MAKİNE ÖĞRENMESİ	15
2.4. SINIFLANDIRMA	19
2.5. MAKİNE ÖĞRENMESİ SÜRECİ ADIMLARI	20
2.6. MODELLEME	21
2.6.1. Naive Bayes Sınıflandırıcı.....	21
2.6.2. K-En Yakın Komşu Algoritması	24
2.6.3. Logistik Regresyon.....	26
2.6.4. Yapay Sinir Ağları.....	28
2.6.4.1. <i>Yapay Sinir Ağlarının Eğitilmesi</i>	31
2.6.4.2. <i>Aktivasyon fonksiyonları</i>	32
2.6.5. Destek Vektör Makinesi (DVM)	33
2.6.5.1. <i>Doğrusal Olarak Ayrılabilme Durumu</i>	35
2.6.5.2. <i>Primal Form</i>	37
2.6.5.3. <i>Yumuşak Marj</i>	39
2.6.5.4. <i>Doğrusal Olmayan Sınıflandırıcı İçin Çekirdek Hilesi</i>	39

2.6.6.	Rasgele Orman Algoritması (Random Forest)	41
2.7.	MODEL DEĞERLENDİRME VE SEÇİMİ	46
2.7.1.	Model Performans Değerlendirme Yöntemleri	46
2.7.2.	Model Performans Değerlendirme Ölçüleri	47
2.7.2.1.	<i>Doğruluk</i>	49
2.7.2.2.	<i>Seçicilik</i>	49
2.7.2.3.	<i>Duyarlılık</i>	50
2.7.2.4.	<i>Kesinlik</i>	50
2.7.2.5.	<i>F-Ölçütü</i>	50
2.7.2.6.	<i>Kappa Değeri</i>	51
2.7.2.7.	<i>ROC Eğrisinin Altında Kalan Alan (AUC)</i>	51
3.	MALZEME VE YÖNTEM	53
3.1.	KULLANILAN YAZILIM.....	53
3.2.	TAHMİN İÇİN KULLANILAN NİTELİKLER	54
4.	BULGULAR	58
4.1.	SINIF DENGELEME YÖNTEMLERİ	58
4.1.1.	Random Over Sampling (ROS).....	60
4.1.2.	SMOTE.....	61
4.1.3.	Random Undersampling (RUS).....	62
4.1.4.	Test ve Eğitim Veriseti	64
4.2.	ANALİZ SONUÇLARI.....	64
4.2.1.	RASGELE ORMAN (RANDOM FOREST).....	66
4.2.2.	DESTEK VEKTÖR MAKİNESİ (DVM)	73
4.2.3.	NAİVE BAYES ALGORİTMASI	77
4.2.4.	K- EN YAKIN KOMŞU (K-NEAREST NEIGHBOR) ALGORİTMASI	81
4.2.5.	LOJİSTİK REGRESYON	85
4.2.6.	YAPAY SİNİR AĞLARI	91
4.2.7.	ALVARADO SKORU BAŞARI ÖLÇÜMÜ	95
4.2.8.	YAŞ VE CİNSİYET NİTELİKLİKLERİ EKLENEREK PERFORMANS BAŞARI ÖLÇÜMÜ	96
4.3.	SINIFLANDIRMA MODELLERİNİN PERFORMANSLARININ KARŞILAŞTIRILMASI	102
4.4.	KARAR DESTEK SİSTEMİ UYGULAMASI.....	108
5.	TARTIŞMA VE SONUÇ	111
	KAYNAKLAR	115

EKLER	127
ÖZGEÇMİŞ	139



ŞEKİL LİSTESİ

	Sayfa No
Şekil 2.1: ROS algoritması (Sun ve diğ., 2018).....	8
Şekil 2.2: SMOTE algoritması (Sun ve diğ., 2018).....	10
Şekil 2.3: SMOTE algoritması adımları (Sun ve diğ., 2018).....	11
Şekil 2.4: Naive bayes sınıflandırıcısı (Friedman ve diğ., 1997).....	23
Şekil 2.5: $k=3$ için öklid mesafesi (Zhang, 2019).....	25
Şekil 2.6: Sigmoid fonksiyonu.....	27
Şekil 2.7: Temel yapay sinir ağı hücresi (Haykin, 2008).....	29
Şekil 2.8: Yapay sinir ağları katmanları (Vassilis, Z., A. ve diğ., 2019).....	31
Şekil 2.9: Sigmoid.....	33
Şekil 2.10: Tanjant fonksiyonu.....	33
Şekil 2.11: ReLU fonksiyonu.....	33
Şekil 2.12: İkili sınıflandırma için eğitim setini $+1$ ve -1 şeklinde ikiye ayıran hiperdüzlemlerin gösterimi (Çatak, F., Ö., 2015).....	36
Şekil 2.13: Sert marjin doğrusal ayrıştırılabilir veri seti (Çatak, F., Ö., 2015).....	36
Şekil 2.14: Rasgele orman şeması (Yu ve diğ., 2018).....	44
Şekil 2.15: Rasgele orman algoritması adımları (Yu ve diğ., 2018).....	45
Şekil 2.16: 5-Kat çapraz geçерleme (Rogers ve Girolami 2011).....	47
Şekil 2.17: ROC eğrisi (Yu ve diğ., 2018).....	52
Şekil 3.1: Anaconda arayüzü.....	53
Şekil 3.2: WBC, NE% ve yaş niteliklerine ait kutu grafiğı çizimleri.....	57
Şekil 4.1: Sınıf dağılımı histogram.....	58
Şekil 4.2: Sınıf dağılımı.....	59
Şekil 4.3: Korelasyon matrisi.....	59

Şekil 4.4: Random oversampling veri dağılımı.	60
Şekil 4.5: Random oversampling veri dağılımı histogramı.	61
Şekil 4.6: SMOTE veri seti sınıf dağılımı.....	62
Şekil 4.7: SMOTE veri seti sınıf dağılımı histogramı.	62
Şekil 4.8: Random undersampling sınıf dağılımı.....	63
Şekil 4.9: Random undersampling sınıf dağılımı histogram.....	63
Şekil 4.10: Normal nitelik önemi.	67
Şekil 4.11: SMOTE nitelik önemi.....	68
Şekil 4.12: ROS nitelik önemi.	68
Şekil 4.13: RUS nitelik önemi.	69
Şekil 4.14: RO sınıflandırma ROC eğrileri.....	73
Şekil 4.15: DVM algoritması ROC eğrileri.	77
Şekil 4.16: Naive bayes algoritması ROC eğrileri.	81
Şekil 4.17: K-EY algoritması ROC eğrileri.	85
Şekil 4.18: Lojistik regresyon ROC eğrileri.	91
Şekil 4.19: YSA ROC eğrileri.....	95
Şekil 4.20: DVM yaş+cinsiyet nitelikleri eklendikten sonra ROC Eğrisi	97
Şekil 4.21: RO algoritması ROS yöntemi yaş+ cinsiyet nitelikleri eklenerek özellik önemi	99
Şekil 4.22: KDS- AA olan hasta örneği.	109
Şekil 4.23: KDS- AA olmayan hasta örneği.	110

TABLO LİSTESİ

Sayfa No

Tablo 2.1: Akut apandisit için alvarado skorlaması tablosu.....	15
Tablo 2.2: DVM yöntemine ait bazı çekirdek fonksiyonlar.	41
Tablo 2.3: İki sınıf için oluşturulmuş örnek bir karmaşıklık matrisi (Han ve Kamber, 2006).....	48
Tablo 3.1: Tahmin için kullanılan nitelikler.	56
Tablo 3.2: Veri setinde niteliklere ait tanımlayıcı bilgi.	57
Tablo 4.1: İki sınıf için oluşturulmuş örnek bir karmaşıklık matrisi.	64
Tablo 4.2: Nitelik önemi.....	66
Tablo 4.3: RO algoritması normal karmaşıklık tablosu.	69
Tablo 4.4: RO algoritması SMOTE karmaşıklık tablosu.	70
Tablo 4.5: RO algoritması ROS karmaşıklık tablosu.	70
Tablo 4.6: RO algoritması RUS karmaşıklık tablosu.	70
Tablo 4.7: RO algoritması holdout metrikleri.	71
Tablo 4.8: RO algoritması 5 kat çapraz geçерleme sonuçları.	72
Tablo 4.9: DVM algoritması normal karmaşıklık tablosu.....	74
Tablo 4.10: DVM algoritması SMOTE karmaşıklık tablosu.	74
Tablo 4.11: DVM algoritması ROS karmaşıklık tablosu.	74
Tablo 4.12: DVM algoritması RUS karmaşıklık tablosu.	75
Tablo 4.13: DVM algoritması metrikleri.....	75
Tablo 4.14: DVM algoritması 5 kat çapraz geçерleme metrikleri.	76
Tablo 4.15: NB algoritması normal karmaşıklık tablosu.	77
Tablo 4.16: NB algoritması SMOTE karmaşıklık tablosu.	78
Tablo 4.17: NB algoritması ROS karmaşıklık tablosu.	78

Tablo 4.18: NB algoritması RUS karmaşıklık tablosu.	78
Tablo 4.19: NB algoritması holdout metrikleri.	79
Tablo 4.20: NB algoritması 5 kat çaprazlama sonuçları.....	80
Tablo 4.21: K-EY algoritması normal karmaşıklık tablosu.	81
Tablo 4.22: K-EY algoritması SMOTE karmaşıklık tablosu.	82
Tablo 4.23: K-EY algoritması ROS karmaşıklık tablosu.	82
Tablo 4.24: K-EY algoritması RUS karmaşıklık tablosu.	82
Tablo 4.25: K-EY algoritması holdut metrikleri.	83
Tablo 4.26: K-EY algoritması 5 kat çapraz geçерleme sonuçları.	84
Tablo 4.27: LR normal karmaşıklık tablosu.	88
Tablo 4.28: LR algoritması SMOTE karmaşıklık tablosu.....	89
Tablo 4.29: LR algoritması ROS karmaşıklık tablosu.	89
Tablo 4.30: LR algoritması RUS karmaşıklık tablosu.	89
Tablo 4.31: LR algoritması holdout metrikleri.....	90
Tablo 4.32: LR algoritması 5 kat çapraz geçерleme sonuçları.	90
Tablo 4.33: YSA algoritması normal karmaşıklık tablosu.	92
Tablo 4.34: YSA algoritması SMOTE karmaşıklık tablosu.....	92
Tablo 4.35: YSA algoritması ROS karmaşıklık tablosu.....	93
Tablo 4.36: YSA algoritması RUS karmaşıklık tablosu.....	93
Tablo 4.37: YSA algoritması holdout metrikleri.....	93
Tablo 4.38: YSA algoritması 5 kat çapraz geçерleme sonuçları.	94
Tablo 4.39: Alvarado karmaşıklık matrisi.	95
Tablo 4.40: Alvarado metrikleri.	96
Tablo 4.41: DVM algoritması holdout yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.	97
Tablo 4.42: DVM yaş+cinsiyet nitelikleri eklendikten sonra karmaşıklık matrisi.....	97
Tablo 4.43: DVM 5-fold yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.	98

Tablo 4.44: RO algoritması holdout yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.	98
Tablo 4.45: RO algoritması 5-fold yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.	98
Tablo 4.46: RO algoritması yaş+ cinsiyet nitelikleri eklenerek özellik önemi.	99
Tablo 4.47: NB algoritması holdout yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.	100
Tablo 4.48: NB algoritması 5 kat çapraz geçерleme yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.	100
Tablo 4.49: K-EY algoritması holdout yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.	100
Tablo 4.50: K-EY algoritması 5-fold yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.	100
Tablo 4.51: YSA algoritması holdout yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.	100
Tablo 4.52: LR algoritması holdout yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.	101
Tablo 4.53: LR algoritması 5-fold yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.	101
Tablo 4.54: LR algoritması yaş+ cinsiyet eklendikten sonra katsayılar.	101
Tablo 4.55: Analizlerin genel performans değерlendirmesi (holdout).	102
Tablo 4.56: Tabakalı 5-kat çapraz geçерleme ile analizlerin performans değерlendirmesi.	104
Tablo 4.57: RO ile niteliklerin modele verdikleri katkılarına göre sıralanması.	105
Tablo 4.58: Lojistik regresyon ile niteliklerin modele verdikleri katkılarına göre sıralanması.	105
Tablo 4.59: Yaş+cinsiyet niteliklerinin eklenmesinden sonraki genel analizler (holdout).	106
Tablo 4.60: Yaş+cinsiyet nitelikleri ile Tabakalı 5-kat çapraz geçерleme genel analizler...	107
Tablo 4.61: Lojistik regresyon ile niteliklerin modele verdikleri katkılarına göre sıralanması.	108

SİMGE VE KISALTMA LİSTESİ

Simgeler	Açıklama
γ	: Gamma
C	: Sınıf Kümesi
M	: Model Kümesi
n	: Veri Setindeki Niteliklerin Sayısı
x	: Veri Seti
y	: Çıktı Uzayı
x_i	: Veri Setinde Yer Alan i. Örnek
x_{ij}	: i. Örneğe ait j. Nitelik Değeri

Kisaltmalar	Açıklama
AA	: Akut Apandisit
DN	: Doğru Negatif
DP	: Doğru Pozitif
DVM	: Destek Vektör Makineleri
RO	: Rasgele Orman
ROC:	: Receiver Operating Characteristic (Türkçesi)
RUS	: Rasgele Az Örnekleme
ROS	: Rasgele Aşırı Örnekleme
SMOTE	: Sentetik Azınlık Aşırı Örnekleme
YN	: Yanlış Negatif
YP	: Yanlış Pozitif
YSA	: Yapay Sinir Ağları

ÖZET

DOKTORA TEZİ

DENGESİZ KLİNİK VERİLER İÇİN KARAR DESTEK ÖNERİSİ: AKUT APANDİSİT ÖRNEĞİ

Kevser ŞAHİNBAŞ

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Enformatik Anabilim Dalı

Danışman : Prof. Dr. Sevinç GÜLSEÇEN

Sağlık alanında karın ağrısı şikâyetiyle hastaneye başvuran hastalarda akut apandisit teşhisin zor tanımlanması sıkça gereksiz akut apandisit ameliyatlarına sebep olmaktadır. Bu çalışmanın amacı karın ağrısıyla hastaneye başvuran hastalarda sınıflandırmaya dayalı makine öğrenmesi algoritmaları aracılığı ile mevcut vakanın ameliyat gerektiren akut apandisit olup olmadığının teşhisini sağlamak. Bu amaca yönelik sağlık alanında sıkça karşılaşılan sınıf dengesizliği probleminin sınıflandırma üzerindeki olumsuz etkisini azaltmak için SMOTE, Random Oversampling ve Random Undersampling yöntemleri önerilmiş ve akut apandisit teşhisinin tahmini için Alvarado Skora ait risk faktörlerinden yararlanılmıştır. Bu amaçla Alvarado skoru ile akut apandisit tanısında altın standart olarak kabul edilen histopatolojik rapor sonucu karşılaştırılmıştır. Sınıflandırma algoritmalarından Destek Vektör Makinesi, Rasgele Orman, Yapay Sinir Ağları, Naive Bayes, k-En Yakın Komşu Algoritması ve Lojistik Regresyon kullanılarak farklı sınıflandırma modelleri oluşturulmuştur. Oluşturulan modellerden elde edilen sonuçlar karşılaştırılarak en iyi performans gösteren sınıflandırma yöntemi önerilmiştir. En iyi performansın tespit edildiği Destek Vektör Makinesi algoritması için arayüz oluşturularak hekimlerin karar verme mekanizmasına katkı sunabilecek Karar Destek Sistemi uygulaması geliştirilmiş ve Alvarado skorlarına yeni bir bakış açısı önerilmiştir.

Ağustos 2019, 155 sayfa.

Anahtar kelimeler: Sınıf dengesizliği, sınıflandırma, alvarado skor.

SUMMARY

Ph.D. THESIS

A DECISION SUPPORT PROPOSAL for IMBALANCED CLINICAL DATA: ACUTE APPENDICITIS CASES

Kevser ŞAHİNBAŞ

İstanbul University

Institute of Graduate Studies in Sciences

Department of Informatics

Supervisor : Prof. Dr. Sevinç GÜLSEÇEN

The difficult diagnosis of acute appendicitis of patients appealing to the hospital with abdominal pain often leads to unnecessary acute appendicitis operations. Accordingly, the aim of this study is to be able to provide the correct diagnosis whether the existing case indeed necessitates operation or not through machine learning algorithms based on classification. To that purpose, SMOTE, Random Oversampling and Random Undersampling methods were proposed to reduce the negative effects of common imbalanced data set problem on classification and it was benefitted from the risk factors in relation to Alvarado Score to predict the diagnosis of acute appendicitis. Therefore, the result of histopathologic report which is accepted as a gold standart for the diagnosis acute appendicitis was also compared with Alvarado Score. Additionally, different classification models were generated by using classification algorithms like Support Vektor Machine, Random Forest, Artificial Neural Network, Naive Bayes, k-Nearest Neighbor and Logistic Regression. Consequently, a decision support system was developed that could contribute to the decision making mechanism by generating interface for the Support Vektor Machine algorithm in which the best performance was obtained.

August 2019, 155 pages.

Keywords: Unbalanced dataset, classification, alvarado score.

1. GİRİŞ

Sağlık alanında karın ağısı şikâyetiyle hastaneye başvuran hastalarda akut apandisit (AA) en sık görülen acil durum olup apendektomi (akut apandisit ameliyatı) tüm dünyada acil yapılan operasyonların ilk sırasında yer almaktadır (Liang ve diğ., 2014, Doğan ve diğ., 2012). Bu nedenle hastanın ameliyat olup olmaması kararı hızlı bir şekilde alınmalıdır. Akut apandisit tanısını koymak oldukça güçtür. Operasyon sonrası materyalin incelenmesi neticesinde belirlenen negatif apendektomi denilen ameliyat sırasında hastaliksız bir appendiks ile karşılaşma oranları nedeniyle onlarca gereksiz operasyona ve hastaların acı çekmesine neden olduğu görülmektedir. Yanlış bir negatif tanı, teşhisi geciktirebilmektedir ve bu nedenle morbidite, mortalite ve yasal sorunların artmasına neden olabilmektedir (Flum ve Koepsell, 2008). Bununla birlikte klinik tanı, deneyimli cerrahlar için bile çoğu zaman zorluk teşkil etmektedir. Negatif apendektomi oranının %20-30'lara ulaşması ne denli zor bir karar olduğunu kanıtlamaktadır (Yang ve diğ., 2006). Akut apandisit klinik tanısının erkeklerde %78, kadınlarda %58 oranında doğru teşhis edildiği bildirilmiştir (Pieper ve diğ., 1982). Apendektomi uygulanan birçok hastanın, apandisit tanısı için altın standart olan cerrahi olarak çıkarılmış appendiksin histopatolojisinde olumsuz olduğu görülmüştür (Munir ve diğ., 2008). Literatürde AA için negatif laparotomi oranının %4-75 arasında değiştiği bildirilmiştir (Eskelinen ve diğ., 1992). Son yıllarda, hastaların ameliyat edilip edilmeme kararı %10-25 arasında negatif apendektomi oranları nedeniyle onlarca gereksiz operasyona neden olan bir zorluk oluşturmaktadır (Hale ve diğ., 1997; Colson ve diğ., 1997). Öte yandan, mevcut istatistikler 5 akut apandisit vakasının 1'inde yanlış tanı olduğunu ve acil apendektomi geçirmiş hastaların %40'ında normal bir appendiksinin olduğunu göstermiştir (Kwok ve diğ., 2004). Diğer bir çalışmada negatif laparotomi ile karşılaşma olasılığı %13-36 arasında bildirilmektedir (Steven ve diğ., 2001; Bergeron, 2006). Teknoloji ve araştırma yöntemlerinde kaydedilen ilerlemelere rağmen, negatif apendektomi oranı %15 ila %50 arasında kalmaktadır (Khan, 2006). Diğer bir çalışmada perfore apandisit oranlarının %3-30 arasında negatif apendektomi oranlarını görmekteyiz (Binnebösel ve diğ., 2009). Günümüz koşullarında oldukça gelişmiş laboratuvar testleri ve görüntüleme tekniklerindeki gelişmelere rağmen negatif apendektomi oranları %14.7 ile %8.47 arasında değiştiği farklı bir çalışmada görülmektedir (Seetahal ve diğ., 2011). Operasyon sonrası materyalin incelenmesi neticesinde belirlenen negatif apendektomi

oranlarının genel olarak %3-4 ila %44 arasında olduđu farklı bir çalışmada belirtilmektedir (Cleeve ve diğ., 2011; Gopal ve Jaffrey, 2011). Kesin tanı konmadan erken yapılan cerrahi müdahaleler özellikle doğurganlık çağındaki kadınlarda olmak üzere %20-30 arasında negatif apendektomi oranı ile sonuçlanabilmektedir (DeKoning, 2016).

Normal bir appendiksin alınması hem hastalar hem de sağlık kaynakları için bir sıkıntı oluşturmaktadır (Khan ve Rehman, 2005). Negatif apendektomi oranlarındaki artış gereksiz morbidite ve komplikasyonlara sebep olabilmekte, tedavi maliyetlerinde artış ile birlikte hekimleri davalarla karşı karşıya bırakabilmektedir (Liang ve diğ., 2014). Bununla birlikte, durumun ve operasyonun erken tanınması, morbidite ve olası mortalite, kalış süresi ve tedavi maliyetini azaltmada en önemli faktörler olmuştur (Limpawattanasiri, 2011).

Akut apandisit; akut karın ağrısı, bağırsak tıkanımları, gastroenterit, akut kolesistit, üreter taşı koliki, pankreatit, divertikülit, üriner sistem enfeksiyonları, sistit, lenfadenopatiler, bağırsak iskemileri gibi birçok hastalığı taklit edebildiği görülmektedir (Humes ve Simpson, 2006).

Söz edilen gerekçeler, negatif apendektomi oranları ve uzman hekimlerle görüşmeler neticesinde bu çalışma ortaya çıkmıştır.

Yanlış teşhis nedeniyle gereksiz yapılan AA operasyonlarını azaltmaya yönelik yapılan bu çalışmadaki hastalara ait veri setleri sağlık alanında sıkça karşılaşılan sınıf dengesizliği problemini de beraberinde getirmektedir.

Gerçek yaşamda özellikle sağlık alanında nadir görülen birçok hastalık teşhisi yapmak için kullanılan gerçek veri setleri mükemmel değildir. Tahmin yapmak için kullanılan bu veriler genellikle dengesizdir. Bu nedenle, son zamanlarda gerçek yaşamda dengesiz veri setleri ile karşılaşılma olasılığı yüksek olması nedeniyle dengesiz veri setlerinin makine yöntemleri ile analizi önemli bir konu haline gelmiştir (Liu ve diğ., 2009). Sınıf dengesizliği veri madenciliği alanındaki on zorlu araştırma problemlerinden biri olarak kabul edilmektedir (Yang ve diğ., 2006). İki sınıflı bir veri setinde bir sınıfın diğer sınıftan çok daha az örneği gözlemlendiğinde veri kümesinin dengesiz olduğu söylenmektedir (Galar ve diğ., 2012). Bir veri setinde daha fazla gözlem sayısına ait sınıfa çoğunluk sınıf, az gözlem sayısına ait sınıf ise azınlık sınıfı olarak adlandırılmaktadır (Chawla ve diğ., 2004). Genellikle azınlık sınıfı çalışma alanında en

çok dikkat çeken durum olmakta ve önemli bir yer teşkil etmektedir; dolayısıyla azınlık sınıfının doğru bir şekilde yüksek tahmini beklenmektedir (Liu ve diğ., 2009). Bu çalışmada, sınıf dengesizliği probleminin sınıflandırma üzerindeki olumsuz etkisini azaltmak için SMOTE, Random Oversampling ve Random Undersampling yöntemleri önerilmiştir. Negatif laparotomi oranını azaltmanın yollarını belirlemek için AA'nın klinik bulguları yanı sıra birçok girişimde de bulunulmuştur. Örneğin WBC sayısı, nötrofil yüzdesi ve CRP (C-reaktif) gibi laboratuvar ölçümleri, AA şüphesi olan hastalarda tanı yardımcıları olarak yaygın şekilde kullanılmaktadır; ancak teşhis değerleri sınırlı olmaktadır (Yang ve diğ., 2006). Negatif apendektomi oranını azaltmak için bazı skoreleme sistemleri oluşturulmuştur. Bunlar klinik semptomları, fiziksel bulguları ve laboratuvar testlerini kapsamaktadır ve görüntüleme testlerinin olmaması ya da onaylanmaması durumunda karar verme sürecine yardımcı olmaktadır (Escribá ve diğ., 2011). Veri setlerindeki dengesizlik problemi azaltılıp veriler dengeli hale getirildikten sonra AA teşhisinin tahmini için Alvarado Skora ait risk faktörlerinden yararlanılmıştır. Bu amaçla Alvarado skoru hesaplanarak akut apandisit tanısında altın standart olarak kabul edilen histopatolojik rapor sonucu ile karşılaştırılmıştır. Yeni gelen bir hasta verisinin hangi sınıfa ait olduğu sınıflandırma algoritmaları aracılığı ile belirlenmektedir.

Sınıflandırma, makine öğrenmesinin önemli bir parçasıdır. Sınıflandırmada veriler eğitim seti ve test seti olmak üzere iki parçaya ayrılır. Model kurma işlemi eğitim setinde yapılır ve holdout, çapraz geçerlilik, bootstrap vb. gibi tekniklerle model sonuçlarının genelleştirilmesi sağlanır. Sonra test seti yardımıyla kurulan modelin performansı bulunur (Zhang ve Mani, 2003). Bu çalışma kapsamında, sınıflandırma algoritmalarından Rasgele Orman, Destek Vektör Makinesi, Yapay Sinir Ağları, Naive Bayes, k-En Yakın Komşu Algoritması, Lojistik Regresyon ve Karar Ağacı kullanılarak farklı sınıflandırma modelleri oluşturulmuştur. Oluşturulan modellerin sonuçları karşılaştırılarak en iyi performansı gösteren sınıflandırma yöntemi önerilmiştir. Dengeleme yöntemleri ve verilerin analizleri için Python kodlama dili kullanılmıştır.

Akut apandisit teşhisine yönelik makine öğrenmesi sınıflandırma algoritmaları aracılığı ile gerçekleştirilen bu çalışmada en iyi performansı gösterdiği tespit edilen Destek Vektör Makinesi algoritması için arayüz oluşturularak hekimlerin karar verme mekanizmasına katkı

sunabilecek Karar Destek Sistemi uygulaması geliştirilmiştir. Hasta verileri girildikten sonra alt yapıda Destek Vektör Makinesi algoritması çalışmakta ve girilen verilere göre müdahale edilmeli ya da gözlemlenmeli sonucu vererek hekime karar verirken yardımcı olmaktadır.

Bu çalışmanın katkıları aşağıda sıralanmıştır:

- Akut Apandisit teşhisi için kullanılan Alvarado Skor ile Makine Öğrenmesi Algoritmaları karşılaştırılmış ve karın ağrısıyla gelen hastaların akut apandisit olup olmadığına karar vermede yardımcı olan en iyi sınıflandırmayı yapan algoritmanın Destek Vektör Makinesi olduğu tespit edilmiştir. Parametre ayarlaması yapılarak ve Alvarado Skorundaki parametrelere ek olarak yaş ve cinsiyet parametreleri de eklenerek 1454 hasta verisi ile %100 doğruluk bulunmuştur.
- Dengesiz veri setlerinde en çok kullanılan yöntemlerden Random undersampling (RUS), Random oversampling (ROS) ve SMOTE algoritmaları kullanılarak veri seti dengeli hale getirilmiştir.
- Bu çalışmada en iyi performansın tespit edildiği Destek Vektör Makinesi algoritması için arayüz oluşturularak hekimlerin karar verme mekanizmasına katkı sunabilecek Karar Destek Sistemi uygulaması geliştirilmiştir.
- Kurallar belirlenmiş olup Alvarado skorunun bu çalışmada tespit edilen yeni kritik değerlerle güncellenmesi gerektiği sonucuna varılmıştır.
- Yaş ve cinsiyet parametreleri de eklenerek bu iki parametrenin akut apandisit teşhisi için önemli olup olmadığının tespiti yapılmıştır.
- Niteliklerin katsayılarına ve niteliklerin önemine bakılarak modele verdikleri katkı sıralanmıştır.

Bu tezin amacı karın ağrısıyla gelen hastanın AA olup olmadığının ayrımını en iyi şekilde yapan ve bu sayede çok sayıdaki hastanın gereksiz ameliyatının önlenmesi sağlayan sınıflandırma algoritmasını tespit etmek ve bu amaca yönelik bir Karar Destek Sistemi oluşturmaktır. Ayrıca sınıf dengesizliği sorunu olduğunda, çözüm algoritmalarının veri madenciliği yöntemlerinin performansları üzerine etkisini incelemektir. Bu amaçlar doğrultusunda tezin kavramsal kısımlarını açıklamak için Genel Kısımlar bölümünde, Dengesiz Veri Setleri, Akut Apandisit, Modelleme bölümünde Sağlık Alanında Kullanılan Sınıflandırma Algoritmaları ve Model Değerlendirme ve Seçimi anlatılmıştır.

Malzeme ve Yöntem bölümünde, tez çalışmasında kullanılan veri setleri tanıtılmış, kullanılan yazılım, Alvarado Skoruna göre seçilen niteliklerden bahsedilmiştir. Bulgular bölümünde veri analizleri, dengesiz veri setlerini dengeli hale getiren yöntemlerin sonuçları, sınıflandırma algoritmalarının sonuçları anlatılmıştır. Sınıflandırmaya dayalı makine öğrenmesi teknikleri karşılaştırılmış en bu veri setine göre en iyi tahminleyici model seçilmiştir. Seçilen en iyi tahminleyici model için bir arayüz tasarlanmıştır. Tartışma ve Sonuç bölümünde tezin tamamının genel değerlendirilmesi yapılmış ve gelecekteki çalışmalar için önerilerde bulunulmuştur.



2. GENEL KISIMLAR

2.1. DENGESİZ VERİ SETLERİ

Bir gruptaki gözlem sayısının diğer gruptakine kıyasla daha az olması olarak tanımlanan sınıf dengesizliği günlük yaşamda sıkça karşılaşılan problemlerden biridir.

Bir veri setinde daha fazla gözlem sayısına ait sınıfa çoğunluk sınıf, az gözlem sınıfına ait sınıf ise azınlık sınıfı olarak adlandırılmaktadır (Chawla ve diğ., 2004). Geleneksel sınıflandırma algoritmaları, sınıf dağılımından bağımsız olarak dengeli sınıf dağılımı olduğunu varsayarak genel doğruluğu en üst düzeye çıkarmak için tasarlanmıştır (He ve Garcia, 2009). Bu nedenle, algoritmaya dengesiz veri setleri gösterildiğinde genellikle çoğunluk sınıfı örnekleri tarafından bastırıldığı için kurulan modelde çoğunluk sınıfı gözlemlerinin etkisi daha fazla olmaktadır (Thai-Nghe ve diğ., 2011). Bu sınıflandırıcıların özellikle azınlık sınıfında fazlaca aşırı uyum ve kötü performans göstermesine neden olur. Ayrıca azınlık sınıfına ait bazı gözlemler de yanlış sınıflandırmalara neden olmaktadır (Krishna, V. ve Sobha R., 2011).

Dengesiz veri setleri; özellikle sağlık bilimlerinde görülen tıbbi teşhis (Mazurowski ve diğ., 2008), dolandırıcılık tespiti (Cieslak ve diğ., 2006), risk yönetimi (Huang ve diğ., 2006) ve kirlilik tespiti (Lu ve Wang, 2008) gibi birçok gerçek dünya olayların sınıflandırma problemlerinde ortaya çıkmaktadır.

2.1.1. Sınıf Dengesizliği Yaklaşımları

Dengesiz veri öğrenme, makine öğrenmesindeki zorlu sorunlardan biridir. Sınıf dengesizliği problemi ile başa çıkmak, dengesizliği azaltmak veya etkisini ortadan kaldırmak için çok sayıda çözüm önerilmiştir (Guo ve diğ., 2017). Sınıf dengesizliği probleminin üstesinden gelmek için kullanılan bazı yeniden örnekleme yöntemleri;

- Az Örnekleme (undersampling)
- Aşırı Örnekleme (oversampling)
- Sentetik (Yapay) Veri Üretme

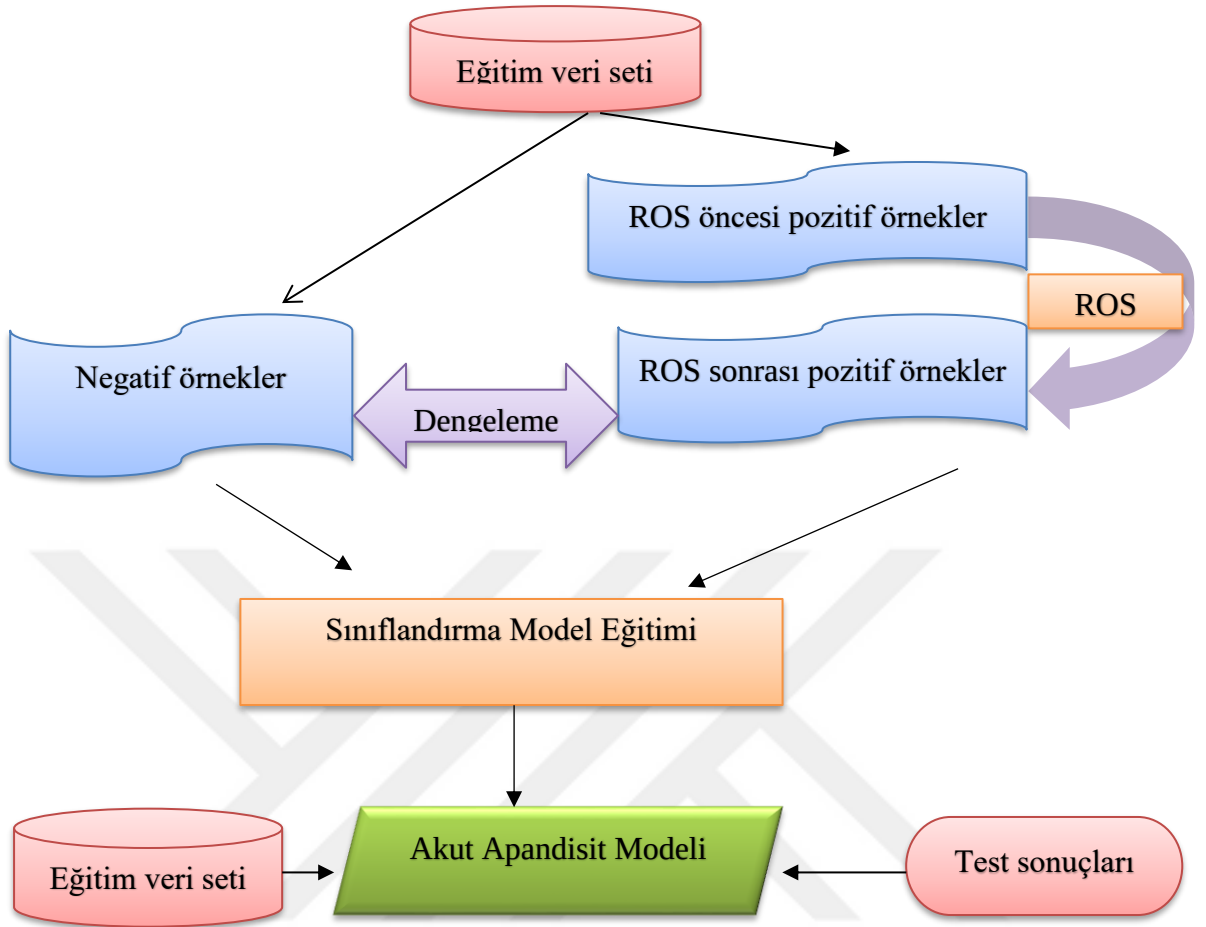
Yeniden örnekleme yöntemleri, eğitim verilerindeki sınıf dağılımını, yeni azınlık örnekleri çoğaltarak veya üretmek (aşırı örnekleme) veya çoğunluk sınıfından (az örnekleme) örnekleri çıkararak dengelemeyi amaçlamaktadır (Thai-Nghe ve diğ., 2010; Blagus ve Lusa, 2017).

Orijinal eğitim setinin 100 pozitif ve 1000 negatif veri grupları içerdiğini varsayalım. Aşırı örneklemede (oversampling) 1000 tane pozitif veri grubu ve 1000 negatif veri grubu içeren bir eğitim seti oluşturmak için azınlık sınıfın kopyası çoğaltılır. Az örneklemede (undersampling) negatif veri gruplarını rasgele ortadan kaldırır, böylece yeni eğitim seti 100 pozitif veri grubu ve 100 negatif veri grubu içerir (Han ve Kamber, 2006).

2.1.1.1. Rasgele Aşırı Örnekleme (ROS)

ROS dengesiz sınıflandırma problemleri ile baş etmenin en basit ve eski yöntemidir. Rasgele aşırı örnekleme (ROS) yönteminde sınıflayıcı istenen sınıf oranına ulaşana kadar eğitirken azınlık sınıf etiketi örneklerini sayıca çok olan sınıf etiketine yaklaştırmak için rasgele çoğaltarak sınıf dağılımını dengelemek için kullanılır (He ve Garcia, 2009; Baesens ve diğ., 2015; Lunardon ve diğ., 2014). S'nin dengesiz veri seti olduğunu, S_{Neg} negatif sınıf (çoğunluk sınıfı) ve S_{Poz} pozitif sınıf (azınlık sınıf) olduklarını varsayalım. ROS'un görevi K örneklerini azınlık/pozitif sınıftan rasgele kopyalamak ve bunları S'ye eklemek, bu da S'deki iki sınıfı aynı örnek sayılarına sahip olmasını sağlamaktadır. Bu yolla, pozitif sınıfın örnek sayısı $S_{Poz} ++K$ 'ya eşittir, farklı K değerleri, eğitim veri seti için farklı denge derecelerine karşılık gelmektedir. $S_{Poz} == S_{Neg}$ olduğu zaman ROS tamamlanmış demektir. ROS algoritması için akış şeması Şekil 2.1'de gösterilmektedir (Sun ve diğ., 2018). ROS etkili bir yöntem olmasına rağmen, azınlık sınıfı örneklerinin tam kopyalarını aldığı için aşırı uyum problemi ortaya çıkabilmektedir (Chawla ve diğ., 2002; Gueo ve diğ., 2008).

Aşırı örnekleme, sınıf dengesizliği derecesini azaltan azınlık sınıfların rastgele yukarı boyutlandırılması anlamına gelir. Azınlık sınıfının iki sınıfın eşit sıklıkta olduğu noktaya kadar çoğaltılmasıyla, aşırı örnekleme yaklaşımı nadiren kullanılır, çünkü modeli azınlık sınıfına doğru yönlendirerek aşırı uydurma riskini artırır. Bu yöntemin diğer dezavantajları, yeni bilgilendirici azınlık örneklerinin eklememesi ve eğitimi yavaşlatmasıdır. Bu, orijinal veri setinin oldukça büyük olması durumunda özellikle etkili olmayabilir (Holte ve Drummond, 2003).



Şekil 2.1: ROS algoritması (Sun ve diğ., 2018).

2.1.1.2. Rasgele Az Örneklemeler (Undersampling)

Rasgele az örneklemede çoğunluk sınıfı örneklerinin rasgele ortadan kaldırılması yoluyla sınıf dağılımını dengelemeyi amaçlayan sezgisel olmayan bir yöntemdir (Baesens ve diğ., 2015). Rasgele az örnekleme yaklaşımı, F negatif örneklerini S değerine düşürür. Farklı F değerleri, RUS işleminden sonra ayarlanan eğitim verileri için farklı denge derecelerine karşılık gelir. $S_{Neg}-F==S_{Poz}$ olduğu zaman RUS işlemi tamamlanmış olur. RUS'un en büyük dezavantajı, potansiyel olarak yararlı örneklerin atılabilmesidir.

Düşük örnekleme, çoğunluk sınıfının, rastgele gözlem bölümünün ortadan kaldırılması yoluyla küçültülmesinden oluşur. Dengesizlikteki problemlerde, çoğunluk sınıfının birçok gözleminin tekrarlandığını ve bunların bir kısmının rastgele ortadan kaldırılması yoluyla, oluşan dağılımının çok farklı olmaması gerektiğini kabul etmek mantıklıdır. Öte yandan, elemanın denetlenmemiş bir şekilde yapılması gerçeğinden dolayı, veri setinden uygun gözlemlerin

kaldırılması riski hala mevcuttur. Bu yöntem, basitliği nedeniyle sıklıkla kullanılır ve öğrenme aşamasının hızını artırır (Holte ve Drummond, 2003) .

2.1.1.3. SMOTE

Azınlık sınıfındaki eğitim verilerinin daha güvenilir sınıflandırma sonucunu edinmek için çoğunluk sınıfına yaklaştırılarak çoğaltılmasına yönelik geliştirilen yöntemlerden biri de SMOTE algoritmasıdır. ROS'un zayıflığının üstesinden gelmek için; Chawla ve arkadaşları tarafından sunulan sentetik azınlık aşırı örnekleme tekniği, basit, etkili ve yaygın olarak kullanılan sınıf dengeleme yöntemidir (Chawla ve diğ., 2002). SMOTE mevcut azınlık örnekleri arasında araya girerek yeni yapay azınlık örnekleri oluşturur. Temel varsayım, birbirine yakın iki gerçek pozitif örnek arasında yapay bir pozitif örnek olduğudur. Bu nedenle, SMOTE algoritması birbirine yakın olan iki gerçek pozitif arasında yapay olarak yeni bir potansiyel örnek bulmaya çalışan sınıf dengeleme yöntemidir (Chawla ve diğ., 2002). Aşırı örneklemeden sonra pozitif örnek sayısının orijinal örnek sayısının $(m++1)$ katı olması gerektiğini varsayalım. Her pozitif örnek x^{Poz}_i ($i \in [1, 2, \dots, S_{Poz}]$ için, $x^{Poz-Near}_{ik}$ ($k=1, 2, \dots, m$) olarak gösterilen en yakın pozitif örnek komşularını bulması gerekir. Daha sonra, m yeni pozitif örnekleri, aşağıdaki formüle göre orijinal pozitif örnekler üretebilir. Son olarak, SMOTE algoritması tarafından yapay olarak yeni pozitif örneklerin sayısı $m \times S_{Poz}$ (Sun ve diğ., 2018).

$$x^{Poz-SMOTE}_{ik} = x^{Poz}_i + \text{rand}(0,1) \times (x^{Poz-Near}_{ik} - x^{Poz}_i) \quad (i \in [1, 2, \dots, m]) \quad (2.1)$$

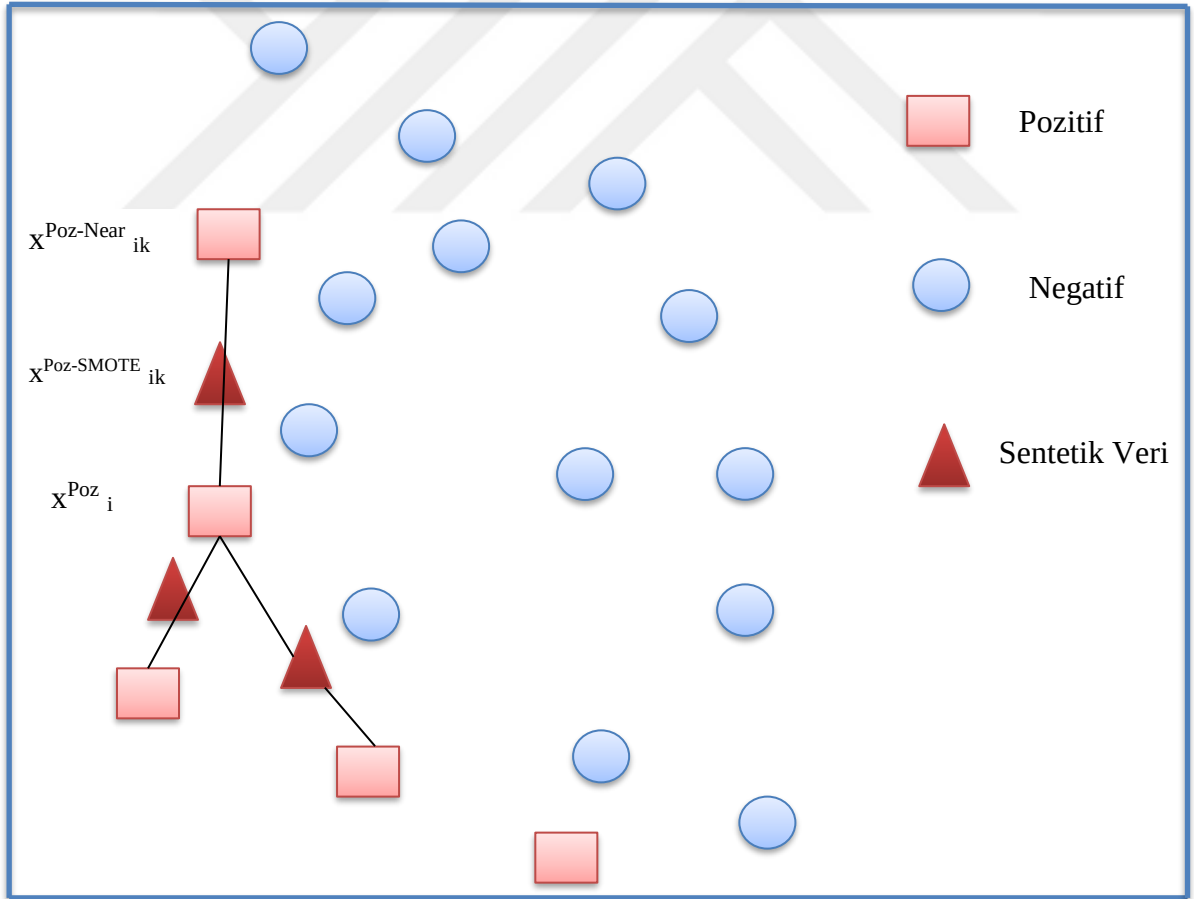
Yukarıdaki formül (2.1)'de, $\text{rand}(0,1)$ fonksiyonu 0 ile 1 arasında rasgele değerler üretir. Yeni çoğaltılmış pozitif örnekler ve orijinal pozitif örnekler, eğitim pozitif örnekleri olarak birlikte kullanılacaktır. Yapay olarak çoğaltılan pozitif örneklerin sayısı, m 'nin farklı değerlerine göre değişiklik gösterir; bu son eğitim veri setinde pozitif sınıf ile negatif sınıf arasında denge derecelerine yol açar. Şekil 2.2, SMOTE algoritmasının temel prensiplerini göstermektedir. ROS ile karşılaştırıldığında SMOTE algoritması birbirinden farklı ve orijinal pozitif örnekleri hiç kopyalanmayan yeni pozitif örnekler üretir. Bu, ROS'un neden olduğu aşırı uyum sorununu etkili bir şekilde önleyebilir. Ayrıca, çok karmaşık değildir.

SMOTE Algoritması aşağıdaki gibi tanımlanabilir: (Chawla ve diğ., 2002)

- 1.Adım: Azınlık sınıfına ait her bir gözlemin k yakın komşusu aranır.
- 2.Adım: Azınlık sınıfına ait gözlem ile k yakın komşusu olan gözlemin farkı alınır.
- 3.Adım: (0,1) arasında rasgele bir sayı seçilir.
- 4.Adım'da bulunan fark ile bu sayı çarpılır.
4. Adım: Eşitlik 1.1 kullanılarak yeni sentetik gözlem elde edilir.

$$X_{\text{yeni}} = X_i + (X_j - X_i) * \alpha$$

- 5.Adım: İstenen sayıda sentetik gözlem oluşturmak için Adım 1-4 yinelenir.



Şekil 2.2: SMOTE algoritması (Sun ve diğ., 2018).

```

SMOTE (S,M,k)
Giriş: Az sayıdaki sınıfa ait örnekler S, Smote oranı M, komşuluk k
Çıkış:  $(M/100)*S$ 
if  $M < 100$ 
    then S grubunu rasgele seç,
         $S = (M/100)*S$ 
         $M = 100$ 
    end if
Ornek[][]: ilk küçük sınıf grubu
uretilen: uretilen örnekler
sanal[][]: sanal örnekler
for i:1 to S
    tüm i elemanları için en yakın k hesapla ndizi'ye ekle
    Türet(M,i,ndizi)
endfor
Türet(M,i,ndizi){
while  $M \neq 0$ 
    for özellik=1:özellikSayisi
        Hesapla: fark=Ornek[ndizi[adet][özellikSayisi]-Ornek[i]
        sanal[uretilen][özellik]=ornek[i][özellik]+fark
    endfor
    uretilen++
     $M=M-1$ 
endwhile
return
}

```

Şekil 2.3: SMOTE algoritması adımları (Sun ve diğ., 2018).

2.2. AKUT APANDİSİT

Acil servislere başvuruların %10'u karın ağrısı nedeniyledir (Young, 2014). Karın ağrısı ile başvurularda saptanan en sık cerrahi neden ise akut apandisit (Kavic, 2010; Liang ve diğ., 2014; Reynolds ve Jaffe, 1992; Scholer ve diğ., 1996). Günümüzde AA en sık görülen cerrahi acillerden biridir ve apendektomi dünya çapında en sık yapılan acil operasyonlar arasında yer almaktadır (Smith, 1965; Williams, 1983; Cusheiri ve diğ., 2003; Paterson, 2003; Liang ve diğ., 2014). AA dünyadaki en yaygın cerrahi acil hastalıklardan biri olduğu için genel nüfusun %7 ila %12'sini etkilemektedir (Chalazonitis ve diğ., 2008).

Lokale peritonit ile ilişkili ortak fiziksel bulguları ortaya çıkarmak için klinik incelemeyle birlikte yeterli bir tıbbi öykü akut apandisit tanısını koymak için genellikle yeterlidir. Bununla birlikte, özellikle kadın hastalarda apandisit teşhisi her zaman açık değildir çünkü jinekolojik patoloji akut apandisiti taklit edebilir. Ayrıca, gizli veya retrokal apandisit gibi apendiküler

konumlardaki deęişkenlik, hastaların akut apandisit tanısını desteklemek için yeterli periton belirtileri göstermesine izin vermeyebilir (Guidry ve Poole, 1994).

Dünyada akut apandisit, 7’de 1 yaşam boyu riski olan yaygın bir cerrahi acil durumdur (Hanif ve dię., 2010; Mardan ve dię., 2007; Owen ve dię., 1992). Bu bireylerin %6’sının yaşamları boyunca bir atak geçirdięi anlamına gelmektedir (Kanumba ve dię., 2011). Bu durumun özellikle klasik belirti ve semptomları genellikle erken aşamalarda teşhis edilmesi zordur (Singh ve dię., 2008). Farklı hastalık süreçleri, özellikle kadın hastalarda sağ iliak fossada ağrıya neden olan çok sayıda parametre olduğundan akut apandisit tanısını taklit etmektedir (Yegane ve dię., 2008). Apendektomi uygulanan birçok hastanın, apandisit tanısı için altın standart olan cerrahi olarak çıkarılmış appendiksin histopatolojisinde olumsuz olduğu görülmüştür (Munir ve dię., 2008). Öte yandan, mevcut istatistikler 5 akut apandisit vakasının 1’inde yanlış tanı olduğunu ve acil apendektomi geçirmiş hastaların %40’ında normal bir appendiksinin olduğunu göstermiştir (Kwok ve dię., 2004). Normal bir appendiksin alınması hem hastalar hem de sağlık kaynakları için bir sıkıntıdır (Khan ve Rehman, 2005). Bununla birlikte, durumun ve erken operasyonun erken tanınması, morbidite ve olası mortalite, kalış süresi ve tedavi maliyetini azaltmada en önemli faktörler olmuştur (Limpawattanasiri, 2011).

Bugüne kadar güvenilir spesifik akut apandisit belirteci henüz tanımlanmamıştır. Teknoloji ve araştırma yöntemlerinde kaydedilen ilerlemelere rağmen, negatif apendektomi oranı %15 ila %50 arasında kalmaktadır (Khan, 2006).

Akut apandisit, lokalize iskemiye, nekroz ve perforasyona yol açan apendiks duvarının bir bölümünde görülen akut iltihabi bir durumdur. Oluşan iltihap ve nekroz nedeniyle, apendiks, karmaşık apandisit olarak kabul edilen perforasyon ve lokalize apse orflegmon oluşumu riski altındadır (Bhangu ve dię., 2015). Apendiks vermiformis çekumunun tabanında, ileoçekal valfin 2-3 cm distalinde tenya kolilerin birleşim noktasında yer alır (Birnbbaum ve Wilson, 2000). Tenya koliler apendiksin longitudinal kasları olarak devam eder (Turhan ve Kapan, 2000). Bu sayede operasyon sırasında apendiksi bulmak kolaylaşır (Jaffe ve Berger, 2010). Apendiksin konumu klinik prezentasyon farklılığının ve tanıda yanlışlığının en önemli nedenlerinden birisidir (Ghorbani ve dię., 2014). Uzanımı farklılık gösterse de apendiksin çekum ile bağlantı noktası deęişmez (Turhan ve Kapan, 2000). Apendiksin uzanımına göre en sık görülen anatomik pozisyonlar sırasıyla retroçekal, pelvik, subçekal, ileoçekal, paraçekal ve promontarik yerleşimler şeklinde görülür (Deshmukh ve dię., 2014). Apendisk boyutları ile

ilgili olarak da pek çok çalışma yapılmıştır. Ortalama boyu 6-9 cm civarında olmakla beraber 1-30 cm arasında değişebileceği belirtilmiştir (Jaffe ve Berger, 2010; Ghorbani ve diğ., 2014). Bugüne kadar ölçülen en uzun apendiks 33 cm'dir (Coyne, 2007).

Akut apandisit her yaş grubunda saptanabilmekle beraber sıklıkla birinci ve üçüncü dekatlar arasında görülmektedir (Kavic, 2010; Williams, 1983). Hayat boyunca apandisit geçirme riski ortalama %7'dir (Kavic, 2010). Bu oran kadınlarda %6.7 ve erkeklerde %8.6 olarak bildirilmektedir (Kavic, 2010; Williams, 1983; Ellis, 1986; Mishra ve diğ., 2013). Apendektomi geçirme oranları ise erkeklerde %12, kadınlarda ise %23.1'dir (Mishra ve diğ., 2013). Erkeklerde akut apandisit sıklığının fazla olmasına rağmen kadınlarda daha çok apendektomi yapılmasının sebebi jinekolojik patolojilerin akut apandisiti taklit etmesi olarak açıklanmaktadır (Turhan ve Kapan, 2000). Apendektominin tanımlandığı günden bu yana mortalitesi %67'den %1'in altındaki değerlere gerilemiştir (Williams, 1983). Laparoskopik cerrahinin gelişimi ise özellikle morbidite üzerinde etkili olmuştur (Andersson, 2007). Mortalite ve morbidite üzerindeki gelişmelere rağmen aynı başarı tanı konulmasında sağlanamamıştır (Liang, Andersson, Jaffe, Berger, 2014).

Akut apandisit kliniği ve fizik muayenesi her hastada farklı şekillerde ortaya çıkabilmektedir. Akut apandisit tanısında anamnez ve fizik muayenenin yeri tartışılmazdır. Tanı öykü, klinik muayene ve rutin laboratuvar testleri ile konmaktadır. Akut apandisitinin klasik prezentasyonu kolayca teşhis edilir ve tedavi edilir. Bununla birlikte, erken evrelerdeki ince klinik özellikler ve hastalığın diğer hastalıklarla karışabilmesi deneyimli cerrahlar için bile teşhisi zorlaştırabilir (Singh, Gupta ve Parga, 2008). Sağlıksız bir apendiksinin derhal apendektomisi, tedavi edilmeyen apandisitinin 24-48 saat içinde ortaya çıkabilecek perforasyon ve kangrenden ölüm oranını azaltmaktadır (O'Connel, 2008).

Akut apandisitinin erken teşhisinde, teşhis doğruluğunu arttırmak ve böylece negatif apendektomi oranını azaltmak için çeşitli skrolama sistemleri, görüntüleme yöntemleri ve yeni teknikler geliştirilmiştir. Bu sistemler akut apandisit ve spesifik olmayan karın ağrısı arasında ayırım yapmak için değerli ve geçerli araçlardır (Singh ve diğ., 2008). Örnek olarak, histopatoloji, fizik muayene ve birkaç laboratuvar araştırmasına dayanan ve uygulaması çok kolay olan Alvarado skrolama sistemi verilebilir (Limpawattanasiri, 2011). Cerrahlar tarafından kullanılan basit bir skrolama sistemi Alvarado skrolama sistemidir (Kanumba ve diğ., 2011).

Alvarado skoru 1986'da tanıtılmış ve akut apandisit tanısında önemli bulunan sekiz parametre göz önünde bulundurulmuştur. Bu parametreler; sağ alt kadranda lokalize hassasiyet, lökositoz, ağrının göç etmesi, lökosit sola kayması, sıcaklık yükselmesi, bulantı-kusma, anoreksi (iştahsızlık) ve rebound ağrısıdır (Alvarado, 1986).

Tarihsel olarak, klinik değerlendirme AA'nın teşhisinde ve yönetiminde her zaman hayati bir rol oynamıştır ve bugün bile tanıların temel taşı olmaya devam etmektedir (Bhangu ve diğ., 2015). Özellikle genç kadınlarda ve ileri yaşlarda, AA'nın doğru klinik tanısı şeklinde gelen bilinen hastalıklar vardır. AA'nın klinik bir varlık olarak tanınmasından bu yana, klinisyenler iki önemli fakat karşılıklı önlemleri dengeleme ikilemiyle karşı karşıya kalmaktadırlar; perforasyon oranı ve negatif apendektomi oranı. Alfredo Alvarado, AA'nın erken teşhisi için klinik ve belirli laboratuvar parametrelerini alarak pratik bir skor geliştiren ilk kişilerden biridir ve tek başına perforasyon oranını orantılı olarak arttırmadan negatif apendektomi oranını düşürmeyi amaçlamaktadır (Alvarado, 1986). O zamandan beri, AA'nın tanısallı doğruluğunu arttırmak için bir dizi puanlama sistemi geliştirilmiştir ve halen geliştirilmektedir (Sammalkorpi ve diğ., 2016). Bu skrolama sistemlerinin çoğu, AA tanısı için klinik ve çeşitli laboratuvar parametrelerini kullanır. Alvarado skrolaması son otuz yılda yoğun bir şekilde kullanılmış ve test edilmiştir. Alvarado skrolaması, AA'nın tanısı için yararlı bir klinik patolojik araçtır, ancak skrolama sisteminin birçok bileşeni spesifik değildir. Zamanında uygun tedavinin başlatılmasıyla hızlı ve doğru tanı AA'nın tanısı için temel taş olmaya devam etmektedir. AA'nın klinik tanı konusuna öykü ve fizik muayene bulguları temelinde, laboratuvar belirteçleri olsun ya da olmasın, yalnızca farklı ülkeler arasında değil, aynı ülke içindeki kurumlar arasında da büyük farklılıklar göstermektedir.

Alvarado skrolama sistemi üç semptom, üç belirti ve iki laboratuvar sonucuna dayanmaktadır (Alvarado, 1986).

Tablo 2.1: Akut apandisit için alvarado skorlaması tablosu.

		Alvarado Skor
Semptomlar	Ağrının sağ alt kadrana göçü	1
	Bulantı/kusma	1
	İştahsızlık	1
Belirtiler	Sağ alt kadranda hassasiyet	2
	Ateş yüksekliği	1
	Rebound ağrı	1
Laboratuar bulguları	Lökositoz	2
	Sola nötrofil hareketi (>75%)	1
Toplam skor		10

Öneriler**Alvarado Skor:**

0-4= düşük,

5-6= apandisit mümkün ve gözlem tavsiye edilir,

7-8= AA olasılığı ve ameliyat tavsiye edilir,

9-10= yüksek olasılıkla AA ve ameliyat tavsiye edilir.

AA'nın klinik şüphesi genellikle daha fazla uygun doğrulama testine ihtiyaç duymaktadır. AA tanısının doğruluğunu artırma stratejileri arasında aktif gözlem, klinik skorlama sistemleri, abdominal bilgisayarlı tomografi (BT), ultrasonografi (US) ve tanı amaçlı laparoskopi bulunur. Aktif gözlem yaygındır (Jones, 2001). Perforasyon riski göz önüne alındığında, çalışmaya karar verecek olan eşik değeri genellikle düşüktür. Alvarado skoru % 64 duyarlılık ve % 84 özgüllük ile ortak bir klinik skorlama sistemidir (Alvarado, 1986; Ohmann ve diğ., 1995). Abdominal BT, doğruluğu % 98,7'ye yükseltebilir (Birnbaum ve Jeffrey, 1998), ancak nispeten yüksek maliyetli bir muayenedir ve her zaman kullanılamaz. US faydalı bir araçtır, ancak doğruluğu operatöre bağlıdır (Birnbaum ve Jeffrey, 1998; Rioux, 1992). Tanısal laparoskopi negatif apendektomi oranını azaltabilir (McGreevy ve diğ., 2002).

2.3. MAKİNE ÖĞRENMESİ

Makine öğrenmesi etkili ve hatasız tahmin yapabilen algoritmaların tasarlanmasından oluşmaktadır (Mohri ve diğ., 2012). Makine öğrenmesi alanı deneyim ile otomatik olarak gelişen bilgisayar programlarının nasıl oluşturulabileceği ile ilgilenmektedir (Mitchell, 1997). Makine öğrenmesi, performans optimizasyonu için örnek veri ya da geçmiş deneyimleri

kullanarak bilgisayar programlamaktır (Alpaydın, 2014). Bir makine, performansını deneyimler ile arttırdığında, o makinenin öğrenmiş olduğu kabul edilmektedir ki burada sözü edilen öğrenme, veriyi tutan ve ilginç ya da faydalı örüntüleri ortaya çıkaran algoritma ve programlara gereksinim duymaktadır (Harrington, 2012). Makine öğrenmesi, veriden kural öğrenebilen, değişikliklere uyum sağlayabilen ve deneyim ile performansını geliştirebilen yazılımları tasarlamakla ilgilenmektedir (Blum, 2007). Makine öğrenmesi sistemleri veriden programları otomatik bir şekilde öğrenmektedir (Domingos, 2012).

Makine öğrenmesi, tıbbî teşhis, optik karakter algılama, yüz tanıma, spam e-posta filtrelemesi, konuşma dili anlama, , müşteri segmentasyonu, sahtekârlık tespiti ve hava durumu tahmini gibi birçok farklı problemin çözümünde kullanılmaktadır (Schapire, 2008).

Makine öğrenimi, bilgisayarların verilere dayanarak nasıl öğrenebileceklerini ve nasıl performanslarını artıracabileceklerini araştırmaktadır. Ana araştırma alanı bilgisayar programlarının karmaşık kalıpları tanımayı ve verilere dayanarak akıllı kararlar almayı otomatik olarak öğrenmesini sağlamaktır. Örneğin, tipik bir makine öğrenme sorunu, bir bilgisayarı bir dizi programdan öğrendikten sonra postadaki el yazısı posta kodlarını otomatik olarak tanıyabilmesi için programlamaktır. Makine öğrenmesi hızlı büyüyen bir disiplindir.

Denetimli öğrenme temel olarak sınıflandırma ile eş anlamlıdır. Öğrenmedeki denetim, eğitim veri setindeki etiketli örneklerden gelir. Örneğin, posta kodu tanıma probleminde, bir dizi el yazısı posta kodu görüntüsü ve bunlara karşılık gelen makineyle okunabilen çeviriler, sınıflandırma modelinin öğrenilmesini denetleyen eğitim örnekleri olarak kullanılır.

Denetimsiz öğrenme temelde kümelenme ile eş anlamlıdır. Girdi örnekleri sınıf etiketli olmadığından öğrenme süreci denetlenmez. Genellikle, veri içindeki sınıfları keşfetmek için kümelemeyi kullanabiliriz. Örneğin, denetlenmeyen bir öğrenme yöntemi, girdi olarak, el yazısı rakamların bir dizi görüntüsünü alabilir. 10 veri kümesi bulunduğunu varsayalım. Bu kümeler, sırasıyla, 0 ila 9 arasındaki 10 ayrı basamağa karşılık gelebilir. Bununla birlikte, eğitim verileri etiketlenmediğinden, öğrenilen model bize bulunan kümelerin anlamsal anlamını söyleyemez. Yarı denetimli öğrenme, bir model öğrenirken hem etiketli hem de etiketlenmemiş örneklerden faydalanan bir makine öğrenme teknikleri sınıfıdır. Bir yaklaşımda, etiketli örnekler sınıf modellerini öğrenmek için kullanılır ve etiketlenmemiş örnekler sınıflar arasındaki sınırları iyileştirmek için kullanılır. İki sınıflı bir problem için, bir

sınıfa ait örnekler kümesini pozitif örnekler olarak ve diğer sınıfa ait örnekleri negatif örnekler olarak düşünebiliriz.

Aktif öğrenme, kullanıcıların öğrenme sürecinde aktif bir rol oynamasını sağlayan bir makine öğrenme yaklaşımıdır. Aktif bir öğrenme yaklaşımı, bir kullanıcıdan örneğin, bir etki alanı uzmanından bir etiketlenmemiş örnek setinden veya bir öğrenme programı tarafından sentezlenebilecek bir örneği etiketlemesini isteyebilir. Amaç, insan örneklerinden aktif olarak bilgi alarak model kalitesini optimize etmektir, kaç örnek vermeleri gerektiği konusunda bir kısıtlama getirilmektedir.

Veri madenciliği ile makine öğrenmesi arasında birçok benzerlik olduğunu görebilmekteyiz. Sınıflandırma ve kümeleme görevleri için, makine öğrenmesi araştırması genellikle modelin doğruluğuna odaklanır. Doğruluğun yanı sıra, veri madenciliği araştırması, büyük veri setlerindeki madencilik yöntemlerinin verimliliğine ve ölçeklenebilirliğine ve ayrıca karmaşık veri türlerini ele almanın ve yeni, alternatif yöntemleri keşfetmenin yollarına büyük önem vermektedir (Han, Kamber, 2006).

Makine öğrenimi, modern klinik karar vermede oldukça katkıda bulunmuştur. Rastgele Ormanlar, Destek Vektör Makineleri ve Yapay Sinir Ağları en önemli modellerdendir ve bu algoritmalar karmaşık klinik durumlarda kullanılabilecek denetimli öğrenme sınıflandırıcılarıdır (Hsieh ve diğ., 2011).

Akut apandisit tanısı ilginç bir sorundur, çünkü güçtür ve karar hızlı bir şekilde alınmalıdır. Preoperatif akut apandisit klinik tanısının erkeklerde %78, kadınlarda %58 oranında doğru olduğu bildirilmiştir (Pieper ve diğ., 1982). Literatürde akut apandisit için negatif laparotomi oranının % 4 ila 75 arasında değiştiği bildirilmiştir (Eskelinen ve diğ., 1992).

Bu problem birçok araştırmacının oldukça dikkatini çekti ve birçok bilgisayar sistemi tanı prosedürüne yardımcı olmak için çalışmalar yapılmıştır (deDombal ve diğ., 1972; Phillips ve diğ., 1991; Eskelinen ve diğ., 1992; Todd ve Stamper, 1994) Bilgisayar destekli tanının erken sonuçları % 90'ın üzerinde genel bir tanı doğruluğunu tespit etmiştir (de Dombal ve diğ., 1972). Ancak yakın zamanda %74'lük bir doğruluk bulunmuştur (Adams ve diğ., 1986), negatif appendektomi oranlarında % 30'luk bir azalma bildirildiği bildirilmiştir (Fenyö, 1987). Ayrıca bazı sinir ağı algoritmaları oldukça başarılı bir şekilde kullanılmıştır (Eberhart ve diğ., 1991; Pesonen ve diğ., 1994).

Breiman tarafından 1999 yılında önerilen Rastgele Orman, bir grup sınıflandırma algoritması olan karar ağaçları koleksiyonundan oluşur (Breiman, 2001). Her bir ağaç “torbalama” denilen teknikler ve rastgele girdi değişkenleri seçimi ile bağımsız olarak oluşturulmuştur. Algoritma tüm ağaçların sınıflandırılmasının çoğunluğuna dayanmaktadır.

Yapay Sinir Ağları son 40 yıldır araştırma alanı olmuştur (Dayhoff ve DeLeo, 2001). İlk hesaplama, eğitilebilir sinir ağları 1959'da Rosenblatt (Rosenblatt, 1962), Widrow ve Hoff (Widrow ve Hoff, 1960) ve Widrow and Stearns (Widrow ve Stearns, 2003) tarafından geliştirilmiştir. İlk YSA'lar basit doğrusal problemlerin çözümünde sınırlı algılayıcılardı. YSA'nın tıpta kullanımı oldukça yeni bir fenomen olmasına rağmen, tanı, görüntüleme, farmakoloji, patoloji ve elbette prognoz alanında olduğu gibi birçok uygulama bulunmaktadır. Apandisit, bel ağrısı, demans, miyokard enfeksiyonu, psikiyatrik bozukluklar, akut pulmoner emboli ve temporal arterlerin tanısında YSA kullanılmıştır (Wei ve diğ., 1998). Yapay sinir ağları birçok teşhis probleminde oldukça başarılı bir şekilde kullanılmıştır (Bounds ve Lloyd, 1990; Macklin ve 1991; Akay, 1992). Yapay sinir ağları biyolojik sinir ağlarından ilham alarak hesaplama modelleridir; teknik, doğrusal olmayan ilişkilerin modellenmesinde güçlü olarak kabul edilir ve tıbbi teşhis ve öngörmede gittikçe artan bir şekilde uygulanmaktadır (Drew ve Monson, 2000; Cross ve diğ., 1995; Baxt, 1995; Lin ve diğ., 2008; Li ve diğ., 2000). Tipik bir sinir ağı, algılayıcı olarak adlandırılan bir dizi işlem biriminden oluşur. Tıpta, en yaygın sinir ağı türü, bir giriş katmanı, birkaç gizli katman ve bir çıkış katmanından oluşan çok katmanlı bir algılayıcıdır. Her katman birkaç algılayıcıdan oluşur. Katmanlar arasındaki algılayıcılar, farklı ağırlıklarla ağır şekilde birbirine bağlıdır. Ağırlıklar, eğitim algoritmalarıyla ayarlanmaktadır. Tekrarlanan eğitim döngüsünden sonra, insanın deneyimden öğrenmesine benzer şekilde, ağırlıklar en iyi tahmin yeteneğini elde etmek için ayarlanmaktadır. Kalp krizi erken teşhis ve tedavisi, göğüs kanseri erken teşhis ve tedavisi, ilaç etkileri analizi, kan analizi sınıflandırma, EEG, ECG, MR ve kalite artırımı gibi sağlık alanlarında Yapay Sinir Ağları kullanılmaktadır. Yapay Sinir Ağları birbirine bağlı hesaplama öğelerinin, nöronların karmaşık bir yapısı ile karakterize edilen doğrusal olmayan bir matematiksel model sınıfıdır. Bu hesaplama öğeleri bir toplama işlemi kullanarak bir dizi girdiyi toplar ve AA'nın ciddiyeti gibi bir çıktı üretir (Sargent, 2001).

Bazı yazarlar, AA'yı farklı derecelerde başarı ile teşhis etmek için öngörücü modeller geliştirmek için yapay sinir ağları kullanmışlardır (Eberhart ve diğ, 1991; Sakai ve diğ., 2007;

Prabhudesai ve diğ., 2008). Prabhudesai ve arkadaşları AA'yı teşhis etmek için %100 duyarlılık, %97 özgüllük, % 96 Pozitif Tahmin Oranı gibi yüksek doğrulukta sonuçlar ile yapay sinir ağları modellerini kullanmış ve % 96 Negatif Tahmin Oranı tespit etmişlerdir (Prabhudesai ve diğ., 2008). Öte yandan, Hsieh ve arkadaşları % 94 duyarlılık ve % 85 özgüllük oranları ile YSA'nın daha az başarılı olduğunu göstermiştir (Hsieh ve diğ., 2011). Sakai ve arkadaşlarının çalışmasında, AUC analizi, YSA'ların orta derecede bir sonuç verdiğini göstermiştir (Sakai ve diğ., 2007).

Lojistik regresyon, ikili öğrenme için yaygın olarak kullanılan standart bir regresyon modelidir, genellikle makine öğrenim çalışmalarında karşılaştırma yapmak için kullanılmaktadır (Dreiseitl ve Ohno-Machado, 2002).

2.4. SINIFLANDIRMA

Veri sınıflandırma, bir öğrenme aşamasından (bir sınıflandırma modelinin yapıldığı yer) ve bir sınıflandırma aşamasından (verilen verinin sınıf etiketlerini tahmin etmek için kullanılan model) oluşan iki aşamalı bir süreçtir. İlk adımda, önceden belirlenmiş bir veri sınıfı veya kavram kümesini tanımlayan bir sınıflandırıcı oluşturulmaktadır. Bu, bir sınıflandırma algoritmasının veri tabanı tablolarından oluşan bir eğitim setini ve bunlarla ilişkili sınıf etiketlerini analiz ederek veya “öğrenerek” sınıflandırıcı oluşturduğu öğrenme aşamasıdır (veya eğitim aşaması). Bir bağlantı, X , bir n boyutlu öznitelik vektörü ile temsil edilir, $X = (x_1, x_n, \dots, x_n)$, sırasıyla n veritabanı özniteliklerinden bağlanmada yapılan n ölçümlerini gösteren, $A_1, A_2, \dots, A_n$ bir. Her bir veri grubu, X , sınıf etiketi niteliği adı verilen başka bir veritabanı niteliği tarafından belirlenen önceden tanımlanmış bir sınıfa ait olduğu varsayılır. Sınıf etiketi niteliği kesikli ve sıralı değildir. Her değer bir kategori veya sınıf olarak işlev görmesi kategoriktir (veya nominaldir). Eğitim setini oluşturan ayrı ayrı veri seti, eğitim veri seti olarak adlandırılır ve analiz edilen veri tabanından rastgele örneklenir. Sınıflandırma bağlamında, veri grupları örnek, nitelik, veri noktası veya nesne olarak adlandırılabilir.

Her eğitim grubunun sınıf etiketi sağlandığı için, bu adım aynı zamanda denetimli öğrenme olarak da bilinir (yani, sınıflandırıcının öğrenilmesi, her eğitim grubunun hangi sınıfa ait olduğu söylendiğinden “denetlenir”). Her eğitim hattının sınıf etiketinin bilinmediği ve öğrenilmesi gereken sınıfların sayısı veya kümesinin önceden bilinemeyeceği denetlenmemiş öğrenme (veya kümelenme) ile çelişir. Örneğin, eğitim seti için kredi karar verisi elimizde

bulunmuyorsa, kredi başvuru verilerindeki risk gruplarına karşılık gelebilecek “benzer tekil gruplarını” belirlemeye çalışmak için kümelemeyi kullanabilir.

Sınıflandırma sürecinin bu ilk adımı, belirli bir X grubunun ilgili sınıf etiketini y tahmin edebilen bir haritalandırma veya fonksiyonun, $y = f(x)$ öğrenilmesi olarak da görülebilir. Tipik olarak, bu haritalama sınıflandırma kuralları, karar ağaçları veya matematiksel formüller şeklinde gösterilir. Kurallar gelecekteki veri dosyalarının sınıflandırılması ve veri içeriğinin daha iyi anlaşılması için kullanılabilir. Ayrıca sıkıştırılmış veri sunumu sağlarlar (Han ve Kamber, 2006).

2.5. MAKİNE ÖĞRENMESİ SÜRECİ ADIMLARI

Günümüzün gerçek dünyadaki veritabanları, tipik büyüklükleri (genellikle birkaç gigabayt veya daha fazla) ve bunların çoklu, heterojen kaynaklardan gelmesi nedeniyle gürültülü, eksik ve tutarsız verilere karşı oldukça hassastır. Düşük kaliteli veriler düşük kaliteli madencilik sonuçlarına yol açacaktır. Veri analizine geçmeden önce verinin analizler için hazır hale getirilmesi önemlidir. Birkaç veri işleme tekniği vardır. Veri temizleme, gürültüyü gidermek ve verilerdeki tutarsızlıkları düzeltmek için uygulanabilir.

Veri temizleme rutinleri, eksik değerleri doldurarak, gürültülü verileri düzgünleştirerek, aykırı değerleri belirleyerek veya kaldırarak ve tutarsızlıkları gidererek verileri “temizlemeye” çalışır. Kullanıcılar verilerin gürültülü olduğuna inanıyorlarsa, uygulanan herhangi bir veri madenciliği sonucuna güvenmeleri pek mümkün değildir. Ayrıca, gürültülü veriler madencilik prosedürü için karışıklığa neden olabilir ve bu da güvenilir çıkıtlara yol açabilir. Çoğu madencilik rutininin eksik veya gürültülü verilerle uğraşmak için bazı prosedürleri olmasına rağmen, her zaman sağlam değildir. Bunun yerine, verilerin modellenmekte olan işleve fazladan sığdırılmaması üzerine konsantre olabilirler. Bu nedenle yararlı bir ön işleme adımı, verilerinizi bazı veri temizleme yordamları üzerinden çalıştırmaktır.

Veri entegrasyonu, birden fazla kaynaktan gelen verileri bir veri ambarı gibi tutarlı bir veri deposunda birleştirir. Veri küçültme, veri toplama özelliğini toplu toplama, gereksiz özellikleri ortadan kaldırma veya kümeleme yoluyla azaltabilir. Verilerin 0.0 ila 1.0 gibi daha küçük bir aralıkta düşmesi için ölçeklendirildiği veri dönüşümleri (örneğin normalizasyon) uygulanabilir. Bu, mesafe ölçümlerini içeren madencilik algoritmalarının doğruluğunu ve verimliliğini artırabilir. Bu teknikler birbirini dışlayan değil; birlikte çalışabilirler. Örneğin, veri temizliği,

bir tarih alanı için tüm girişleri ortak bir biçime dönüştürmek gibi yanlış verileri düzeltmek için dönüşümler içerebilir (Han ve Kamber 2006).

Bu çalışmada herhangi bir veri temizleme, eksik verileri tamamlama yöntemi kullanılmamıştır.

2.6. MODELLEME

Makine öğrenmesi alanında analiz yapanlar, farklı niteliklerden oluşan veriyi çeşitli yöntemler kullanarak modellemektedir; bu modellemedeki amaç veri içindeki gizli olan anlamı ve örüntüyü ortaya çıkarmaktır (Vellido ve diğ., 2012). Model bir görevin niteliklerle tanımlanan veriyi çıktılarına uygun ihtiyaç duyduğu resimlendirme olarak tanımlanış olup öğrenme problemlerinin öğrenme algoritmaları tarafından analiz edildiğini, görevler de modeller tarafından yönlendirilmektedir (Flach, 2012). Tüm modellerin yer aldığı $M = \{M_1, M_2, \dots, M_n\}$ kümesindeki asıl hedef bu modeller arasından en iyisini seçmektir (Rai, 2011):

- Aynı öğrenme modeli farklı biçimlerde denenebilir. Bu çalışmada Destek Vektör Makinesi için Doğrusal, Radyal, Sigmoid ve Polinomial çekirdeklerle çalışılmıştır. Optimal çekirdeği seçmek için grid yöntemi kullanılmıştır. Ayrıca parametre ayarlaması (parameter tuning) yapılarak farklı C ve gamma değerleri uygulanarak en iyi performansı gösteren parametreler belirlenmiştir. K-En yakın komşu algoritması için $k=3, 5$ ve 7 değerleri ile çalışılmıştır. Yapay sinir ağlarında farklı sayıda gizli katmanlarla çalışılmıştır. Rasgele Orman için farklı sayıda ağaçlarla çalışılmıştır.
- Birbirinden farklı öğrenme modelleri kullanılmıştır. Rasgele Orman, Yapay Sinir Ağları, Destek Vektör Makinesi gibi.

2.6.1. Naive Bayes Sınıflandırıcı

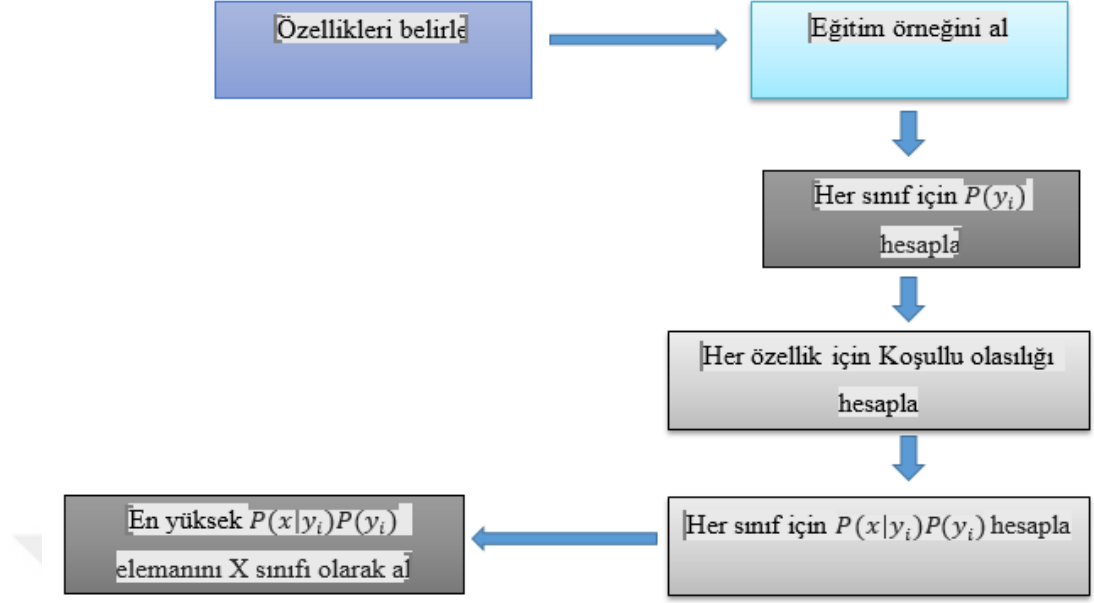
Naive Bayes sınıflandırıcısı sadeliği, hesaplama etkinliği, rekabetçi performansı, uygulanmasının ve yorumlanmasının kolaylığı, iyi performans göstermesi açısından gerçek dünyadaki uygulamalarda sınıflandırma problemlerini çözmek için yaygın olarak kullanılmaktadır (Friedman ve diğ., 1997; Hindi, 2014; Jiang ve diğ., 2009; Langley ve diğ., 1992).

En popüler Bayes algoritmalarından biri olan Naive Bayes Sınıflandırıcısı, Bayes teorisinin değişkenler arasındaki bağımsızlık, normallik ve bütün niteliklerin aynı derecede önemli olması varsayımlarına dayanan bir makine öğrenme algoritmasıdır. Naive Bayes Sınıflandırıcı

sayesinde bir örneğin hangi sınıfa ait olduğunun olasılıkları hesaplanmaktadır. Çeşitli sağlık alanlarında sınıflandırma araçlarını geliştirmek için verimli çalışabilme yeteneğine sahip olduğu bilinmektedir (Testi ve diğ., 2001; Kessler ve diğ. 2016; Miranda ve diğ., 2016). Sağlık alanında Naive Bayes Sınıflandırıcısı, osteoporotik hastalarda femur boynu kırılma riskini öngörmeye başarıyla kullanılmıştır (Testi ve diğ., 2001). Ordu çalışmasında intiharı öngörmesi, (Kessler ve diğ., 2016), kardiyovasküler hastalık risk seviyesinin öngörülmesi (Miranda ve diğ., 2016) ve protein-protein etkileşim bölgelerinin öngörülmesi (Geng ve diğ., 2015) sağlık alanlarındaki Naive Bayes sınıflandırıcı örneklerindendir. Hesaplama karmaşıklığının az olması ve büyük verilerle iyi çalışması bu algoritmanın tercih edilme nedenlerindendir. Metin sınıflandırma temel kullanım alanlarındandır. Bunun yanında spam filtreleme ve medikal araştırma gibi farklı alanlarda da kullanılmaktadır (Zhang ve Su, 2004). Hızlı ve kolay geliştirebilir olması Naive Bayes'in avantajlarındandır.

Bir Bayes sınıflandırıcısı, bir sınıfın rolünün o sınıfın üyeleri için özniteliklerin değerlerini tahmin etmek olduğu fikrine dayanır (Poole ve Mackworth, 2010). Her test vektörünün hangi sınıfa ait olduğunu Naive Bayes sınıflandırıcısı belirler. En yüksek olasılığa sahip sınıf tahmin edilen sınıftır.

Bir Bayes sınıflandırıcısının ardındaki fikir, eğer sınıf biliniyorsa, niteliklerin değeri tahmin edilebilir, eğer sınıf bilinmiyorsa, Bayes kuralının verilen özelliklerin verilen sınıfı tahmin etmek için kullanılabileceğidir. Bir Bayes sınıflandırıcısında, öğrenme aracı niteliklerin olasılıksal bir modelini oluşturur ve bu modeli yeni bir örneğin sınıflandırmasını tahmin etmek için kullanır. Eğitim örneğini aldıktan ve modelin özelliklerini belirledikten sonra, Bayes modellemesinin bir sonraki adımı, her bir sınıf için $P(x|y_i)P(y_i)$ değerini hesaplamaktır.



Şekil 2.4: Naive bayes sınıflandırıcısı (Friedman ve diğ., 1997).

Naive Bayes sınıflandırıcı kavramı şu şekilde açıklanabilir (Friedman ve diğ., 1997);

X veri örneği için, $X = X_1, X_2, X_3, \dots, X_n$ örnekler kümesi olsun. n özellik değişkenlerinin bir vektörü ve Y bir ikili sınıf değişkeni olduğunu varsayalım. X'in hangi sınıfa ait olduğunu tahmin etmek için Bayes Teoremi'nde sınıf olasılığı $P(Y|X)$ şu şekilde tanımlanabilir:

$$P(Y|X) = P(Y|X_1, X_2, X_3, \dots, X_n) = \frac{P(Y) \cdot P(X_1, X_2, X_3, \dots, X_n|Y)}{P(X_1, X_2, X_3, \dots, X_n)} \quad (2.2)$$

$X_1, X_2, X_3, \dots, X_n$ 'nin birbirinden bağımsız olduğu kabul edilerek basitleştirme yoluna gidilebilir. Bu durum koşullu bağımsızlık varsayımı olarak adlandırılmaktadır (Lewis, 1998).

Böylece aşağıdaki formül elde edilebilir:

$$P(X_1, X_2, X_3, \dots, X_n|Y) = \prod_{i=1}^n P(X_i|Y) \quad (2.3)$$

Denklem 2.2 şu şekilde değiştirilebilir:

$$P(Y|X) = \frac{P(Y)}{P(X_1, X_2, X_3, \dots, X_n)} \prod_{i=1}^n P(X_i|Y) \quad (2.4)$$

2.6.2.K-En Yakın Komşu Algoritması

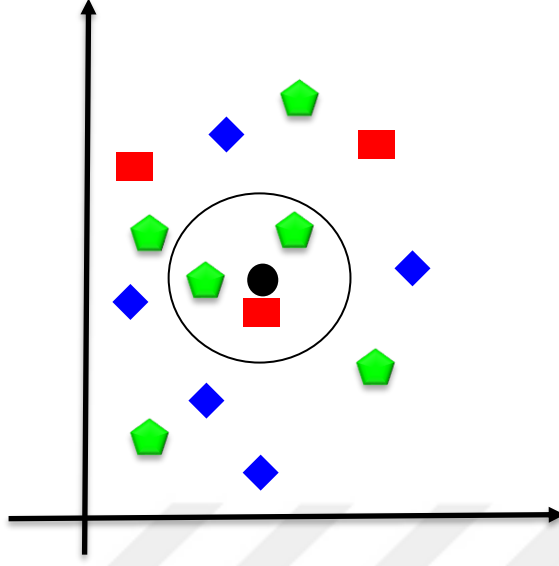
K-en yakın komşu sınıflandırması, dengesiz verilerin sınıflandırılmasını amaçlamaktadır ve veri madenciliğinin ilk 10 algoritması olarak seçilmiştir (Zhang, 2019).

K-en yakın komşu yöntemi, test örneğinin sınıf etiketini, test örneğinin en yakın komşularının etiketleri ile hesaplamaktadır (Soucy, P. ve Mineau, G.W., 2001).

Sınıflandırma yapılırken veri setindeki her verinin diğer verilere olan uzaklıkları hesaplanmaktadır. Bu uzaklığı belirlemek amacıyla bir kayıt için diğer kayıtlardan K adedi göz önünde bulundurulur. K adet kayıt mesafesi hesaplanan noktaya diğer kayıtlara göre en yakın olanlardır. K değerini belirlemek önemlidir çünkü K değerinin çok küçük olması modeli çok etkilemektedir. Aynı sınıfın noktaları olmalarına rağmen bazı noktaların ayrı sınıflara konmasına veya o noktalar için ayrı sınıf açılmasına sebep olmaktadır. Aynı şekilde çok büyük olması sadece tek bir sınıf varmış gibi olmasına ve birbirine benzemeyen noktaların bir arada sınıflandırılmasına neden olur. Bu nedenlerden dolayı K sayısının sınıflandırma üzerine etkisinin olduğu görülmektedir. Çoğunluk oylamasını referans alan 3, 5, 7 en çok kullanılan k değerleridir (Khan, 2002).

K-en yakın komşu algoritmanın adımları;

1. Adım: Yeni gelen örnek sınıfa eklenir.
2. Adım: k komşusuna bakılır
3. Adım: Uzaklık fonksiyonları kullanılarak uzaklık mesafesi hesaplanır.
4. Adım: En yakın neresiyse örnek o sınıfa atanır.



Şekil 2.5: k=3 için öklid mesafesi (Zhang, 2019).

Şekil 2.5'te ortadaki yuvarlak simgenin hangi sınıfa ait olduğu tahmin edilmek istendiğinde k değerine bakıldığında ortadaki yuvarlak veri beşgen nesnelere daha yakın mesafe olduğu için beşgen nesnelerle aynı sınıfta etiketlenecektir.

K-en yakın komşu algoritmasının iki ana araştırma yönü vardır. Birincisi, uygun bir K değeri belirlemektir. Bir diğeri K'ya en yakın komşuları tanımlamak için mesafe işlevidir.

K değerini ayarlamak için, genellikle kullanılan bir yöntem, olasılık teorisindeki çapraz doğrulamadır. Bir eğitim veri seti verildiğinde uygun bir K değeri belirlemek için kullanışlıdır. Ancak, eğitim örnekleri eğitim örnek alanında farklı yoğunluklarda dağıtılmaktadır. Bu, farklı örneklerin sınıf tahmini için farklı K değerlerine ihtiyaç duyduğu yeni zorlu bir soruna yol açmaktadır.

Pek çok mesafe fonksiyonu olmasına rağmen, çoğu K-en yakın komşu sınıflandırma algoritması, aşağıdaki gibi tanımlanan Öklid mesafesini kullanır (Dekhtyar, 2009).

$$d(i, j) = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2} \quad (2.5)$$

$d(i, j)$ iki örnek arasındaki mesafe, p , örnek sayısı, x_{ik} k örneklemindeki i örneğinin bir ağırlığıdır, x_{jk} , k örneklemindeki j örneğinin bir ağırlığıdır.

Uzaklık fonksiyonları Manhattan uzaklık fonksiyonu, Eşitlik 2.6'da, Minkowski uzaklık fonksiyonu, Eşitlik 2.7'de, Öklid uzaklık fonksiyonu da Eşitlik 2.8'de gösterilmektedir.

$$d(i, j): |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots \quad (2.6)$$

$$d(i, j): (|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \dots)^{1/q} \quad (2.7)$$

$$d(i, j): (|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots)^{1/2} \quad (2.8)$$

Seçilen en yakın komşular göz önüne alındığında, test örneğini sınıflandırmak ağırlıklar Bölgesel Ağırlıklandırılmış Ortalama Yöntemine göre $W = 1/d^2$ olarak hesaplanarak belirlenir. Bu kestirme ve tahminleme için kullanılır. (Archana, 2014).

2.6.3. Lojistik Regresyon

Lojistik regresyon modelinin uygulanması, biyolojik bilimler gibi birçok alanda belirgin bir şekilde öne çıkmıştır. Lojistik regresyon algoritması, amaç veri öğelerini kategorilere ayıran durumlar olduğunda kullanılır. Genellikle lojistik regresyonda hedef değişken ikilidir, yani bu sadece 1 veya 0 olarak sınıflandırılan verileri içerir; bu, bizim çalışmamızda akut apandisit için pozitif veya negatif olan bir hastayı ifade eder. Lojistik regresyon algoritmasının amacı, hedef değişkeni ile tahmin değişkenleri arasındaki ilişkiyi tanımlamak için tanısıl olarak makul olan en uygun seçimi bulmaktır. Lojistik regresyon algoritması, aşağıdaki denklemde (Wu ve diğ., 2018) verilen doğrusal regresyon modeline dayanmaktadır.

$$y = p_1(x) = \theta^T x \quad (2.9)$$

Denklem 2.9, ikili değerleri ($y^{(i)} \in \{0,1\}$) öngörmekte oldukça verimsiz olacaktır; bu nedenle, belirli bir hastanın (verilen niteliklere sahip olan) ait olma olasılığını tahmin etmek için (Han, Kamber ve Pei, 2012):

d-boyutlu özellik vektörü $X = (X_1, \dots, X_d)$ ve sınıf değişkenleri $Y \in \Theta = \{\theta_1, \theta_2\}$ olan ikili sınıflandırma problemi olduğunu düşünelim. $P_1(x), X = x$ olarak verilen $Y = \theta_1$ olasılığını gösterebilir. Aşağıdaki gibi logit dönüşüm yapılmaktadır. İkili lojistik regresyon modelinde;

$$\text{logit } p_1(x) = \ln \frac{p_1(x)}{1 - p_1(x)} = \beta^T x + \beta_0 \quad (2.10)$$

Olduğu varsayılır. Eşitlik 2.10'da gösterilen lojistik regresyon modeli arasındaki ilişkiyi kurmak için odds oranı kullanılmıştır (Hosmer, Lemeshow, ve Sturdivant, 2013).

$\beta \in \mathbb{R}^d$ ve $\beta_0 \in \mathbb{R}$ parametrelerdir.

$p_1(x)$ için aşağıdaki denklem elde edilir.

$$p_1(x) = \frac{1}{1 + \exp[-\beta^T x + \beta_0]} \quad (2.11)$$

$\{(x_i, y_i)\}_{i=1}^n$ olarak verilen öğrenme setinde, β ve β_0 parametreleri genellikle koşullu log olabilirliği en üst düzeye çıkarılarak tahmin edilir.

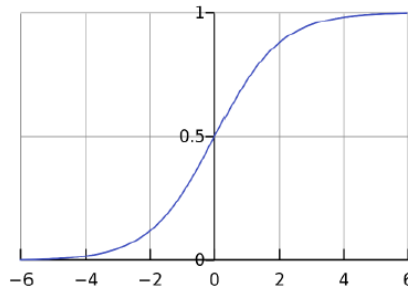
$$\ell(\beta, \beta_0) = \sum_{i=1}^n (y_{i1} \ln p_1(x_i) + (1 - y_{i1}) \ln[1 - p_1(x_i)]) \quad (2.12)$$

Burada $y_{i1} = 1$ $y_i = \theta_1$, $y_{i1} = 0$ ise

Sigmoid fonksiyonu, nonlineerite ve türevinin hesaplamalı basitliğinden dolayı ileri beslemeli sinir ağlarında kullanılan en bilinen fonksiyondur (Nauck, D. ve Kruse, R., 1999). Sigmoid fonksiyonu formun lojistik fonksiyonudur.

$$p = \frac{1}{1 + e^{-z}} \quad (2.13)$$

Bu sigmoid işlevi, Z'nin değerini 0 ila 1 arasında bir değerle eşler. Sigmoid işlevi, Şekil 2.13'te gösterilen forma sahiptir.



Şekil 2.6: Sigmoid fonksiyonu.

Seçilen eşik 0,5 olarak seçilmektedir. Hipotez, 0,5'ten büyük bir olasılık için 1 çıktısını ve 0,5'ten küçük bir olasılık için 0 sonucunu tahmin etmektedir.

2.6.4.Yapay Sinir Ağları

Yapay Sinir Ağları (YSA) biyolojik sinir hücresinin karmaşık yapısından ve çalışma prensibinden ilham alarak karmaşık biyolojik olayları öngörmede kullanılan standart istatistiksel tekniklere alternatif olarak önerilen bir yöntemdir (Sargent, 2001). YSA biyolojik beyin hücreleri olan nöronların çalışma prensibini taklit eden, öğrenebilen yapay nöron adı verilen bağlı birimler veya düğümlere dayanmaktadır (Abraham, 2004). YSA'da birbirlerine bağlı nöronlar, ağırlıkların belirlenmesi ve aktivasyon (ateşleme) fonksiyonu bulunmaktadır. Şekil 2.7'de basit bir nöronun bileşenleri gösterilmektedir.

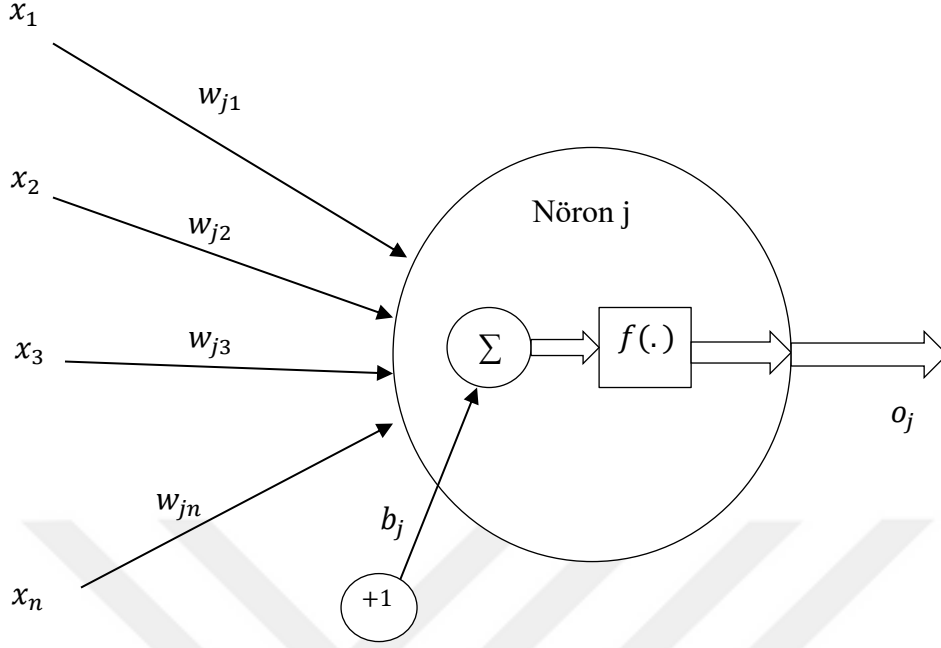
Ağın eğitimi için n giriş sayısı olmak üzere nörona verilen girdiler $X(x_1, x_2, x_3, \dots, x_n)$ her bir nöronda bulunan W ağırlıklar matrisi $(w_1, w_2, w_3, \dots, w_n)$ ile çarpılır. Tek tek tüm girdilere bu işlem uygulanır. Elde edilen çarpımlar ve bias nöronunun ağırlık değeri toplama fonksiyonuna tabi tutularak bir toplam oluşturulur. Toplam değer bir aktivasyon fonksiyonuna tabi tutularak çıktı katmanına iletilir (Fyfe, 2000).

İfadeler eşitlik olarak yazılırsa;

$$net_j = \sum_{i=1}^n w_i x_i + b_j \text{ ve } o_j = f(net_j) \quad (2.14)$$

$$o_j = f\left(\sum_{i=1}^n w_i x_i + b_j\right) \quad (2.15)$$

şeklinde de yazılabilir. Burada $f(\cdot)$ aktivasyon fonksiyonudur. Basit bir sinir modeli, n adet girişi, belirli ağırlıklarla çarparak toplamakta, bir eşik değeri de işleme katılıp sonucu aktivasyon fonksiyonundan geçirmektedir. Sinir modeli, sinirin eşik değerine ve kullanılan aktivasyon fonksiyona bağlı olmaktadır.



Şekil 2.7: Temel yapay sinir ağı hücresi (Haykin, 2008).

YSA'nın tipik ağ mimarisi giriş katmanı, gizli katman ve çıktı katmanı olmak üzere üç katmandan oluşmaktadır. Giriş katmanı, tahmin değişkenlerinin değerlerini (örneğin, rebound, WBC) alır ve çıkış katmanı, öngörülen sonucu (örneğin hasta veya sağlıklı) sağlar. Gizli ve çıkış katmanlarındaki her bir nöron, nörona giren tüm girdilerin ağırlıklı toplamını alır. Ardından, çıktı sonucunu hesaplamak için aktivasyon fonksiyonu uygulanır. Bu işlem aşağıdaki formülle gösterilebilir:

$$y(x) = f\left(\sum_{i=1}^N w_i x_i\right) \quad (2.16)$$

x_i giriş değeri, w_i ilgili ağırlık, N giriş sayısı ve f aktivasyon işlevidir.

Genel olarak, YSA aşağıdaki üç katmana ayrılmaktadır (Silva ve diğ., 2017). Giriş katmanı girdi veri gruplarının girildiği ve veri veya özelliklerin alındığı ilk katmandır. Bazı normalizasyon teknikleri, girişleri belirli bir aralıkta sınırlandırmak için bu girişlere uygulanır. Normalleştirilmiş girdiler sinir ağının çalışmasını kolaylaştırır ve daha iyi hassasiyet sağlar. Giriş katmanlarındaki parametrelerin analizden önce seçilmesi gerekir (Jock ve diğ., 1999). Bir giriş katmanındaki nöronların sayısı, giriş verisinin sayısına eşittir, her giriş nöronu bir sonraki katman olan gizli katmana iletilir. Gizli katman, ağın uygulamasına bağlı olarak bir katman koleksiyonudur. Bu katmanların sorumluluğu bir işlem veya sistemin modelini tanımadır.

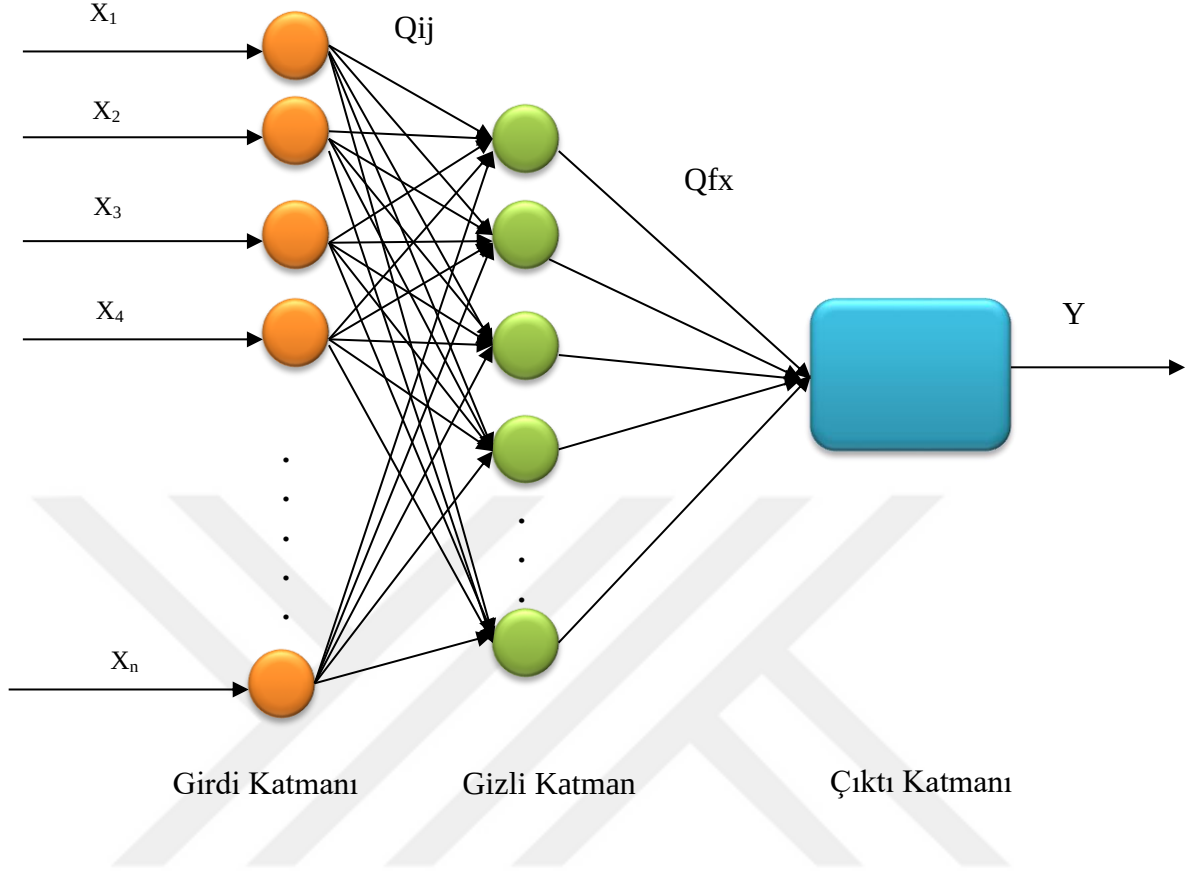
Çoğu sinir ağı aktivitesi bu katmanlarda gerçekleştirilir. Gizli katman, ağıın temel işlevidir. Bu katmanda, giriş katmanından alınan veriler düzgün bir şekilde işlenir ve ardından çıkış katmanına iletilir (Dag, 2012). Gizli katmanların belirlenmesinde çeşitli yöntemler kullanılmaktadır. YSA modelini oluşturma sürecinde, gizli bir katmandaki düğümlerin sayısını belirlemek için aşağıdaki denklem kullanılmaktadır (Kong ve diğ., 2016).

$$h = \sqrt{m + n} + a \quad (2.17)$$

h , gizli katmandaki hedef düğüm sayısıdır, m , çıktı katmanındaki nöronların sayısıdır, n , girdi katmanındaki değişkenlerin sayısıdır ve a , 1 ila 10 arasında değişen bir tam sayı sabitidir.

Choy ve diğerleri çeşitli kaynaklarda ara katmanlardaki işlem eleman sayısını hesaplama ile ilgili olarak giriş katmanındaki işlem elemanı sayısının iki katının bir fazlası, %75'i veya giriş ve çıkış katmanlarındaki işlem elemanı sayısının %50'si gibi teknikler verildiğini aktarmışlardır. Ancak yazarlar söz konusu çalışmada doğruluk seviyesinden ödün vermeden minimum ara katman işlem eleman sayısı ile çalışmanın doğru olacağını ifade etmişlerdir (Choy, K., L. ve diğ., 2003).

Öğrenme çıktı katmanında gerçekleşir. Doğrusal birimler, gizli katmanlardan oluşan çıkışa bağlanır. Ağdaki son katmandır ve gizli katmandan alınan verileri işler ve çıktıyı oluşturur. Nöronların sayısı ağ tarafından alınan çıkış sayısına eşittir. Elde edilen değerler, yapay sinir ağıındaki problemin çıktı değerleridir (Dağ, 2012).



Şekil 2.8: Yapay sinir ağı katmanları (Vassilis, Z., A. ve diğ., 2019).

2.6.4.1. Yapay Sinir Ağlarının Eğitilmesi

Yapay Sinir Ağlarında öğrenme sürecinde girişlerin alınarak aktivasyon fonksiyonundan geçirilerek bir çıkış üretilmektedir. Çıktı sonucu gerçek çıkışla karşılaştırılarak hata bulunmaktadır. Öğrenme algoritmalarıyla hata azaltılmaya ve gerçek çıkışa yaklaşılmaya çalışılır. Bu çalışma esnasında yenilenen YSA ağırlıklarıdır. Her bir iterasyonda ağırlıklar yenilenmek kaydıyla amaca ulaşılmaya çalışılır. Amaca ulaşıldığı zaman ağırlık değerleri saklanmaktadır. YSA öğrenme sürecini tamamladıktan sonra daha önce verilmeyen girişler verilir, YSA çıkışıyla gerçek çıkış karşılaştırılır. Yeni verilen örnekler doğru değere yaklaşıyorsa öğrenme tamamlanmaktadır. Test verileriyle ağ test edilir (Öztemel, 2003).

Geri Yayılım (Backpropagation) algoritması çok katmanlı ileri beslemeli YSA modeline göre çalışmakta olup çalışmalarda sıklıkla kullanılmaktadır. YSA'nın çıktı değeri ile gerçek çıktı karşılaştırılır. Eğer YSA'nın çıktı değeri istenildiği gibi değilse, aralarındaki fark değerini gösteren hata, ağırlıklarına geriye doğru yayılır ve ağırlıklar güncellenerek tekrar

hesaplama yapılarak çıktı katmanına ulaşılır. Algoritma ağı hatasının minimum olduğu noktaya kadar devam eder (Silahtaroglu, G., 2016).

Geri Yayılım, eğim düşümü olarak bilinen yöntemle güncelleme yapmaktadır. Eğitim örneği r üzerindeki hata E_r ile gösterilsin. Eğim düşümü, her w_{ij} ağırlığının aşağıdaki gibi güncellenmesini önermektedir (Kulkarni ve Harman, 2011).

$$\Delta w_{ij} = -\eta \frac{\partial E_r}{\partial w_{ij}} \quad (2.18)$$

Denklem'deki η katsayısı 0-1 aralığında bir değer alan parametre öğrenme katsayısıdır ve ağırlıklardaki değişim miktarını gösteren öğrenme oranını gösteren bir sabittir (Duda ve diğ., 2000).

2.6.4.2. Aktivasyon fonksiyonları

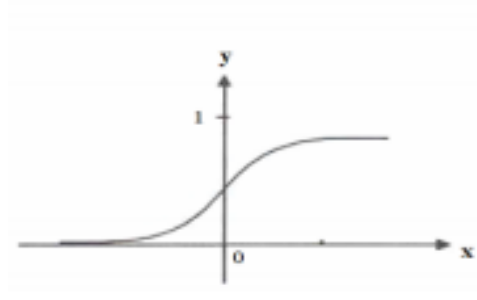
Aktivasyon fonksiyonu, her bir algılayıcının nihai çıktısını belirlemek için, algılayıcının girişinin ağırlıklı toplamını temsil eden net sinyale bağlı olarak kullanılır (Richards, Jia, 2006). Aktivasyon fonksiyonunun seçimi, algılayıcıyı ve net davranış ve yetenekleri belirler. En popüler aktivasyon fonksiyonları, lojistik sigmoid (Suzuki, 2013) ve hiperbolik tanjanttır (Haykin, 1999).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.19)$$

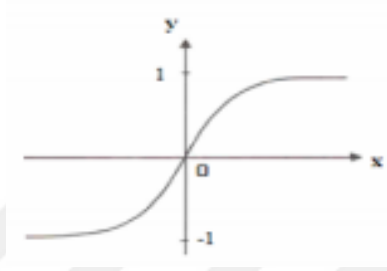
$$f(x) = \frac{2}{(1 + e^{-2x})} - 1 \quad (2.20)$$

Burada x nöron girdisidir.

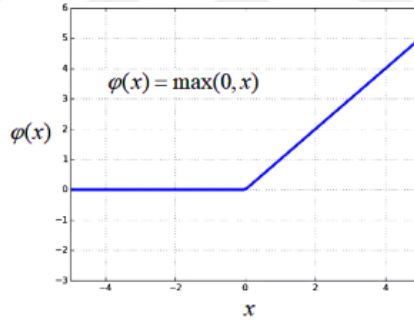
Aktivasyon fonksiyonları için giriş değerleri aralığı $[+\infty, -\infty]$ iken, çıkış değerleri lojistik ve hiperbolik fonksiyonlar kullanılarak eşik değere göre ikili hale getirilmektedir. Bu işlevler sırasıyla $[0, 1]$ ve $[-1, +1]$ aralığında sınırlandırılmıştır, bunlardan herhangi birinin kullanımı uygulamalara bağlı olmaktadır (Cartwright, 2015). Genellikle, lojistik fonksiyon aralığının kısıtlılığının üstesinden gelmek için hiperbolik fonksiyon kullanılır $[0, 1]$. Bir diğer aktivasyon fonksiyonu ReLU (düzeltilmiş doğrusal birim) aktivasyon fonksiyonudur. Popüler aktivasyon fonksiyonlarından biridir. 0-1 aralığında değer almaktadır. Bu işlevlerin eğrileri, Şekil 2.9, 2.10 ve 2.11 'de gösterilmektedir (Mohammed, A., 2018).



Şekil 2.9: Sigmoid.



Şekil 2.10: Tanjant fonksiyonu.



Şekil 2.11: ReLU fonksiyonu.

2.6.5. Destek Vektör Makinesi (DVM)

Destek Vektör Makinesi (DVM) ilk olarak Cortes ve diğerleri tarafından 1990'ların ortalarında geliştirilmiştir (Cortes ve Vapnik, 1995). DVM'nin amacı, farklı örnek kategorileri arasında ayırım yapmak için yüksek boyutlu uzayda optimal bir sınıflandırma sınırı (maksimum marjlı hiper düzlemler) oluşturmaktır. Boser çekirdek fonksiyonunu maksimum marjlı hiper düzlemlere uyguladıktan sonra DVM en esnek ve etkili veri madenciliği araçlarından biri haline gelmiştir (Boser ve diğ., 1992). Çekirdek işlevi sayesinde, DVM düşük boyutlu alandan yüksek boyutlu alana veri eşleyebilir. Yaygın olarak kullanılan çekirdek işlevleri doğrusal çekirdeği, polinom çekirdeğini ve radyal temel çekirdeğini içerir.

1990'ların sonunda, Vapnik, Cortes ve Boser, istatistiksel öğrenme teorisinden elde edilen bir sınıflandırıcı tanıtmıştır (Brereton ve Lloyd, 2010). Bu sınıflandırıcı, aynı yıl içinde makine öğrenen toplulukta yaygın olarak tanınan DVM'dir. DVM, regresyon ve sınıflandırmada sıklıkla kullanılan denetimli bir makine öğrenme tekniğidir. DVM, ikili sınıflandırma problemlerinde en yaygın şekilde kullanılır. DVM gittikçe büyümeye devam etmiş ve en popüler makine öğrenme tekniklerinden biri haline gelmiştir. DVM, biyoinformatik ve el yazısı tanıma dâhil birçok alanda kullanılmıştır (Brereton ve Lloyd, 2010). DVM, tıbbi teşhis, hava durumu tahmini, borsa analizi ve görüntü işleme gibi diğer uygulamalarda da kullanılmaktadır. Diğer tüm makine öğrenme tekniklerine benzer şekilde, DVM deneyimleri ve örnekleri nesnelere etiket dağıtmayı öğrenen bilgisayarlı bir algoritmadır. DVM'nin temel işlevi, etiketli veriler arasında maksimum mesafeye ulaşan bir çizgiye dayanan ikili etiketli verileri ayırmaktır (Furey ve diğ., 2000).

DVM iyi performans gösterir ve sınırlı örneklerle bile iyi bir hassasiyete sahiptir. Bu verimli avantajlar nedeniyle, DVM diğer makine öğrenme tekniklerinden farklıdır. DVM, etiketli verilerin ayrılmasına yardımcı olmak için çekirdek adı verilen işlevi kullanır. DVM'de çekirdek kullanımı ile ilgili avantajlardan biri, DVM'nin, vektör dışı girdilere, yani tıbbi alanda özellikle önemli olan ve biyolojik uygulamalar için kritik olan, büyüklük veya yönü olmayan girdilere çekirdek tanımları uygulamasıdır. Böyle bir avantaj DVM'nin DNA ve protein dizilerini etiketlemesini sağlar. Ek olarak, çekirdekler farklı veri tiplerinin bir kombinasyonuna dayanarak tanımlanabilir (Noble, 2006).

DVM doğrusal olarak ayrılamayan sınıfları ayırmak için önerilmiş iyi bilinen yöntemlerden biridir (Han ve Kamber, 2006).

Günümüzde DVM biyolojik uygulamalarda, tıpta ise kanser morfolojisinde, tedavi başarısının ve ilgili genin belirlenmesinde, çeşitli hastalıkların (şeker hastalığı, göğüs kanserleri, tiroit hastalığı ve kalp hastalıkları vs.) teşhisinde da büyük başarı göstermektedir (Guyon ve ark., 2002)

Sınıflandırma noktalarının bir çizgiyi veya hiper düzlemi kullanarak üç boyutlu olarak bölerek ayırma kavramı diğer makine öğrenim tekniklerinde de kullanılır. Bununla birlikte, DVM'de,

hiper düzlemin seçildiği teknik farklıdır. Hiper düzlemi maksimum hale getirmek için aşağıdaki fonksiyon kullanılmaktadır (Cortes ve Vapnik, 1995),

$$f(\vec{x}) = \text{sgn}\left(\sum_{i=1}^n y_i \alpha_i \cdot K(\vec{x}, \vec{x}_i) + b\right) \quad (2.21)$$

burada y , çıkış değişkenidir, b , önyargıdır ve α , her biri üzerinde aşağıdaki denklemin çözülmesinden elde edilir. giriş:

$$\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j \cdot y_i y_j \cdot K(\vec{x}_i, \vec{x}_j) \quad (2.22)$$

burada karşılık gelen α_i 0'dan büyükse, $\vec{x}_j$, adlandırılmış destek vektörleridir (Canu ve diğ., 2008).

2.6.5.1. Doğrusal Olarak Ayrılabilir Durumu

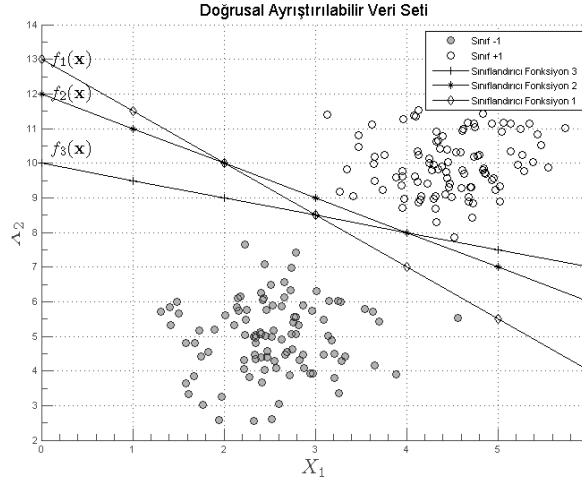
$(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$ D veri kümesi olduğunu varsayalım. Burada n $y \in \{+1, -1\}$ olduğu kabul edilir ve veri kümesinin eleman sayısıdır (Cristianini, 2000).

Her bir örneğin $\vec{x}_1$, m-boyutlu reel vektör ile ifade edilmekte, bu örneklerin sahip olduğu sınıf bilgisi y_i olarak gösterilmektedir. Her bir örneğin ait olabileceği sınıf kümesi -1 veya +1 şeklinde etiketlenmiştir. D veri setinde yer alan örnekler ve bu örneklerin ait oldukları sınıflar şu şekilde gösterilmektedir:

$$D = \{(x_i, y_i) | x_i \in R^n, y_i \in \{-1, 1\}\}_{i=1}^l \quad (2.23)$$

$$X = \begin{bmatrix} \vec{x}_1 \\ \vdots \\ \vec{x}_n \end{bmatrix} \in R^{n \times m} \text{ ve } \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \{-1, +1\}^n \quad (2.24)$$

DVM sınıflandırma algoritması, D veri setini ikiye ayırmak için l boyutlu bir hiperdüzlem bulmaya çalışmaktadır. Doğrusal olarak ayrıştırılabilir örnek veri setini ikiye bölen örnek hiperdüzlemler ve fonksiyonlar gösterilmektedir.

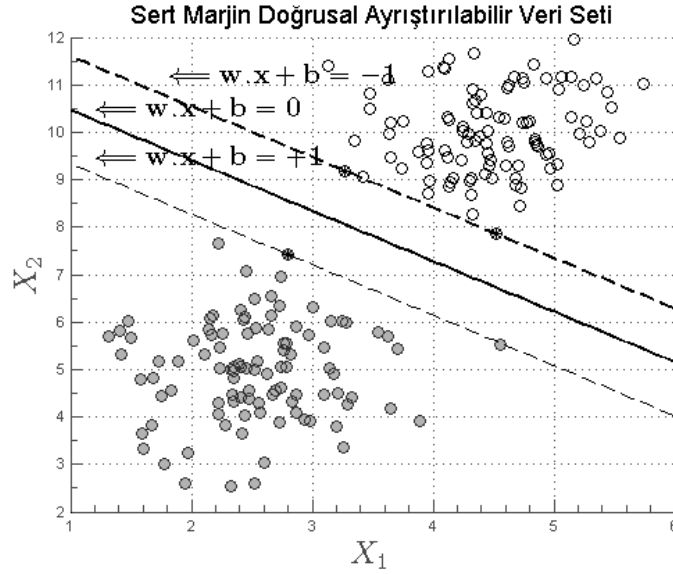


Şekil 2.12: İkili sınıflandırma için eğitim setini +1 ve -1 şeklinde ikiye ayıran hiperdüzlemlerin gösterimi (Çatak, F., Ö., 2015).

n: örneklem sayısı

m: Nitelik sayısı

k: Model yineleme endeksi



Şekil 2.13: Sert marjin doğrusal ayrıştırılabilir veri seti (Çatak, F., Ö., 2015).

Eğitim veri setini ikiye bölen sonsuz sayıda hiperdüzlem bulunabilir. İki sınıf arasında en geniş ayrımı gerçekleştirmek için DVM algoritması, bu hiperdüzlem içerisinde en geniş marjine sahip olan hiperdüzlemi seçmektedir. En geniş marja sahip olan hiperdüzlem seçmesinden dolayı, DVM algoritmasının genelleme özelliği artmaktadır. Veri kümelerini birbirinden ayırmak için

birçok hiper düzlemin kullanılabilir. En uygun yol ise iki hiper düzlem arasında en büyük marjine sahip olanları seçmektir.

Öğrenme probleminde yer alan veri seti -1 ve +1 şeklinde gösterilen iki farklı sınıfa ait olsun. Bu durumda eğitim setimiz $D = \{(x_i, y_i) | x_i \in R^n, y_i \in \{-1, 1\}\}_{i=1}^l$ şeklinde gösterilebilir. Burada $\vec{x}_i$ veri seti içerisinde yer alan i. vektör, y_i 'de bu vektörün ait olduğu sınıfın etiketini göstermektedir. Doğrusal sınıflandırıcı fonksiyonun tanımlanmasında $\vec{w}^T \vec{x} = \sum_i w_i x_i$ şeklinde iki vektörün iç çarpımı kullanılmaktadır. Doğrusal sınıflandırıcı, doğrusal diskriminant fonksiyonu temel alınarak aşağıdaki şekilde gösterilmektedir.

$$f(\vec{x}) = \vec{w}^T \vec{x} + b \quad (2.25)$$

Bu denklemde $\vec{w}$ ağırlık vektörünü, b ise hiperdüzlemin orijinden ne kadar uzak olduğunu belirten sabit değerini göstermektedir. Bu durumda hiperdüzlem şu şekilde tanımlanmaktadır.

$$(\vec{x}) = \vec{w}^T \vec{x} + b = 0 \quad (2.26)$$

Eğitim veri setinin bulunduğu uzayı ikiye bölen diskriminant fonksiyonuna dönüşmektedir. Örneklerin sahip olması gereken hedef nitelikleri ise eğitim uzayında yer alan bu hiperdüzleme göre bulundukları bölgeye göre atanmaktadır. Sınıflandırıcı denklemin işareti, örnek vektörün hedef sınıfını belirtmektedir.

$$y_m = \begin{cases} 1, & \vec{w}^T \cdot \vec{x} \geq 1 \\ -1, & \vec{w}^T \cdot \vec{x} \leq -1 \end{cases} \quad (2.27)$$

Denklem 2.27'de bulunan iki ifade tek bir denklem haline dönüştürülebilir ve şu şekilde yazılabilir:

$$y_i(\vec{w}^T \cdot \vec{x} - b) \geq 1 \quad (2.28)$$

2.6.5.2. Primal Form

Geometride iki paralel düzlem arasında ki uzaklık $\frac{2}{||w||}$ ile bulunmaktadır. DVM sınıflandırma algoritmasının amacı birbirinden oldukça uzak sınıflandırıcı hiperdüzleme paralel marjin bulmaktır. Bu durumda ortaya bir optimizasyon problemi çıkmaktadır.

Buradaki amaç $\frac{2}{||w||}$ ifadesinin maksimize etmektir. Bu maksimizasyon problemi, $||w||$ ifadesinin minimizasyonu şeklinde de ifade edilebilir. Bu ifade daha da geliştirilerek çözümü değişmeyecek şekilde $\frac{1}{2} ||w||^2$ ifadesine çevirilebilir. Bu şekilde optimizasyon problemimiz

kuadratik optimizasyon problemine dönüştürülebilmektedir. Yeni optimizasyon problemimi şu şekilde olacaktır.

$$w = \arg \min_{w \in R^m} \frac{1}{2} ||w||^2, \quad \text{kısıt: } y_i(\vec{w}^T \cdot \vec{x}) \geq 1, \quad i = 1, \dots, n \quad (2.29)$$

Bu denklem doğrusal kısıtlara sahip kuadratik fonksiyon minimizasyon problemine dönüşmektedir. Bu problemin çözümünde Lagrange fonksiyonu kullanılmaktadır.

Lagranj çarpanları yöntemi, matematiksel optimizasyonda eşitlik kısıtlarını kullanarak bir fonksiyonun lokal maksimum ve minimum değerlerini bulmak için kullanılan bir yöntemdir.

$$\begin{aligned} & f(x_1, \dots, x_n) \\ & g_i(x_1, \dots, x_n) = b_i, \quad i = 1, \dots, m, \quad m < n \end{aligned} \quad (2.30)$$

Denklem 2-30'da gösterilen fonksiyon için bir maksimizasyon problemi olsun. Lagranj çarpanları $(\alpha_i, i = 1, \dots, m)$ kullanılarak lagranj fonksiyonu şu şekilde olacaktır [45].

$$\mathcal{L}(x, \alpha) = f(x) + \sum_{i=1}^m \alpha_i (b_i - g_i(x)) \quad (2.31)$$

Lagranj fonksiyonu kullanılarak DVM sınıflandırma algoritmasında doğrusal kısıtlara sahip kuadratik fonksiyon minimizasyon problemi şu şekilde olacaktır.

$$\mathcal{L}(\vec{w}, b, \alpha) = \frac{1}{2} ||w||^2 - \sum_{i=1}^n \alpha_i (y_i(\vec{w}_i \cdot \vec{x}_i + b) - 1) \quad (2.32)$$

Lagranj fonksiyonuna, $\mathcal{L}$, dönüştürülmüş olan optimizasyon probleminde eğitim setinin olduğu uzayı ayıran hiper düzelm için w ve b değerlerinin bulunması gerekmektedir. Bunun için lagranj fonksiyonu $\mathcal{L}$ 'nin sırası ile w ve b 'ye göre türevlerinin alınması gerekmektedir.

$$\frac{\partial \mathcal{L}(\vec{w}, b, \alpha)}{\partial w_i} = 0, \quad i = 1, \dots, n, \quad \vec{w} = \sum_{i=1}^n \alpha_i y_i \vec{x}_i \quad (2.33)$$

$$\frac{\partial \mathcal{L}(\vec{w}, b, \alpha)}{\partial b} = 0, \quad i = 1, \dots, n, \quad 0 = \sum_{i=1}^n \alpha_i y_i \quad (2.34)$$

2.6.5.3. *Yumuşak Marj*

DVM sınıflandırıcısı, bir hiper düzlemi kullanarak veri noktalarını ayırmak için kullanılır. Ancak, zaman zaman, veri kümesindeki bir anormallik nedeniyle veri noktaları çizginin yanlış tarafında olabilir. Bu tür noktalar için, sınıflandırıcının yüzdelik doğruluğunu etkilemeden bazı noktaların hiper düzlemin yanlış tarafında olmasına izin vermek için yumuşak bir kenar boşluğu oluşturulmalıdır. Yumuşak kenarlık, DVM algoritmasının belirli parametrelerini değiştirerek ayarlanabilir. Bu parametreler, hiper düzlemin yanlış tarafında olabilecek noktaların sayısını izler. Bu parametreler ayrıca noktalar ve hiper düzlem arasındaki mesafeyi de kontrol eder.

2.6.5.4. *Doğrusal Olmayan Sınıflandırıcı İçin Çekirdek Hilesi*

Eğer bir veri seti doğrusal olarak ayrıştırılamıyor ise sınıfları ayırabilecek doğrusal bir hiperdüzlem bulunamaz. Bu durumda doğrusal olmayan sınıflandırıcı fonksiyon bulunması için doğrusal olarak ayrıştırılamayacak veri seti için doğrusal DVM sınıflandırma algoritmasının doğrusal olmayan DVM sınıflandırma algoritmasına dönüştürülmesi gerekmektedir.

Çekirdek işlevi, daha düşük bir boyutta ayrılmazsa, veri noktalarının daha yüksek bir boyut uzayında ayrılmasına yardımcı olur. Ancak, noktaları daha yüksek bir boyutta yansıtmak, boyutsallığın lanetine yol açabilir. Boyutluluk lanetine göre, yeterli bir doğruluk düzeyi elde etmek için, eğitim örneklerinin sayısı, özelliklerin sayısı arttıkça katlanarak artmalıdır. Noktaları daha yüksek bir boyuta yansıtmak, fazla uydurma sorununa da yol açabilir. Overfitting, sınıflandırıcının eğitim seti ile iyi çalıştığını ancak yeni örneklerde iyi bir hassasiyet seviyesine sahip olmadığı anlamına gelir. Bu nedenle, en iyi çekirdek işlevini geliştirmek için deneme yanılma yoluyla elde edilebilecek çekirdek için doğru sayıda boyut seçmeye ihtiyaç vardır. Daha sonra, çekirdeklerin doğruluk seviyelerini belirlemek için çapraz doğrulama uygulanabilir (Noble, 2006).

Çekirdek işlevi matematiksel denklemler kullanılarak temsil edilir. Aşağıdaki örneklerde gösterilen farklı çekirdek fonksiyonlarını temsil etmek için farklı denklemler kullanılmıştır. Doğrusal olmayan DVM sınıflandırıcı fonksiyonlarının bulunması iki adımdan oluşmaktadır.

- Eğitim veri setimizin boyutu girdi uzayından doğrusal olarak ayrıştırılabilecek aynı veya daha yüksek boyutlu nitelik uzayına dönüştürülür.

- Bu yeni nitelik uzayında maksimum marj ile doğrusal destek vektör makinesi ile hiperdüzlem bulunur. Bu hiperdüzlem yeni oluşturulan nitelik uzayında doğrusal iken, girdi uzayında ise doğrusal olmayan bir yapıdadır.

$\mathcal{D}$ veri setinde $\vec{w}$ örneği n -boyutlu girdi uzayı, $\phi(.)$ ise girdi uzayından daha yüksek boyutlu nitelik, $\mathcal{H}$ öklid uzayına dönüştürme yapan fonksiyon olsun. Bu fonksiyonu şu şekilde ifade edebiliriz.

$$\phi: \mathbb{R}^d \rightarrow H \quad (2.35)$$

2.6.5.4.1. En Sık Kullanılan Çekirdek Fonksiyonları

Literatürde sık olarak kullanılan çekirdek fonksiyonları doğrusal, polinom, sigmoid ve radyal tabanlı çekirdek fonksiyonlarıdır.

Doğrusal çekirdek fonksiyonu en temel fonksiyondur. $\vec{x}_i$ ve $\vec{x}_j$ vektörlerinin iç çarpımı ve seçimli olarak c sabitinden oluşmaktadır.

$$K(\vec{x}_i, \vec{x}_j) = \langle \vec{x}_i, \vec{x}_j \rangle + c \quad (2.36)$$

Polinom tabanlı çekirdek fonksiyonları özellikle eğitimde kullanılan veri setinin normalize olduğu durumlar için oldukça uygundur.

$$K(\vec{x}_i, \vec{x}_j) = (s\langle \vec{x}_i, \vec{x}_j \rangle + c)^d \quad (2.37)$$

Sigmoid çekirdek fonksiyonu literatürde hiperbolik tanjant çekirdek fonksiyonu ve çok katmanlı algılayıcı (perceptron) çekirdek fonksiyonu olarak da geçmektedir. Sigmoid çekirdek fonksiyonu, yapay sinir ağlarından destek vektör makinesi alanına gelmiştir. Yapay sinir ağlarında aktivasyon fonksiyonu olarak kullanılmaktadır.

$$(\vec{x}_i, \vec{x}_j) = \tanh(s\langle \vec{x}_i, \vec{x}_j \rangle + c) \quad (2.38)$$

Radyal Tabanlı Çekirdek Fonksiyonu, gauss çekirdek fonksiyonuna oldukça benzerdir. Normda bulunan kare ifadesi kaldırılmıştır.

$$K(\vec{x}_i, \vec{x}_j) = e^{-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{2\sigma_0^2}} \quad (2.39)$$

Tablo 2.2: DVM yöntemine ait bazı çekirdek fonksiyonlar.

Çekirdek Fonksiyonlar	
Doğrusal	$K(x_i, x_j) = x_i^T x_j$
Polinomiyal	$K(x_i, x_j) = (1 + x_i^T x_j)^d$
Radyal Tabanlı	$K(x_i, x_j) = \exp\left(\frac{\ x_i - x_j\ ^2}{2\gamma}\right)$
Sigmoid	$K(x_i, x_j) = \tanh(kx_i^T x_j - \delta)$

2.6.6. Rasgele Orman Algoritması (Random Forest)

Rasgele Orman (RO) çok sayıda karar ağacının sonuçlarını çoğunluk oyuyla toplayan topluluk sınıflandırıcısıdır (Kulkarni ve diğ., 2016). Topluluk öğrenme yöntemlerinde (ensemble learning) birden çok sınıflayıcının ortaya koyduğu sonuçlar bir araya getirilerek, topluluk adına tek bir karar verilmektedir. Ormandaki her karar ağacı, orijinal veri setinden bootstrap (yeniden yerleştirilerek örneklenen) tekniği ile farklı örneklemeler seçilerek oluşturulur ve yine rastgele torbalama (bagging) mekanizması ile seçilen bir öznitelik kümesi ile eğitilir (Breiman, 2001). Daha sonra, birbirinden farklı çok sayıda bireysel ağaç tarafından verilen kararlar oylamaya tabi tutar ve oylama sonucunda en çok oyu alan sınıfı topluluğun (komitenin) sınıf tahmini olarak sunar. RO yönteminde, ağaçlar CART (Sınıflandırma ve Regresyon Ağaçları) algoritmaları ve önyükleme torbalama birleşimi yöntemi ile oluşturulur (Breiman, 2001). Veri seti eğitim ve test verisi olmak üzere ayrılır. Eğitim veri setinden Bootstrap (yeniden yerleştirilerek örneklenen) tekniği ile ağaç oluşturacak veri (inbag) ve ağaç oluşturmayacak veri (out of bag) olmak üzere örneklemeler seçilir. Eğitim setinin 1/3'ü ağaç oluşturmayacak veri ve 2/3'ü ağaç oluşturacak veri olacak şekilde ayrılır. Ağaç oluşturulacak veri ile tüm değişkenler içinden her bir düğümde m tane değişken seçilir ve gini indeksi kullanılarak en iyi ayrılmalar sağlanır. Gini indeksini hesaplamak için matematiksel formül aşağıdaki gibidir:

$$Gini(p) = \sum_{k=1}^K P_k(1 - P_k) \quad (2.40)$$

P_k , düğümdeki k sınıfına ait örneklerin sıklığıdır ve k, toplam sınıf sayısıdır.

Ağaç oluşturulmayan veriler sayesinde kurulan modelde tahminler yapılır ve tahmin hataları hesaplanır. Her bir ağacın yaptığı out of bag kestirimleri birleştirilerek karar ağacının hatası kestirilir. Her bir ağaca out of bag hata oranına göre bir ağırlık verilir ve en düşük hata oranına

sahip ağaç en yüksek ağırlığı alırken en yüksek hata oranına sahip ağaç en düşük ağırlığı almaktadır (Han ve Kamber, 2006; Ho, 1995; Breiman, 2001). Sınıflama yapan her bir karar ağacı bireysel oy almakta ve işlem sonunda en yüksek oyu alan karar ağacının yaptığı sınıflama kullanılmaktadır. Her bir karar ağacı eğitildiği veri grubundan farklı bir veri grubuyla karşılaştığında aynı performansı gösteremeyeceği için, yöntem çok sayıda karar ağacını birleştirmekte ve bu sayede sınıflama performansını ve doğru sınıflama oranını artırmaktadır. RO yönteminde ağaç, tek bir düğümlle başlamakta, örneklerin hepsi aynı sınıfa ait ise düğüm, yaprak olarak sonlanmakta ve sınıf etiketi verilmektedir. Örnekler aynı sınıfa dahil değilse, örnekleri sınıflara en iyi bölecek olan özellik seçilmektedir. Şekil 2.14'te RO yöntemine ait basit bir gösterim yer almaktadır. RO yönteminin avantajlarından birisi de değişkenler arasında önem derecesini belirlemesidir. Önem derecesi belirlenirken aşağıdaki adımları gerçekleştirilmektedir:

- Karar ağacı oluşturulduktan sonra out of bag'e göre kestirilen sınıflar yukarıdan aşağıya doğru yerleştirilir ve doğru sınıflama sayısı kaydedilir.
- out of bag'deki m. değişkenin değerlerinin yeri değiştirilir ve artık değiştirilmiş out of bag' olur.
- Değiştirilmiş out of bag değerleri daha önceden oluşturulan karar ağacı üzerinde yukarıdan aşağıya doğru yerleştirilir ve doğru sınıflama sayısı hesaplanır.
- out of bag doğru sınıflama sayısından değiştirilmiş out of bag doğru sınıflama sayısı çıkarılır ve d_i hesaplanmış olur. Yukarıda sayılan tüm işlemler her bir karar ağacı için yapılır ve elde edilen d_i 'lerin ortalaması alınır ve $\bar{d}$ ile ifade edilir.
- Bulunan d_i değerlerinin standart hatası (SH) hesaplanır ve aşağıdaki Eşitlik yardımı ile önem derece skoru elde edilmiş olur.

$$\text{Önem Derece Skoru} = \frac{\bar{d}}{SH_{d_i}} \quad (2.41)$$

Anlatılan adımlar her bir değişken için tek tek uygulanır. Böylelikle her bir değişkene ait önem derece skorları hesaplanmış olunur. Değişken önem dereceleri için diğer bir yöntem ise gini değerlerinin yardımı ile hesaplanmaktadır. k. değişkenden dallara ayırma gerçekleşmeden önceki gini indeksi değerleri ile dallara ayırım yapıldıktan sonraki gini değerleri arasındaki fark alınır. Tüm ağaçlar için bu değer hesaplanır ve elde edilen değerler toplanır. Tüm değişkenler için bu değer hesaplanır ve buradan önem dereceleri hesaplanmış olunur.

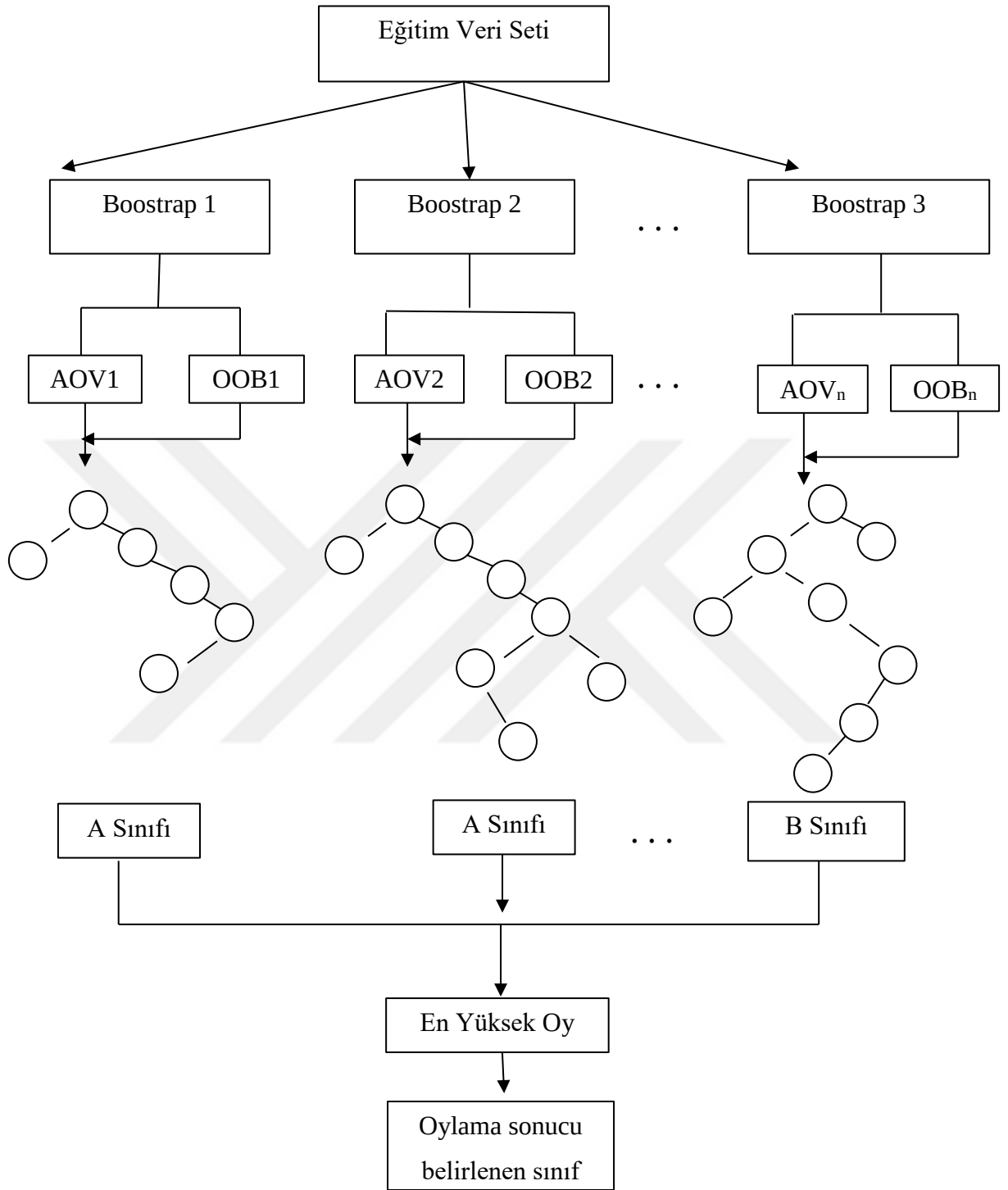
Bu tür bir stratejiye dayanarak, bir topluluk sınıflandırıcısı olarak, RO genellikle tek bir karar ağacı (Quinlan, 1986) ve Bayesian ağı (Cooper ve Herskovits, 1992; Tan, Steinbach ve Kumar, 2005) gibi tek bir sınıflandırıcıdan daha yüksek bir tahmin doğruluğu oranına sahiptir.

Gini indeksi, yükselen düğüm ve inen düğüm arasındaki fark, bir bilgi kazanım değeri sağlar (sınıfların daha iyi ayrılması). Bilgi kazancı:

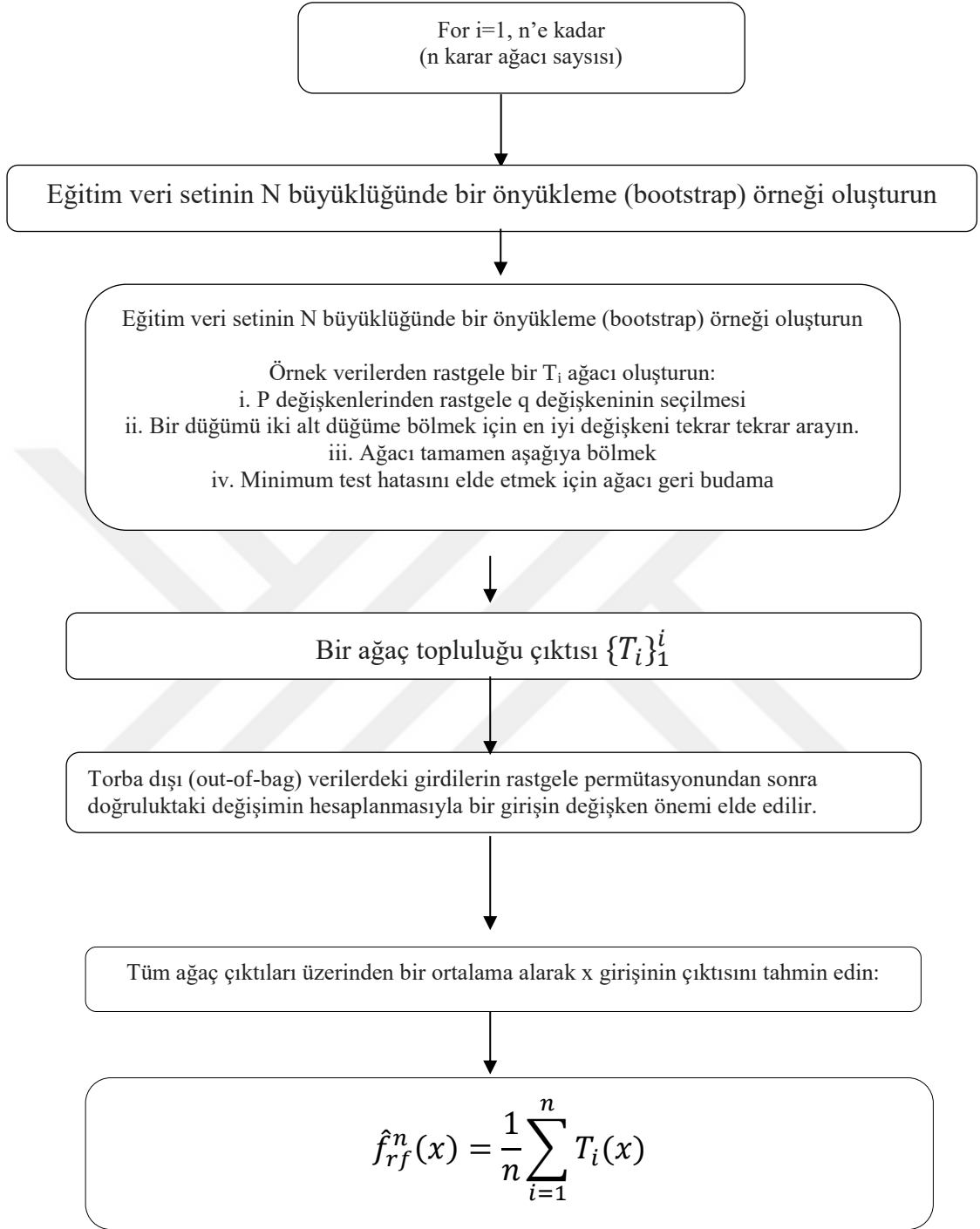
$$Gain_{Gini} = Gini(ascending) - \sum_{nd=1}^n \left(\frac{nd}{na} \right) * Gini(nd) \quad (2.42)$$

n azalan düğümlerin sayısı, nd azalan düğüm olaylarının toplam sayısı ve na, artan düğüm olaylarının toplam sayısıdır (Shang, Huang, Zhu, Lin, Qu, Wang, 2007; Park ve Kwon; 2011).

RO yönteminin genel yapısı Şekil 2.14'te gösterilmiştir.



Şekil 2.14: Rasgele orman şeması (Yu ve diğ., 2018).



Şekil 2.15: Rasgele orman algoritması adımları (Yu ve diğ., 2018).

RO modeli, bir hastanın hastanede yatma sırasında her hasta için ölüm olasılığı sağlayarak ölüp ölmeyeceğini tahmin edebilir ve olasılık, sonuç alan karar ağaçlarının toplam karar ağacı sayısına oranla pozitif sonuç vermesi ile belirlenir.

2.7. MODEL DEĞERLENDİRME VE SEÇİMİ

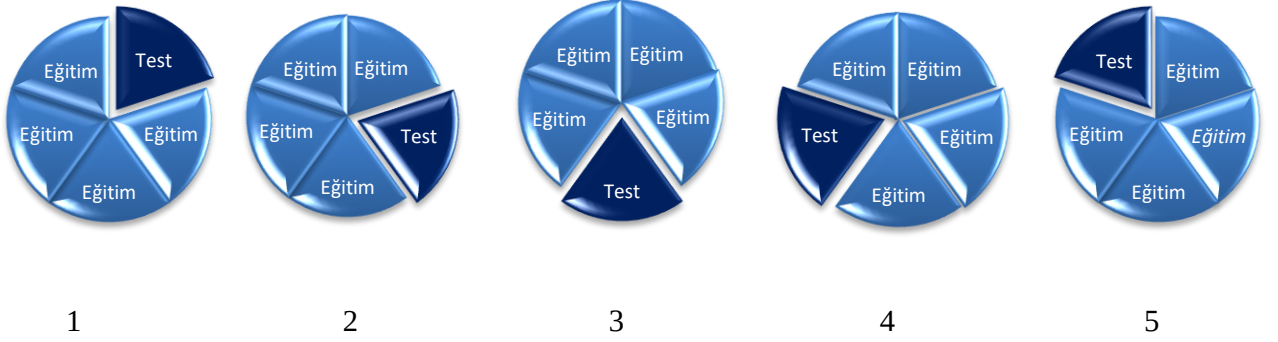
Modellerin tümünü içeren M kümesine ait her bir modelin performans değerlendirmesi için çeşitli performans değerlendirme yöntemleri vardır. Bu çalışmada performans değerlendirme yöntemleri olarak holdout ve çapraz geçерleme kullanılmıştır. Bu yöntemlerle ilgili genel bilgilere kısaca yer verilecektir.

2.7.1. Model Performans Değerlendirme Yöntemleri

Kullanılacak bir modelin performansını ölçerken sadece kullanılan öğrenme algoritmasına bakmamak gerekmektedir, bunun yanında eğitim ve test kümelerinin büyüklüğü, hatalı sınıflandırma ya da sınıf dağılımı önemli rol oynamaktadır (Zhang, 2014). Bunun için geliştirilen çeşitli yöntemlerden bir holdut ve çapraz geçерleme metodudur (Kartal, E., 2015).

Holdout yönteminde veri seti eğitim ve test veri seti olmak üzere ikiye ayrılmaktadır. Burada eğitim veri seti, seçilen sınıflandırıcı modelinin eğitimi için kullanılmaktadır. Modeli oluşturma aşaması için en uygun parametreleri değerleri ve performans ölçüsü bu aşamada belirlenmektedir. Modelin genel genel performansını ölçmek için de test veri seti kullanılmaktadır.

Çapraz geçерleme metodunun bir yöntemi olan k -kat çapraz geçерleme (k -fold) veri seti k eşit parçaya ayrılmaktadır. Bu k parçalarında her defasında bir tanesi test, geriye kalan $k-1$ tanesi de eğitim için kullanılmaktadır. Bu yöntemin sonucunda ortaya çıkan k tane hata oranı elde edilir ve hataların ortalaması alınarak tüm tahmin hatası hesaplanmaktadır. k kat sayısını belirlerken dikkatli olunmalıdır çünkü belirlenen k kat sayısının büyük olması durumunda, tahminci daha doğru olması nedeniyle gerçek hata tahmincisinin sapması (bias) küçük, varyans ve hesaplama zamanı ise büyük olacaktır (Gutierrez, 2013; Saharidis ve diğ., 2011). Diğer durumda, k kat sayısı küçük seçilirse, tahmincinin sapması küçük ve sapması gerçek hata tahmincisinin büyük olacaktır (Remesan ve Mathew, 2014; Gutierrez, 2013). Sapmadan kaynaklanan hata gerçek değer ile tahmin edilen değer arasındaki farktır, varyanstan kaynaklanan hata ise verilen bir nokta için model tahminindeki değişikliklerdir (Scott, 2012). Bu çalışmada kullanılan 5-kat çapraz geçерleme 2.16'da gösterilmektedir.



Şekil 2.16: 5-Kat çapraz geçiş (Rogers ve Girolami 2011).

2.7.2. Model Performans Değerlendirme Ölçüleri

Değerlendirme yöntemi, sınıflandırma performansını değerlendirmede ve sınıflayıcı modellemesini yönlendirmede kilit bir faktördür. Sınıflandırma sürecinin üç ana aşaması vardır; eğitim aşaması, geçerlilik aşaması ve test aşaması. Model girdi kalıpları kullanılarak eğitilmiştir ve bu aşama eğitim aşaması olarak adlandırılır. Bu girdi modellerine modelin eğitimi için kullanılan eğitim verileri denir. Bu aşamada, bir sınıflandırma modelinin parametreleri ayarlanır. Eğitim hatası, eğitilmiş modelin eğitim verilerine ne kadar uygun olduğunu ölçer. Ancak, eğitim hatası her zaman test hatasından ve doğrulama hatasından daha küçüktür, çünkü eğitilmiş model, eğitim aşamasında kullanılan aynı veriye uymaktadır. Bir öğrenme algoritmasının amacı, görünmeyen veriler için sınıf etiketlerini tahmin etmek için eğitim verilerinden öğrenmek; bu test aşamasında. Bununla birlikte, test hatası veya örnek dışı hata tahmin edilemez çünkü test örneklerinin sınıf etiketleri veya çıktıları bilinmemektedir. Test aşamasının, eğitilmiş modelin performansını değerlendirmek için kullanılmasının nedeni budur. Test aşamasında, test verileri, modelin hiperparametrelerini ayarlarken eğitilmiş modelin tarafsız bir değerlendirmesini sağlar.

Sınıf sayısına göre, iki tür sınıflandırma problemi vardır, yani sadece iki sınıfın bulunduğu ikili sınıflandırma ve sınıf sayısının ikiden yüksek olduğu çoklu sınıflandırma. Pozitif sınıf için P ve Negatif sınıf için N olmak üzere iki sınıf olduğu varsayılırsa, bilinmeyen bir örnek P veya N'ye sınıflandırılır. Eğitim aşamasında eğitilmiş olan sınıflandırma modeli, bilinmeyen örneklerin gerçek sınıflarını tahmin etmek için kullanılır. Bu sınıflandırma modeli sürekli veya ayrık çıktılar üretir. Bir sınıflandırma modelinden üretilen ayrık çıktı, bilinmeyen / test

örneğin öngörülen ayrık sınıf etiketini temsil ederken, sürekli çıktı, örneğin sınıf üyeliği olasılığının tahminini temsil eder.

Sınıflandırma sonucu elde edilen sonuçlar karmaşıklık matrisi (confusion matrix) veya karmaşıklık tablosu ile ifade edilmektedir. Bu matriste sınıflandırılan verinin gerçek ve tahmin sınıfları gösterilmektedir. Matrisinin satırları sınıflandırma sonucu elde edilen sonuçları sütunları gerçek değerleri göstermektedir. Tablo 2.3'te iki sınıfa ait karmaşıklık matrisi sunulmuştur.

Karmaşıklık matrisi, birçok yaygın sınıflandırma ölçütünü hesaplamak için kullanılır. Makine öğrenmesinde, sınıflandırma yöntemlerinde kurulan modelin performansını ölçen çeşitli ölçüler vardır. İkili sınıflama durumu için aşağıda verilen 2x2'lik sınıflama tablosundan yararlanılarak hesaplanmaktadır. Bunlardan bazıları aşağıdaki gibidir:

- Genel doğruluk oranı (overall accuracy)
- Duyarlılık (Sensitivity)
- Seçicilik (Specificity)
- Eğri altında kalan alan (Area Under Curve)
- F-Ölçüsü (F-Measure)

Tablo 2.3: İki sınıf için oluşturulmuş örnek bir karmaşıklık matrisi (Han ve Kamber, 2006).

Karmaşıklık Matrisi		Verinin Tahmin Edilen Sınıfı	
		Negatif (AA+) Müdahale Edilmeli	Pozitif (AA-) Gözlemlenmeli
Verinin Gerçek Sınıfı	Negatif (AA+) Müdahale Edilmeli	Doğru Negatif (DN)	Yanlış Pozitif (YP)
	Pozitif (AA-) Gözlemlenmeli	Yanlış Negatif (YN)	Doğru Pozitif (DP)

Karmaşıklık matrisi tablosunda yer alan DN gerçekte akut apandisit olan ve sınıflandırma sonucunda da akut apandisit olarak etiketlenen örnek sayısını, DP gerçekte akut apandisit olmayan ve sınıflandırma sonucunda da akut apandisit olmayan olarak etiketlenen örnek sayısını gösterir. YP gerçekte akut apandisit olup ve sınıflandırma sonucunda akut apandisit olmayan olarak etiketlenmiş örnek sayısını göstermektedir. YN gerçekte akut apandisit olmayan olup sınıflandırma sonucunda akut apandisit olarak etiketlenen örnek sayısını gösterir.

Bu tez kapsamında model başarıları değerlendirilirken doğruluk, seçicilik, duyarlılık, kesinlik, F-ölçütü, kappa değeri ve ROC eğrisi altında kalan alan (AUC) ölçütleri kullanılmıştır.

2.7.2.1. Doğruluk

Doğruluk, oluşturulan modelin genel doğru sınıflandırma başarısını gösteren ve sınıflandırma performansı için en yaygın kullanılan ölçütlerden biridir. Doğru sınıflandırılmış örnekler ile toplam örnek sayısına oranı Denklem 2.43'teki gibi hesaplanmaktadır (Witten ve diğ., 2016):

$$\text{Doğruluk} = \frac{DP + DN}{DP + DN + YP + YN} \quad (2.43)$$

Doğruluk oranının sınıf dengesizliğinde kullanılmaması gerektiği çeşitli kaynaklarda belirtilmektedir (Han, Kamber, 2006; He, Garcia, 2009; Menardi ve Torelli, 2014; Karaağaoğlu, ve diğ., 2016). Doğruluk oranı, sıfır ile bir arasında değişmektedir. Bu değer bir e yaklaşıması sınıf başarısının yüksek olduğunu göstermektedir.

2.7.2.2. Seçicilik

Seçicilik gerçek durumda negatif olan bir örneğin tahmin sonucunda da negatif çıkması olasılığıdır (Han, Kamber, 2006; Karaağaoğlu ve diğ., 2016, Pepe, 2003).

Testin, hasta olanlar içinden hasta olanları ayırma durumu olarak adlandırılmaktadır. Yani gerçekte akut apandisit olan hastaların kaç tanesinin model tarafından doğru tahmin edildiğini göstermektedir (Larose, D.T., 2014). Seçicilik Denklem 2.44'teki gibi hesaplanır:

$$\text{Seçicilik} = \frac{DN}{DN + YP} \quad (2.44)$$

Seçicilik değeri, 0 ile 1 arasında değişmektedir. Yüksek seçicilik değerleri yüksek sınıflandırma performansını gösterir.

2.7.2.3. *Duyarlılık*

Duyarlılık gerçek durumda pozitif bir örneğin tahmin sonucunda da pozitif çıkma olasılığıdır (Han, Kamber, 2006; Karaağaoğlu ve diğ., 2003). Sınıflama performanslarını gösteren sonuçlarda "geri çağırma (recall)" diye adlandırılmaktadır.

Testin, akut apandisit olmayanlar içinden akut apandisit olmayanları ayırma yeteneğidir. Yani gerçekte akut apandisit olmayan hastaların ne kadarının sistem tarafından doğru tahmin edildiğini verir (Larose, D.T., 2014). Duyarlılık Denklem 2.45'deki gibi hesaplanır:

$$Duyarlılık = \frac{DP}{DP + YN} \quad (2.45)$$

Duyarlılık değeri, sıfır ile bir arasında değişmektedir. Yüksek duyarlılık değerleri yüksek sınıflandırma performansını gösterir.

Sınıflandırma performansını dengesiz verilerle değerlendirmek için hem duyarlılık hem de özgüllük kullanılabilir (Sokolova ve diğ., 2006).

2.7.2.4. *Kesinlik*

Kesinlik (Precision), pozitif olan bir örneğin model tarafından hangi oranda doğru sınıfa atandığını veren değerdir. Akut apandisit olmayan olarak doğru sınıflandırılmış örnek sayısının, model tarafından akut apandisit olmayan olarak sınıflandırılan toplam örnek sayısına oranıdır (Larose, D.T., 2014). Kesinlik Denklem 2.46'daki gibi hesaplanır:

$$Kesinlik = \frac{DP}{DP + YP} \quad (2.46)$$

Kesinlik değeri, sıfır ile bir arasında değişmektedir. Yüksek kesinlik değerleri yüksek sınıflandırma performansını gösterir.

2.7.2.5. *F-Ölçütü*

F-skoru olarak da adlandırılan F-Ölçütü sınıflandırma başarımlarını ölçütleri hesaplanırken tek başına kesinlik veya duyarlılığın hesaplanması eksik kalabileceği düşünülerek hesaplanmıştır

(Powers, 2007). F-Ölçütü değeri, kesinlik ve duyarlılık değerlerinin harmonik ortalamasını temsil etmektedir. F-ölçütü değeri sıfır ile bir arasındadır ve yüksek F-ölçütü değerleri yüksek sınıflandırma performansını gösterir (Sokolova ve Japkowicz, 2006).

$$F - \text{Ölçütü} = 2 * \frac{\text{Duyarlılık} * \text{Kesinlik}}{\text{Duyarlılık} + \text{Kesinlik}} \quad (2.47)$$

2.7.2.6. Kappa Değeri

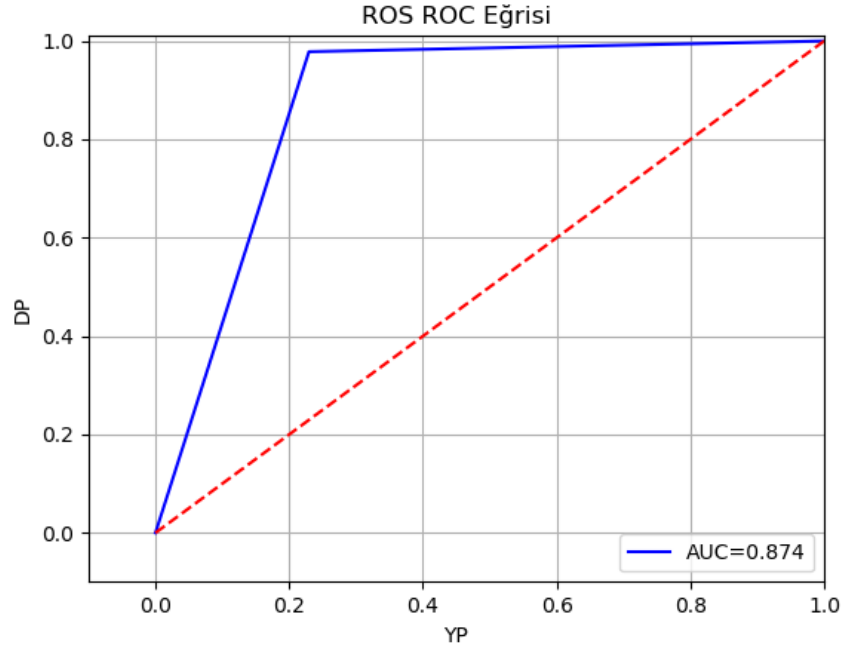
Kappa değeri sağlık alanında hastalık veya başka bir faktörün varlığını veya yokluğunu belirlemek için isimsel ölçekte uyumluluğunu belirlemede sıklıkla kullanılmaktadır (Fleiss ve diğ., 2013). Kappa değeri yapılan sınıflandırmanın başarılı olup olmadığı hakkında fikir vermektedir. Kappa değerinin 0'dan küçük olması sınıflandırmanın zayıf, 0.0-0.20 aralığı önemsiz olarak, 0.21-0.40 aralığı orta, 0.41-0.60 aralığı makul, 0.61-0.80 aralığını önemli ve 0.81-1.00 aralığı ise neredeyse mükemmel olarak yorumlanmaktadır (Landis ve Koch, 1977).

2.7.2.7. ROC Eğrisinin Altında Kalan Alan (AUC)

ROC (İşlem Karakteristik Eğrisi- Receiver Operating Characteristics) eğrisi, Doğru Pozitif Oranının (duyarlılık) y eksenini ve Yanlış Pozitif Oranının (seçicilik) x eksenini oluşturduğu iki boyutlu bir grafikdir. ROC eğrisi, tanı sistemleri, psikoloji, tıbbi karar verme sistemleri, radyoloji ve makine öğrenme sistemleri gibi birçok sistemi değerlendirmek için kullanılmıştır. ROC eğrisi, teşhis durumlarında ayrımı gerçekleştirdiği için yaygın olarak kullanılmaktadır. Eğrinin altında kalan alan (Area Under Curve) AUC ile gösterilmektedir. Eğri Altında Kalan Alan (EAA) farklı kesim noktalarında farklı duyarlılık ve seçicilik değerleri ile elde edilen ROC eğrisi altında kalan alandır (Han ve Kamber, 2006; Karaağaoğlu ve diğ., 2016). EAA değeri, sıfır ile bir arasında değişmektedir. Bu değer yüksek olması performansın yüksek olduğunu göstermektedir. Şekil 2.17'da AUC değerleri gösterilmektedir.

Sınıflandırmanın başarısı için (1,0) noktası en iyi sonuç veren noktadır. Bu durumda modelin tamamen doğru tahminde bulunduğu anlamına gelir. Bunun tersine, (0,1) noktası ise tahminin en kötü noktasıdır ve sınıflandırmanın tamamen yanlış tahminde bulunduğunu göstermektedir.

$$\widehat{EAA} = \int_0^1 \widehat{ROC}(t) dt \quad (2.48)$$



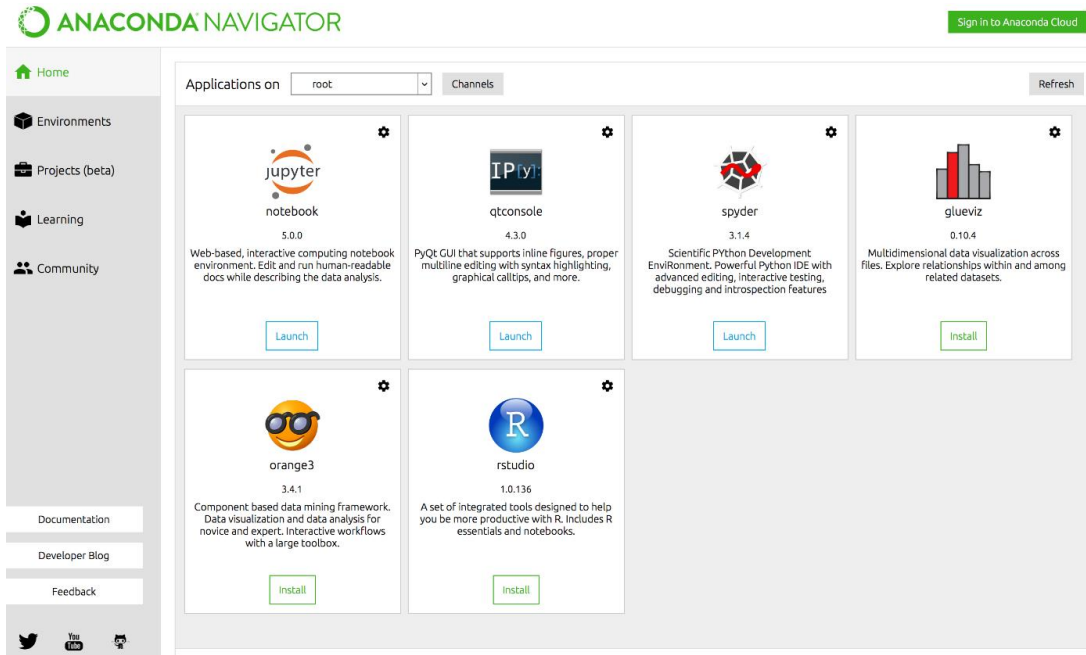
Şekil 2.17: ROC eğrisi (Yu ve diğ., 2018).

3. MALZEME VE YÖNTEM

Çalışmanın malzeme ve yöntem bölümünde sınıflandırma analizi için kullanılan yazılım ve sınıflandırma modeli için kullanılan veriler hakkında bilgi verilmiştir.

3.1. KULLANILAN YAZILIM

Bu bölümde kullanılan sınıflandırma modelleri analizi ve çıktıları Anaconda3 sürüm 5.0.1 kullanılarak elde edilmiştir. Anaconda dağıtım paket yöneticisi, çevre yöneticisi ve Şekil 3.1'de gösterilen sürümü kurmak kolaydır ve ücretsizdir. 1.000'den fazla kombinasyonu içeren Python dağıtımı ve paketleri ücretsiz topluluk desteğine sahip açık kaynaktır. Anaconda platform Windows, macOS veya Linux'ta kolaylıkla kullanılabilir. Anaconda Enterprise, veri bilimi varlıklarını yöneten, işbirliği yapan ve veri bilimi projelerini kullanan ekipleri güçlendiren, işleme hazır, güvenli ve ölçeklenebilir bir veri bilimi platformunda birinci sınıf kurumsal destek sunan Anaconda'dır. Anaconda tüm python platformları için bir şemsiye olarak kabul edilir ve bu çalışmada Spyder 3.3.1 Python kodlama dili kullanılmıştır, çünkü yaklaşık olarak tüm veri madenciliği teknikleri ile güçlüdür ve ayrıca doğruluk, duyarlılık, kesinlik gibi performans ölçümleri için yüksek doğruluk sağlamaktadır.



Şekil 3.1: Anaconda arayüzü.

Bu çalışmada sınıflandırma yöntemleri için kullanılan Python Kütüphaneleri aşağıda listelenmiştir:

```
import numpy as np
import pandas as pd
from collections import Counter
from sklearn.model_selection import train_test_split
from imblearn.pipeline import make_pipeline as make_pipeline_imb
from sklearn.model_selection import KFold
from matplotlib import pyplot as plt
from sklearn.decomposition import PCA
import pydotplus
from imblearn.over_sampling import SMOTE
from imblearn.under_sampling import RandomUnderSampler
from imblearn.under_sampling import RandomOverSampler
from imblearn.metrics import classification_report_imbalanced
from sklearn.metrics import accuracy_score, precision_score, recall_score
from sklearn.metrics import roc_auc_score, roc_curve, cohen_kappa_score
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from keras.models import Sequential, Model
from keras.layers import Dense, Input
from keras.utils import plot_model
import tkinter as tk
from PIL import ImageTk, Image
```

3.2. TAHMİN İÇİN KULLANILAN NİTELİKLER

Hastaneye karın ağrısı şikâyeti ile başvuran ve akut apandisit ön tanısı ile ameliyat edilen 1454 hastanın tüm klinik verileri Medipol Mega Üniversite Hastanesi, Başkent Üniversitesi Tıp Fakültesi Genel Cerrahi Anabilim Dalı ve İstanbul Esenyurt Necmi Kadioğlu Devlet Hastanesi Genel Cerrahi Kliniğinden temin edilmiştir. Medipol Mega Üniversite Hastanesi veritabanından 796 hastanın anamnez (öykü), fiziki muayene bulguları, biyokimya laboratuvar test sonuçları ve epikriz raporları retrospektif olarak alınmıştır. Başkent Üniversitesi Tıp Fakültesi'nde Laparoskopik ve Tek İnsizyon Laparoskopik Apendektominin Prospektif Karşılaştırma Sonuçları adlı uzmanlık tezini hazırlamış olan Doç.Dr. İlker Murat Arer'in hastane sistemi veri tabanından retrospektif taranarak kayıt altına aldığı akut apandisit ön tanısı ile ameliyat edilen 74 hastanın verileri kullanılmıştır. İstanbul Esenyurt Necmi

Kadıoğlu Devlet Hastanesi için oluşturulan Klinik Karar Destek sistemi için eski Başhekim Op. Dr. Ahmet Atilla Yılmaz'ın hastane sistemi veri tabanından retrospektif taranarak kayıt altına aldığı ve apandisit ön tanısıyla ameliyat edilen 584 hastanın verileri kullanılmıştır.

Akut apandisit olma riskinin tahmini için kullanılan Alvarado skoruna ait risk faktörleri kullanılmıştır. Alvarado skora göre iştahsızlık, bulantı veya kusma, ağrının periumblikal bölgeden sağ alt kadrana yer değiştirmesi, sağ alt kadranda hassasiyet, ateş yüksekliği ve rebound ağrı verileri edinilmiştir. Literatür çalışmalarına göre vücut sıcaklığının 38°C 'nin üzerinde olması yüksek ateş olarak kabul edilmiştir. Laboratuvar bulguları taranarak lökositoz (WBC) ve nötrofil yüzdesi (NE%) değerleri kaydedilmiştir. Alvarado skoruna göre WBC için alt sınır 10.000/mm³ ve nötrofili için nötrofil yüzdesinin alt sınırı %75 olarak kabul edilmiştir. Makine öğrenmesi algoritmaları uygulanırken WBC ve NE% değerleri sayısal olarak kullanılmıştır. Tüm hastaların patoloji raporları incelenerek histopatolojik olarak akut apandisit tanısı alanlar 0 ve almayanlar 1 olarak kaydedilmiştir. Bunlara ek olarak tüm hastaların yaş ve cinsiyet bulguları da verilere eklenerek bu iki niteliğin sınıflandırma üzerinde etkisi olup olmadığı tespit edilmiştir.

Akut apandisit tanısı için çeşitli yöntemler mevcuttur. Bu çalışmada Alvarado skoruna karşı yapay zekânın alt dalı olan makine öğrenmesi algoritmaları kullanılmıştır. Kullanılan hasta verileri dengesiz veri setleri olduğundan veri ön işleminde verileri dengeli hale getiren Random Oversampling, SMOTE ve Random Undersampling yöntemleri kullanılmıştır. Sınıflandırma algoritmalarından Yapay Sinir Ağları, Lojistik Regresyon, Naive Bayes, Destek Vektör Makinesi, K-en yakın komşu algoritması, Rasgele Orman algoritmaları kullanılmıştır.

Algoritmaları birbirlerine göre kıyaslayabilmek model performans değerlendirme yöntemlerinden tabakalı örnekleme (holdout) ve tabakalı 5-kat çapraz geçişleme yöntemleri uygulanmıştır.

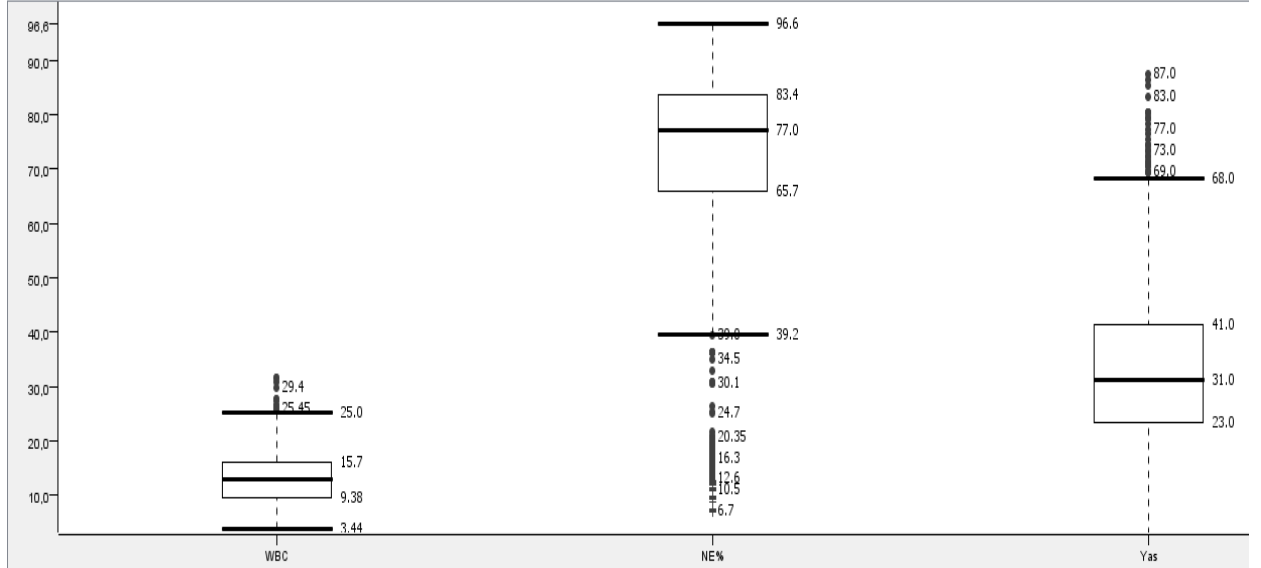
Tablo 3.1: Tahmin için kullanılan nitelikler.

TAHMIN İÇİN KULLANILAN NİTELİKLER (BAĞIMSIZ DEĞİŞKENLER)			
Nitelik Adı	Kısaltma	Veri Tipi	Kodlama Biçimi
Ağrının sağ alt kadrana yer değiştirmesi	(AgSagDeg)	İkili Kategorik (binary)	1: Var, 0: Yok
İştahsızlık	(Istah)	İkili Kategorik (binary)	1: Var, 0: Yok
Bulantı/Kusma	(BulKus)	İkili Kategorik (binary)	1: Var, 0: Yok
Sağ alt kadranda hassasiyet	(SagHas)	İkili Kategorik (binary)	1: Var, 0: Yok
Rebound ağrı	(RebAgr)	İkili Kategorik (binary)	1: Var, 0: Yok
Ateş yüksekliği	(Ates)	İkili Kategorik (binary)	>38: 1: Var, <38: 0: Yok
Lökositoz(WBC)	(Lok)	Nümerik	
Artmış nötrofil yüzde oranı (NE%)	(NE%)	Nümerik	
Yaş	(Yas)	Nümerik	
Cinsiyet	(Cnsyt)	İkili Kategorik (binary)	1: Erkek, 0:Kadın
HEDEF NİTELİK (BAĞIMLI DEĞİŞKEN)			
Apandisit Tanı	(AATani)	İkili Kategorik (binary)	0: Negatif-AA+ 1: Pozitif-AA-

Tablo 3.2: Veri setinde niteliklere ait tanımlayıcı bilgi.

Nitelik Adı	Min	Max	Mean	Std. Deviation	Variance	Skewness	Kurtosis	Median
sag_alt_kadrana_agri_gocu	0	1	0.957	0.204	0.041	-4.491	18.191	1
sag_alt_kadran_hassasiyet	0	1	0.93	0.255	0.065	-3.37	9.367	1
rebound	0	1	0.756	0.43	0.185	-1.192	-0.579	1
istahsizlik	0	1	0.403	0.491	0.241	0.396	-1.846	0
bulanti_kusma	0	1	0.423	0.494	0.244	0.312	-1.905	0
aksiler_ates	0	1	0.208	0.406	0.165	1.437	0.066	0
Yas	0	87	32.979	14.549	211.662	0.688	0.555	31
WBC	3.44	31.25	12.785	4.543	20.638	0.500	0.251	13
NE%	6.7	96.6	71.821	17.991	323.665	-1.744	3.182	77
Cinsiyet	0	1	0.391	0.488	0.238	0.446	-1.804	0

Niteliklere ait istatistiksel bilgiler Tablo 3.2’de gösterilmektedir. Verilere ait her hangi bir kayıp değeri tespit edilmemiştir.

**Şekil 3.2:** WBC, NE% ve yaş niteliklerine ait kutu grafiği çizimleri.

Nümerik niteliklerdeki ait olası uç noktalar hakkında bilgi edinmek için kutu grafiği çizimleri Şekil 3.3’te gösterilmektedir. Şekil 3.3 incelenirse ilk bakışta veri setindeki WBC, NE% ve yaş niteliklerine ait uç değerlerin varlığı, niteliklerin aldıkları değerlerin farklı aralıklarda bulunduğu göstermektedir.

Sınıflandırma modelleri sonucunda elde edilen bulgular Bölüm 4’te yer almaktadır.

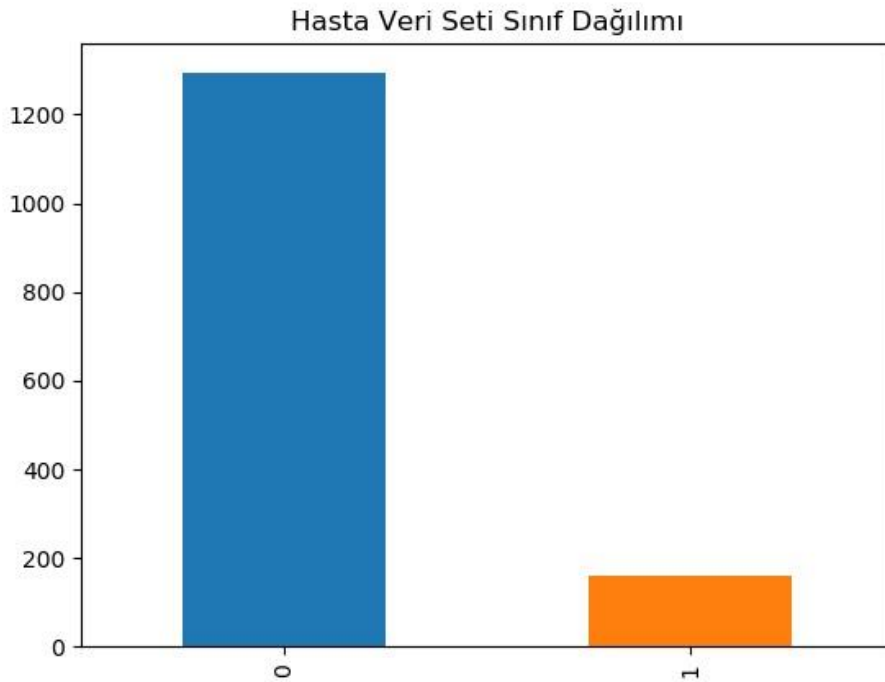
4. BULGULAR

Çalışmanın bu bölümünde Dengeleme Yöntemleri, Rasgele Orman, Destek Vektör Makinesi, Naive Bayes Algoritması, k-En Yakın Komşu, Lojistik Regresyon ve Yapay Sinir Ağları algoritmalarının akut apandisit veri setlerine uygulanması ve bu analizlerin sonucunda elde edilen bulgulara ayrıntılı bir şekilde yer verilmektedir.

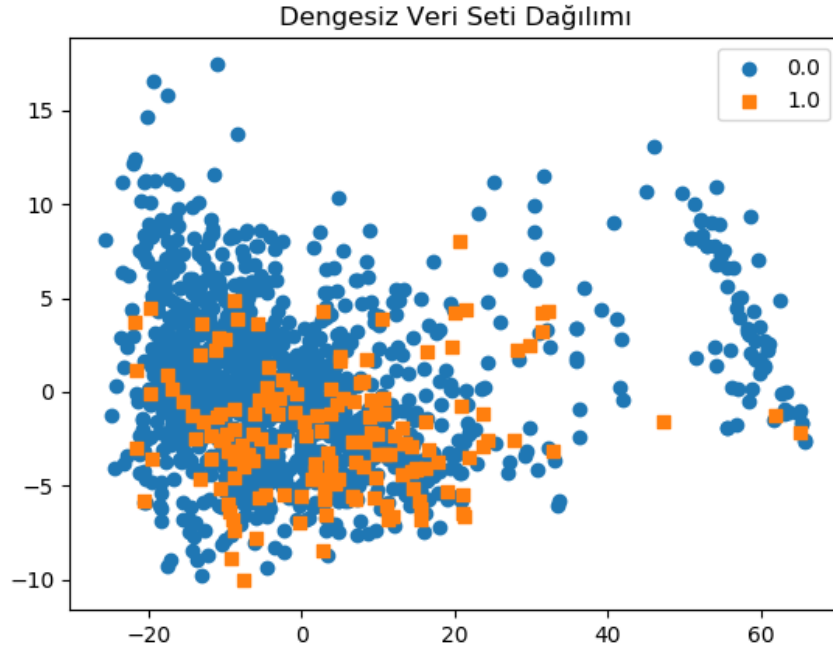
4.1. SINIF DENGELEME YÖNTEMLERİ

Bu çalışmada veri seti 1454 hastadan oluşmaktadır. Histopatolojik raporlara göre akut apandisit olan 1294 hasta 0 olarak ve akut apandisit olmayan 160 hasta ise 1 olarak etiketlenmiştir.

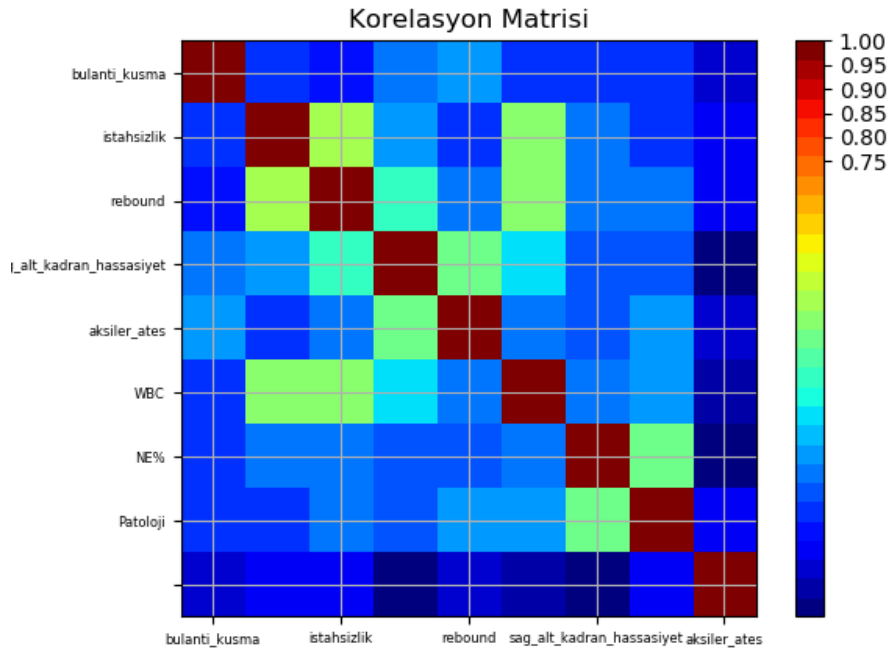
Yüksek dengesizlik veri setinin dağılımı Şekil 4.1 ve 4.2’de gösterilmektedir. Çalışmamızda sınıf dengesizliği probleminin sınıflandırma üzerindeki olumsuz etkisini azaltmak için SMOTE, Random Oversampling ve Random Undersampling algoritmaları önerilmiştir.



Şekil 4.1: Sınıf dağılımı histogram.



Şekil 4.2: Sınıf dağılımı.



Şekil 4.3: Korelasyon matrisi.

Veri setindeki özelliklerin ilişki matrisi Şekil 4.3'te verilmiştir. İlişki (korelasyon) matrisindeki koyu kırmızı renk değişkenler arasındaki güçlü pozitif ilişkiyi gösterirken koyu mavi renk ise

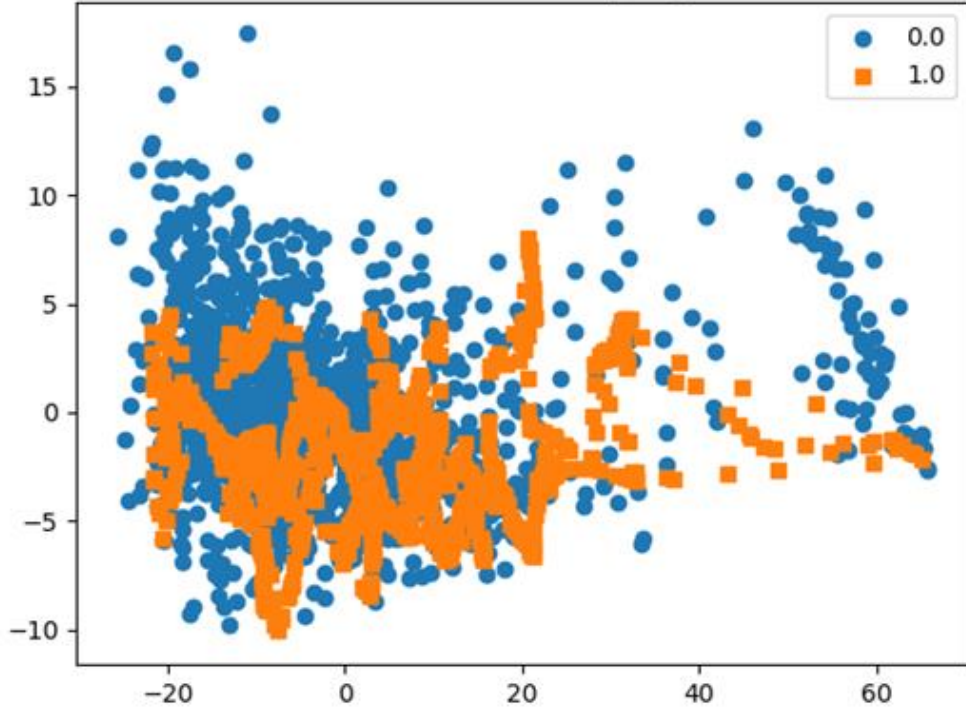
güçlü negatif ilişki olduğunu göstermektedir. Değişkenler arasındaki ilişkinin gücü arttıkça renk koyulaşmaktadır.

4.1.1. Random Over Sampling (ROS)

Rasgele aşırı örnekleme (ROS) yönteminde sınıflayıcı istenen sınıf oranına ulaşana kadar eğitirken azınlık sınıf etiketi örneklerini sayıca çok olan sınıf etiketine yaklaştırmak için rasgele çoğaltarak sınıf dağılımını dengelemek için kullanılır. 1 olarak etiketlenen 160 azınlık sınıf etiketi 0 olarak etiketlenen 1294 çoğunluk sınıf etiketine yaklaştırılmıştır.

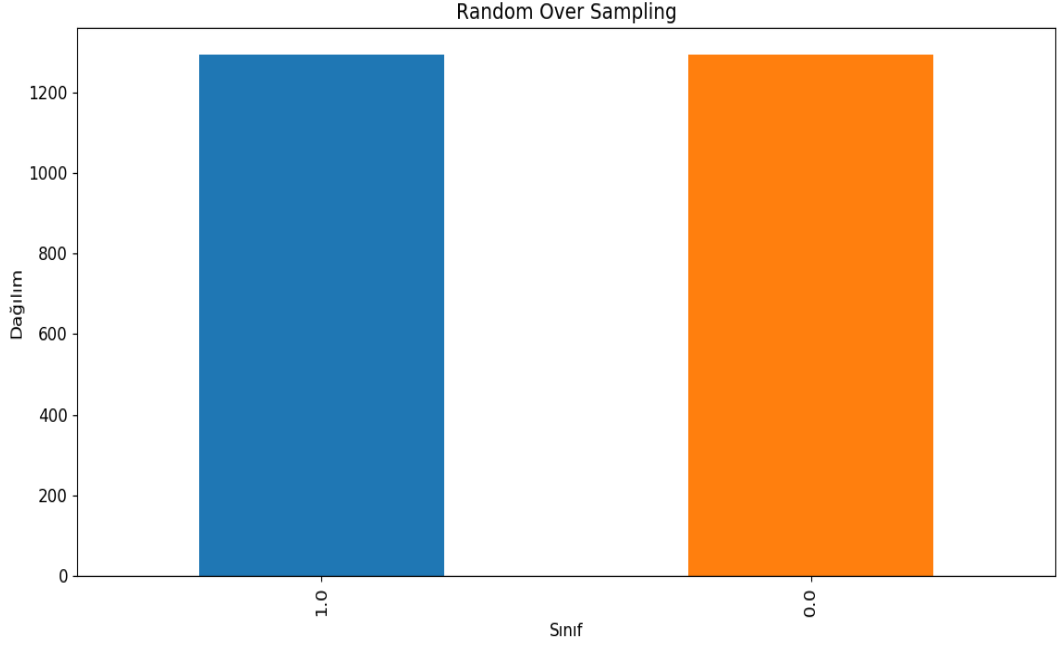
Orijinal veri dağılımı: ({0: 1294, 1: 160})

Yeniden Örneklenen veri dağılımı: ({0: 1294, 1: 1294})



Şekil 4.4: Random oversampling veri dağılımı.

Şekil 4.5'te ROS yönteminden sonraki veri dağılımı gösterilmektedir.



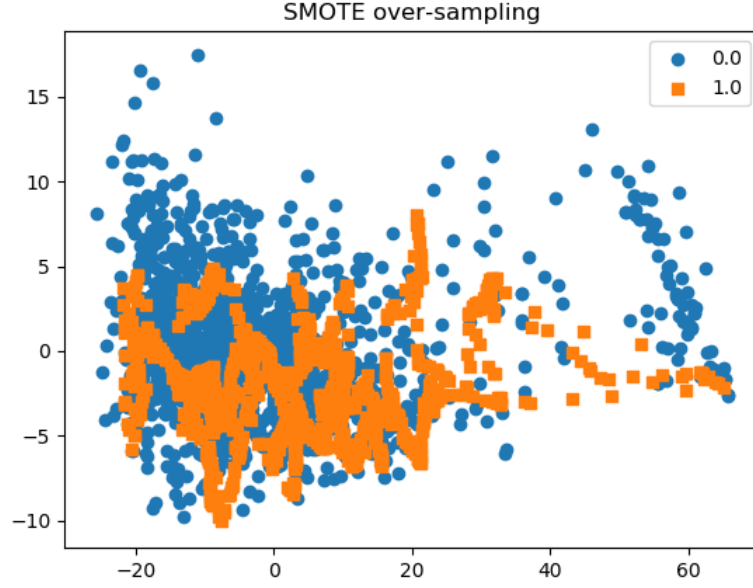
Şekil 4.5: Random oversampling veri dağılımı histogramı.

4.1.2. SMOTE

SMOTE mevcut azınlık örnekleri arasında araya girerek yeni yapay azınlık örnekleri oluşturur. Temel varsayım, birbirine yakın iki gerçek pozitif örnek arasında yapay bir pozitif örnek olduğudur. Bu nedenle, SMOTE algoritması birbirine yakın olan iki gerçek pozitif örnek arasında yapay olarak kullanılan sınıf dengeleme yöntemidir. 1 olarak etiketlenen 160 azınlık sınıf etiketi 0 olarak etiketlenen 1294 çoğunluk sınıf etiketine yaklaştırılmıştır.

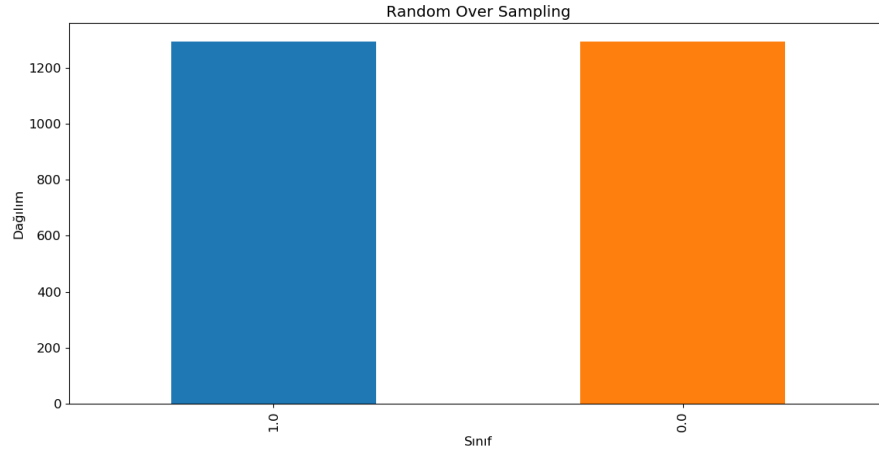
Orjinal veriseti dağılımı: ({0: 1294, 1: 160})

SMOTE veriseti dağılımı: ({0: 1294, 1: 1294})



Şekil 4.6: SMOTE veri seti sınıf dağılımı.

Şekil 4.7’de azınlık sınıfı, çoğunluk sınıfına yaklaştırıldıktan sonraki veri dağılımı gösterilmektedir.



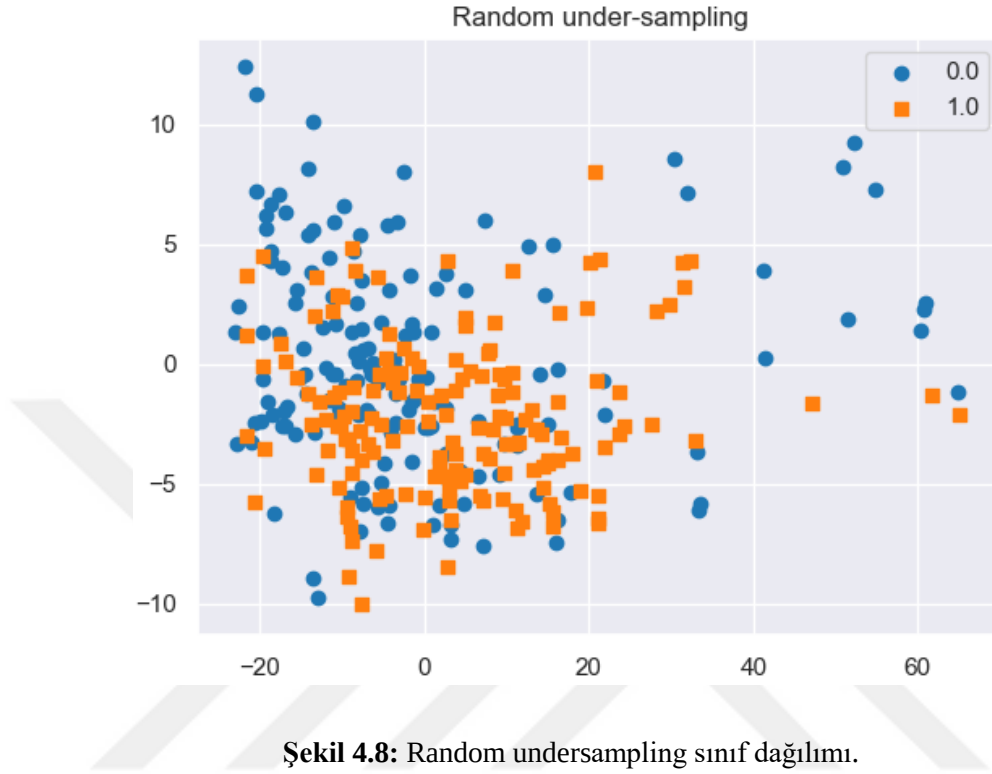
Şekil 4.7: SMOTE veri seti sınıf dağılımı histogramı.

4.1.3. Random Undersampling (RUS)

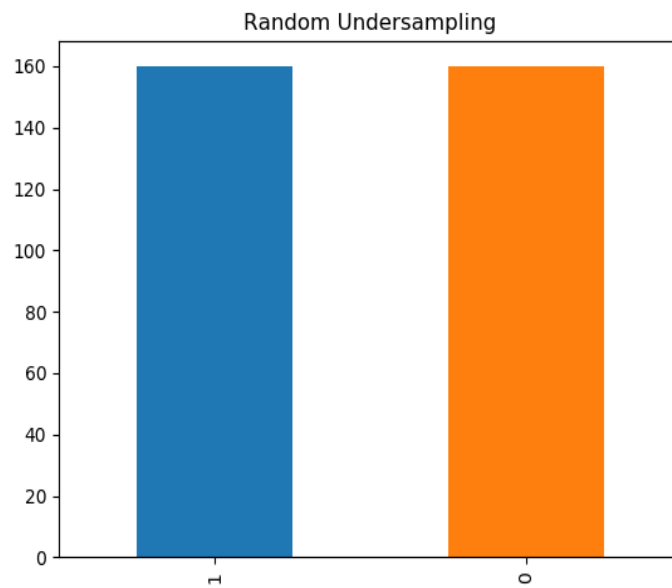
Rasgele az örneklemede çoğunluk sınıfı örneklerinin rasgele ortadan kaldırılması yoluyla sınıf dağılımını dengelemeyi amaçlayan sezgisel olmayan bir yöntemdir. 0 olarak etiketlenen 1294 çoğunluk sınıf etiketi 1 olarak etiketlenen azınlık sınıfına yaklaştırılmıştır.

Orijinal veriseti dağılımı: ({0: 1294, 1: 160})

Yeniden örnekleme veriseti dağılımı: ({0: 160, 1: 160})



Şekil 4.9'da çoğunluk sınıfı, azınlık sınıfına yaklaştırıldıktan sonraki veri dağılımı gösterilmektedir.



Şekil 4.9: Random undersampling sınıf dağılımı histogram.

4.1.4. Test ve Eğitim Veriseti

Akut Apandisit hastalık sınıflandırması yapılırken test ve eğitim veri seti holdout (75/25) ve çapraz geçerleme metodunun k-kat çapraz geçerleme (5-fold) yöntemi kullanılmıştır. Veri seti holdout yönteminde %75 eğitim kümesi, %25 test kümesi olarak ayrılmıştır. Sınıflandırma modelleri bu eğitim kümesi ile eğitilmiştir. Model başarısı daha önce %25 oranında ayrılan test veri seti ile ölçülmüştür. K-kat çapraz geçerleme için de 5-fold olarak belirlenmiştir.

4.2. ANALİZ SONUÇLARI

Tezin bu bölümünde Rasgele Orman, Destek Vektör Makinesi, Naive Bayes Sınıflandırıcı, Yapay Sinir Ağları, k-En Yakın Komşu Algoritması, Logistik Regresyon Analizi sınıflandırma algoritmalarının akut apandisit verileri üzerinde sınıflandırma uygulaması yapılmıştır ve analiz sonucunda elde edilen bulgular ayrıntılı olarak gösterilmiştir.

Sınıflandırma modellerinde Naive Bayes, k-en yakın komşu, yapay sinir ağları ve rastgele orman, destek vektör makinesi algoritmaları sıklıkla tercih edilmektedir. En iyi model başarımını test etmek için bahsedilen sınıflandırıcı model algoritmaları Python yazılımı aracılığıyla test edilmiştir.

Makine öğrenmesinde, sınıflandırma yöntemlerinde kurulan modelin performansını ölçen çeşitli ölçüler vardır. Karmaşıklık matrisi, birçok yaygın sınıflandırma ölçütünü hesaplamak için kullanılır.

Tablo 4.1: İki sınıf için oluşturulmuş örnek bir karmaşıklık matrisi.

Karmaşıklık Matrisi		Verinin Tahmin Edilen Sınıfı	
		AA+ (Akut Apandisit)	AA- (Gözlemlenmeli)
Verinin Gerçek Sınıfı	AA+ (Akut Apandisit)	Doğru Negatif (DN)	Yanlış Pozitif (YP)
	AA- (Gözlemlenmeli)	Yanlış Negatif (YN)	Doğru Pozitif (DP)

Karışıklık matrisi tablosunda yer alan DN gerçekte akut apandisit olan ve sınıflandırma sonucunda da akut apandisit olarak etiketlenen örnek sayısını, DP gerçekte akut apandisit olmayan ve sınıflandırma sonucunda da akut apandisit olmayan olarak etiketlenen örnek sayısını gösterir. YP gerçekte akut apandisit olup ve sınıflandırma sonucunda akut apandisit olmayan olarak etiketlenmiş örnek sayısını göstermektedir. YN gerçekte akut apandisit olmayan olup sınıflandırma sonucunda akut apandisit olarak etiketlenen örnek sayısını gösterir. Birçok çalışmada sadece doğruluk ölçütü değeri göz önüne alınarak en yüksek değere sahip olan model en iyi performansa sahip model olarak nitelendirilir. Oysa sınıflar arası uygun dağılım gösteren dengeli veri setinde doğruluk değeri, dengesiz veri setine göre daha iyi performans göstermektedir. Eşit sınıf dağılımına sahip veri setinde doğruluk değeri ile dengeli doğruluk değeri aynı sonuca sahip olurken, veri setinin dengesiz olması durumunda ise dengeli doğruluk değeri doğruluk değerinden daha düşüktür.

Literatürde sıklıkla kullanılan genel doğruluk oranının sınıf dengesizliğinde kullanılmaması gerektiği (çeşitli makale ve kitaplarda) vurgulanmıştır (Han, Kamber, 2006; He, Garcia, 2009; Menardi ve Torelli, 2014; Karaağaoğlu ve diğ., 2016).

Doğruluk doğru uyarı oranını verir (tüm uyarı oranlarından). Tüm olarak doğru olarak sınıflandırma oranını verir.

Seçicilik oranı gerçekte akut apandisit olan hastaların hangi oranda sınıflandırma tarafından doğru tahmin edildiğini göstermektedir. Her ne kadar akut apandisit olan hastaları doğru tahmin etmek önemli olsa da bizim için asıl önemli olan akut apandisit olabilecek hastaların tespit edilerek önlemler alınmasıdır. Bu nedenle de duyarlılık ölçüsü önemsenecek ve göz önünde bulundurulacak kıstastır.

Spesifiklik bunun yerine, doğru olarak negatif olarak sınıflandırılan gerçek olumsuz durumların oranını ifade eder.

Duyarlılık oranı gerçekte akut apandisit olmayan hastaların ne kadarının sistem tarafından akut apandisit olmayan olarak tahmin edildiğini verir. Veri setindeki 1454 hastandan 160 tanesi akut apandisit. Akut apandisit olmayan hastaların önceden tahmin edilmesi, laparotomi ameliyatı olmaması açısından fayda sağlayacaktır.

Hem duyarlılık hem de özgüllük, klinik tanımlar için bir sınıflandırmanın (tahmin) etkinliğini değerlendirmek için yaygın ölçümlerdir.

F-ölçütü, kesinlik (precision) ve duyarlılığın ağırlıklandırılmış ortalamasını göstermektedir. Başlı başına kesinlik ve duyarlılığa bakmak yerine F-ölçütüne bakmak daha doğru sonuç vermektedir.

Sınıflandırma başarısını ölçerken seçicilik değeri veya duyarlılık değerinin tek başına kullanılması uygun görülmemekte olup eksik kalabilmektedir. Bunun yerine kesinlik ve duyarlılık değerlerinin harmonik ortalaması alınarak hesaplanan F-Ölçütü değeri kullanılmaktadır.

4.2.1. RASGELE ORMAN (RANDOM FOREST)

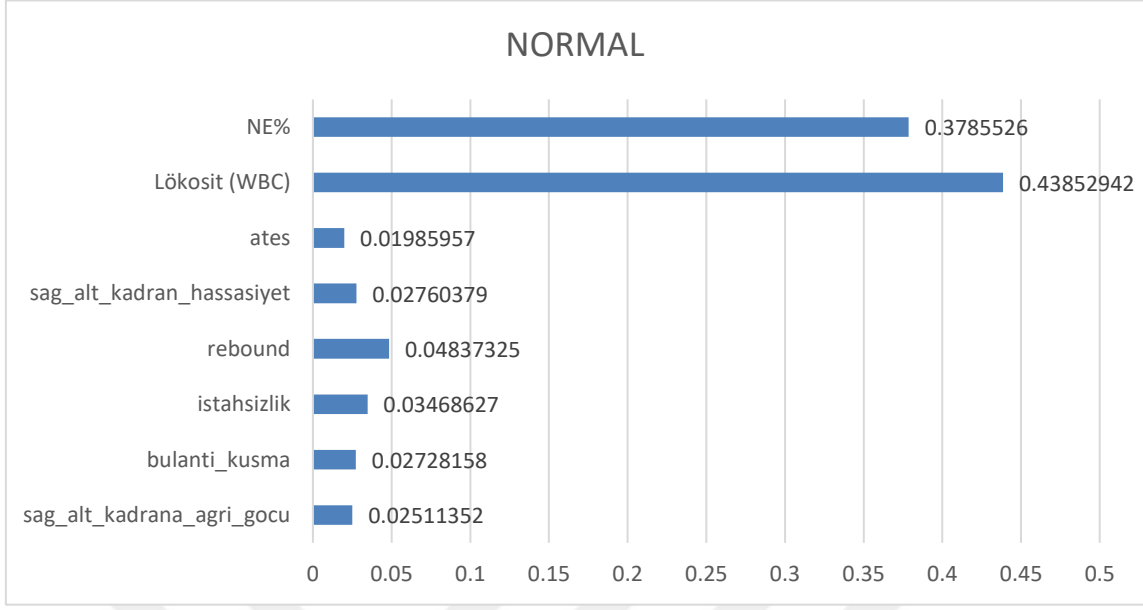
RO algoritmasını uygulamak için Python'da 'RandomForestClassifier' fonksiyonu kullanılmıştır.

100 model kullanılmıştır. Bu ağacı 100 defa tekrarlayacaktır. Kriter olarak Gini Index kullanılmıştır.

Hangi değişkenin sınıflandırma üzerine ne kadar katkı sağladığını görmek amacıyla nitelik önemi analizi yapılmıştır. Niteliklerin Gini Index değerleri aşağıdaki tabloda gösterilmektedir.

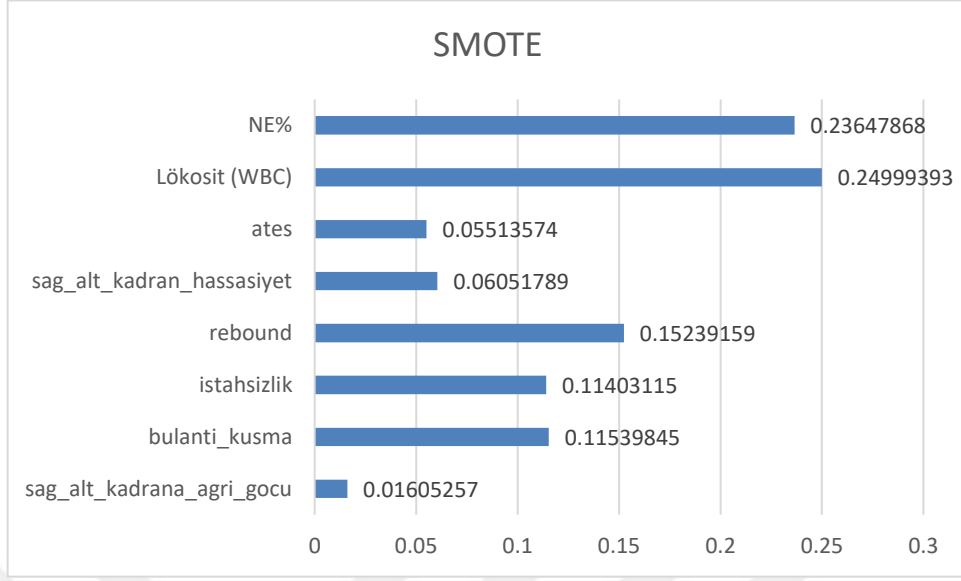
Tablo 4.2: Nitelik önemi.

NİTELİK ÖNEMİ (FEATURE IMPORTANCE)				
	NORMAL	SMOTE	ROS	RUS
sag_alt_kadrana_agri_gocu	0.02511352	0.01605257	0.01877	0.03332
bulanti_kusma	0.02728158	0.11539845	0.04071	0.04595
İstahsızlık	0.03468627	0.11403115	0.03099	0.02314
Rebound	0.04837325	0.15239159	0.05995	0.06522
sag_alt_kadran_hassasiyet	0.02760379	0.06051789	0.01566	0.02096
Ateş	0.01985957	0.05513574	0.05058	0.03152
Lökosit (WBC)	0.43852942	0.24999393	0.42294	0.41589
NE%	0.3785526	0.23647868	0.36042	0.36399



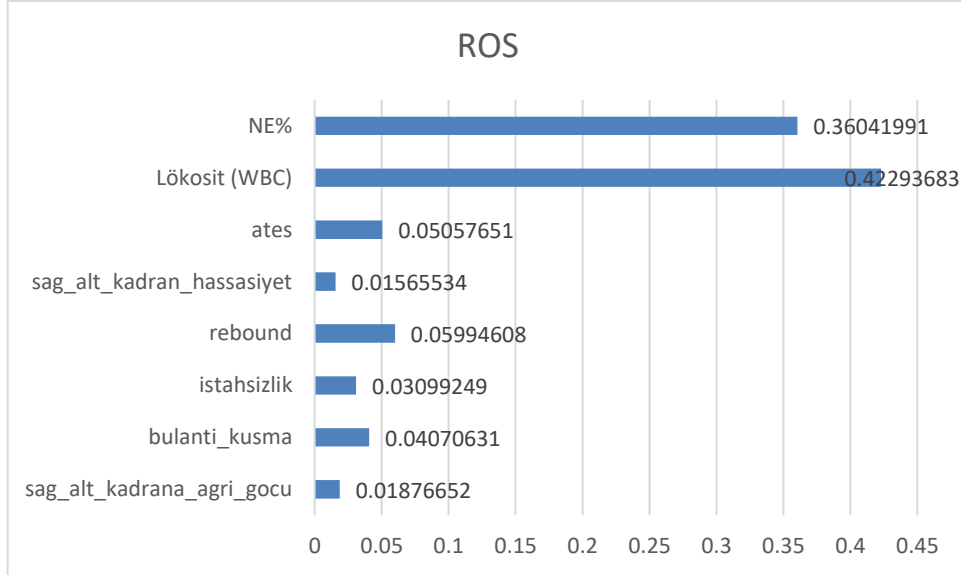
Şekil 4.10: Normal nitelik önemi.

Dengeleme yöntemi kullanılmadan niteliklerin önemine bakıldığında, Şekil 4.10'da RO ile yapılan sıralamaya göre Lökosit (WBC) değeri AA teşhisinde en önemli değişken (0.43), daha sonra NE% (0.37) ve rebound (0.04) önemlilik göstermektedirler. Ancak ateş (0.01) sınıflandırmaya çok az katkı sağladığı, sağ alt kadrana ağrı göçü (0.02), sağ alt kadranda hassasiyet (0.02) ve bulantı/kusma (0.02) değişkenleri birbirine yakın önemlilik göstermektedirler.



Şekil 4.11: SMOTE nitelik önemi.

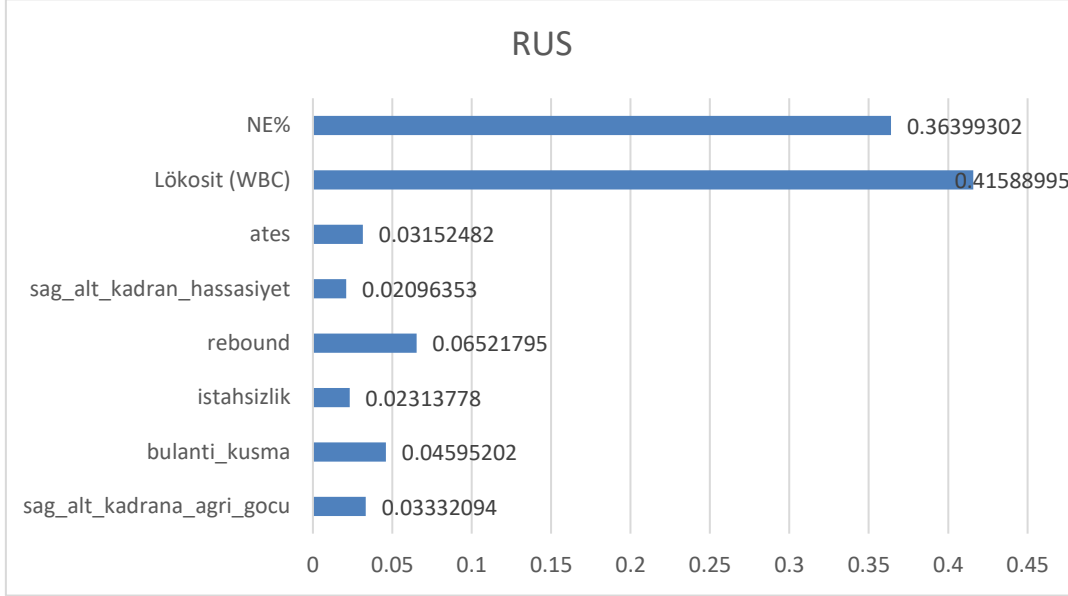
SMOTE yöntemi kullanıldıktan sonra niteliklerin önemine bakıldığında, Şekil 4.11’de RO ile yapılan sıralamaya göre Lökosit (WBC) değeri AA teşhisinde en önemli değişken (0.24), daha sonra NE% (0.23) ve rebound (0.15) önemlilik göstermektedirler. Bulantı/kusma (0.11) ve iştahsızlık (0.11) değişkenleri birbirine yakın önemlilik göstermektedirler. Sağ alt kadrana ağrı göçü değişkeninin (0.01) sınıflandırmaya çok az katkı sağladığı görülmektedir.



Şekil 4.12: ROS nitelik önemi.

ROS yöntemi kullanıldıktan sonra niteliklerin önemine bakıldığında, Şekil 4.12’de RO ile yapılan sıralamaya göre Lökosit (WBC) değeri AA teşhisinde en önemli değişken (0.42), daha

sonra NE% (0.36) ve rebound (0.05) ile Ateş (0.05) birbirine yakın önemlilik göstermektedir. Bulantı/Kusma (0.04) ve iştahsızlık (0.03) değişkenleri sınıflandırmaya az katkı sağladığı ve en az katkıyı Sağ alt kadrana ağrı göçü (0.01) ve sağ alt kadranda hassasiyet (0.01) değişkenleri sağlamaktadır.



Şekil 4.13: RUS nitelik önemi.

RUS yöntemi kullanıldıktan sonra niteliklerin önemine bakıldığında, Şekil 4.13’de RO ile yapılan sıralamaya göre Lökosit (WBC) değeri AA teşhisinde en önemli değişken (0.41), daha sonra NE% (0.36) ve rebound (0.06) gelmektedir. Bulantı/Kusma (0.04), Ateş (0.03) sağ alt kadrana ağrı göçü (0.03) birbirine yakın önemlilik göstermektedir. Sağ alt kadranda hassasiyet (0.02) ve iştahsızlık (0.02) değişkenleri sınıflandırmaya çok az katkı sağladığı görülmektedir.

Rasgele Orman sınıflandırma algoritmasının akut apandisit hasta veri setlerine uygulanması öncelikle normal haliyle daha sonra verileri dengeli hale getiren SMOTE algoritması, Random Oversampling ve Random Undersampling algoritmaları yapıldı. Aşağıdaki tablolarda Karmaşıklık tabloları gösterilmektedir.

Tablo 4.3: RO algoritması normal karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	DN-316	YP-2
	Pozitif (AA-)	YN-44	DP-2

Verileri dengeleme yöntemleri kullanılmadan önce normal Karmaşıklık tablosunu incelediğimizde DN=316, DP=2, YN=44 ve YP=2 değerlerini görmekteyiz. Doğru olarak sınıflanan hasta sayısı 318, gerçekte AA olmayan 2 hasta sınıflandırma sonucunda AA olarak, gerçekte AA olan 44 hastanın AA olmayan olarak etiketlendiğini görmekteyiz.

Tablo 4.4: RO algoritması SMOTE karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	308	10
	Pozitif (AA-)	9	37

Azınlık sınıfını yapay olarak çoğaltıp çoğunluk sınıfına yaklaştıran SMOTE yöntemini kullandıktan sonra karmaşıklık tablosunda DN=308, DP=37, YN=9 ve YP=10 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 345'tir. 10 hastanın gerçekte AA olup sınıflandırma sonucunda AA olmayan, 9 hastanın AA olmayan olup sınıflandırma sonucu AA olarak etiketlendiğini görmekteyiz. Sınıflandırılan hata sayısı azalmaktadır.

Tablo 4.5: RO algoritması ROS karmaşıklık tablosu.

		Tahmin Edilen	
		Pozitif (AA+)	Negatif (AA-)
Gerçek Sınıf	Negatif (AA+)	308	10
	Pozitif (AA-)	0	46

Verileri rasgele seçip azınlık sınıfından çoğunluk sınıfına yaklaştıran yöntemlerden biri olan Random Oversampling yöntemini kullandıktan sonra Karmaşıklık tablosunda DN=308, DP=46, YN=0 ve YP=10 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 354'tür. 10 hastanın gerçekte AA olup sınıflandırma sonucunda AA olamayan olarak etiketlendiğini görmekteyiz. Sınıflandırılan hata sayısı azalmaktadır.

Tablo 4.6: RO algoritması RUS karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	228	90
	Pozitif (AA-)	4	42

Verileri rasgele seçip çoğunluk sınıfından eksiltip azınlık sınıfına yaklaştıran yöntemlerden biri olan Random Undersampling algoritmasını kullandıktan sonra Karmaşıklık tablosunda

DN=228, DP=42, YN=4 ve YP=90 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 270'dir. 90 hastanın gerçekte AA olup sınıflandırma sonucunda AA olmayan olarak ve 4 hastanın gerçekte AA olmayan olup sınıflandırma sonucunda AA olarak etiketlendiğini görmekteyiz. RUS algoritması kullanıldığında sınıflandırılan hata sayısı artmaktadır.

Rasgele Orman sınıflandırma algoritmasından holdout yöntemi ile elde edilen bulgular aşağıdaki tabloda gösterilmektedir.

Tablo 4.7: RO algoritması holdout metrikleri.

	Normal	SMOTE	ROS	RUS
Doğruluk	0.87362	0.94780	0.97252	0.74175
Kesinlik	0.5	0.78723	0.82142	0.31818
Duyarlılık	0.04347	0.80434	1.0	0.91304
F-Ölçütü	0.08	0.79569	0.90196	0.47191
Kappa Değeri	0.0610	0.76578	0.88616	0.35009

Rasgele Orman Algoritmasının uygulanması sonucunda genel doğruluk oranları Normal, SMOTE, ROS ve RUS için sırasıyla 0.87, 0.94, 0.97 ve 0.74'tür. En yüksek genel doğruluk oranı ROS algoritmasına aittir. Birçok çalışmada sadece doğruluk ölçütü değeri göz önüne alınarak en yüksek değere sahip olan model en iyi performansa sahip model olarak nitelendirilir. Genel doğruluk oranının sınıf dengesizliğinde kullanılmaması gerektiği için bunun yanında diğer metrikler de göz önünde bulundurulmalıdır.

Kesinlik değeri akut apandisit olmayan olarak doğru sınıflandırılmış örnek sayısının model tarafından hasta olmayan olarak sınıflandırılan toplam örnek sayısına oranıdır. Kesinlik değeri Normal, SMOTE, ROS ve RUS için sırasıyla 0.5, 0.78, 0.82 ve 0.31'dir. En yüksek kesinlik değeri ROS algoritmasına aittir.

Bunun yanında duyarlılık ölçüsü önemsenerek ve göz önünde bulundurulacak kıstastır. Duyarlılık testi akut apandisit olmayanlar içinden akut apandisit hasta olmayanları ayırma yeteneğidir. Yani gerçekte akut apandisit hasta olmayanların ne kadarının sistem tarafından

doğru tahmin edildiğini verir. Dengesiz verilerde Duyarlılık oranı 0.04 iken SMOTE için 0.80, ROS için 1.0 ve RUS algoritmasında bu değer 0.91 olarak hesaplanmıştır. En yüksek duyarlılık değeri ROS algoritmasına aittir.

F-ölçütü, kesinlik ve duyarlılığın ağırlıklandırılmış ortalamasını göstermektedir. Başlı başına kesinlik ve duyarlılığa bakmak yerine F-ölçütüne bakmak daha doğru sonuç vermektedir. F-Ölçütü değeri, kesinlik ve duyarlılık değerlerinin harmonik ortalaması alınarak hesaplanır. F-Ölçütü oranının Normal veri seti için 0.08, SMOTE için 0.79, ROS için 0.90 ve RUS için 0.47 olduğunu görmekteyiz. En yüksek F-Ölçütü değeri SMOTE algoritmasına aittir. Normal veride bu ölçüt 0.08'den 0.90'a yükselmesi yeniden örnekleme yönteminin başarılı olduğunu göstermektedir.

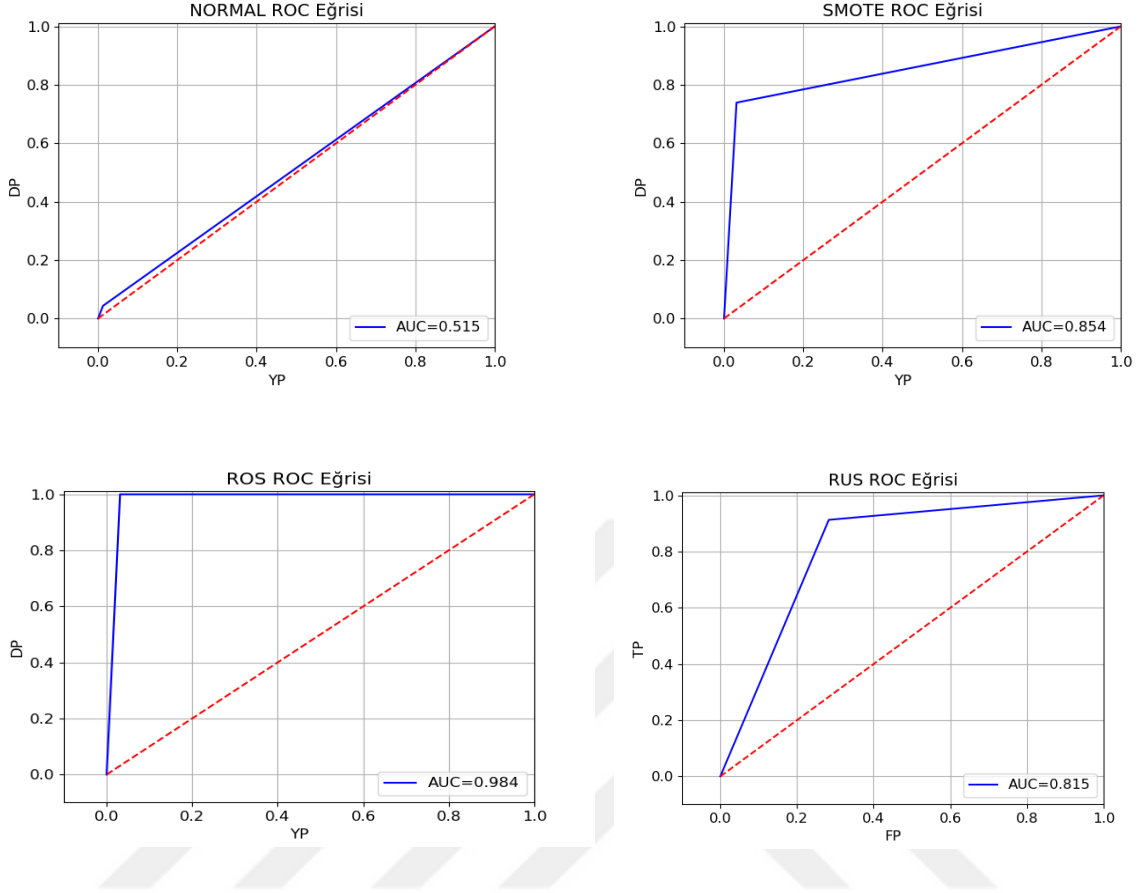
Kappa değeri Normal veri seti için 0.06, SMOTE için 0.76, ROS için 0.88 ve RUS algoritması için 0.35 oranlarını görmekteyiz. Kappa değeri ROS algoritması için neredeyse mükemmel olarak yorumlanabilmektedir.

Tablo 4.8: RO algoritması 5 kat çapraz geçişleme sonuçları.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.87889	0.90731	0.94318	0.5564
Kesinlik	0.22380	0.72050	0.75748	0.5222
Duyarlılık	0.06606	0.78419	0.99534	0.2872
F-Ölçütü	0.10048	0.74507	0.84010	0.3673

5 kat çapraz geçişleme sonuçlarına bakıldığında oranlarda düşüş olduğu görülmektedir. Doğruluk oranı Normal için 0.87, SMOTE için 0.90, ROS için 0.94 ve RUS için 0.55'dir. Kesinlik oranı Normal için 0.22, SMOTE için 0.72, ROS için 0.75 ve RUS için 0.52'dir. Duyarlılık oranı Normal için 0.06, SMOTE için 0.15, ROS için 0.20 ve RUS için 0.62'dir. F-Ölçütü Normal için 0.1, SMOTE için 0.78, ROS için 0.99 ve RUS için 0.28'dir. En Yüksek F-Ölçütü ve doğruluk oranı ROS yöntemine aittir.

Özellikle sağlık alanında yapılan modellerin değerlendirilmesinde sıklıkla kullanılan AUC kriteri sonuçları Şekil 4.14'te gösterilmiştir.



Şekil 4.14: RO sınıflandırma ROC eğrileri.

AUC değeri Normal veri seti için 0.51, SMOTE için 0.85, ROS için 0.98 ve RUS için 0.81'dir. En iyi performansı ROS yöntemi göstermektedir.

4.2.2.DESTEK VEKTÖR MAKİNESİ (DVM)

DVM algoritmasının uygulanmasında, en iyi sınıflandırma performansını veren parametre değerleri ayarlanmıştır (parameter tuning). Yapılan denemeler sonucunda grid yöntemi kullanılarak en ideal çekirdek ve nitelik ayarlaması yapılmıştır. DVM modelini oluşturma sürecinde Radyal temelli çekirdeği çekirdek işlevi olarak kullanılmıştır, gamma değeri (γ) 1 ve C (maliyet parametresi) 1 olarak ayarlanmıştır.

```
classifier = svm.SVC(kernel='rbf', gamma=1, C=1, random_state = 0)
```

DVM veri seti için normal haliyle, SMOTE algoritması, Random Oversampling ve Random Undersampling algoritmaları ile uygulanmıştır. Aşağıdaki tablolarda Karmaşıklık tabloları gösterilmektedir.

Tablo 4.9: DVM algoritması normal karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	DN-318	YP-0
	Pozitif (AA-)	YN-46	DP-0

Verileri dengeleme yöntemleri kullanmadan normal Karmaşıklık tablosunu incelediğimizde DN=318, DP=0, YN=46 ve YP=0 değerlerini görmekteyiz. Doğru olarak sınıflanan hasta sayısı 318, gerçekte akut apandisit olmayan 46 hasta sınıflandırma sonucunda akut apandisit olarak etiketlendiğini görmekteyiz.

Tablo 4.10: DVM algoritması SMOTE karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	DN-204	YP-114
	Pozitif (AA-)	YN-16	DP-30

Verilerden azınlık sınıfı yapay olarak çoğaltıp çoğunluk sınıfına yaklaştıran SMOTE yöntemi kullandıktan sonra Karmaşıklık tablosunda DN=204, DP=30, YN=16 ve YP=2 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 234'tür. 114 hastanın gerçekte AA olup sınıflandırma sonucunda akut apandisit olmayan 16 hastanın gerçekte AA olmayan olup sınıflandırma sonucunda AA olan olarak etiketlendiğini görmekteyiz.

Tablo 4.11: DVM algoritması ROS karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	303	15
	Pozitif (AA-)	1	45

Verileri rasgele seçip azınlık sınıfından çoğunluk sınıfına yaklaştıran yöntemlerden biri olan Random Oversampling yöntemi kullanıldıktan sonra Karmaşıklık tablosunda DN=303, DP=45, YN=1 ve YP=15 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 348'dir. 15 hastanın gerçekte AA olup sınıflandırma sonucunda AA olmayan ve 1 hastanın gerçekte AA olmayan olup sınıflandırma sonucunda AA olan olarak etiketlendiğini görmekteyiz.

Tablo 4.12: DVM algoritması RUS karmaşıklık tablosu.

		Tahmin Edilen	
		Pozitif (AA+)	Negatif (AA-)
Gerçek Sınıf	Pozitif (AA+)	233	85
	Negatif (AA-)	5	41

Verileri rasgele seçip çoğunluk sınıfından eksiltip azınlık sınıfına yaklaştıran yöntemlerden biri olan Random Undersampling yöntemini kullandıktan sonra Karmaşıklık tablosunda DN=233, DP=41, YN=5 ve YP=85 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 274'tür. 85 hastanın gerçekte AA olup sınıflandırma sonucunda AA olmayan olarak ve 5 hastanın gerçekte AA olmayan olup sınıflandırma sonucunda AA olan etiketlendiğini görmekteyiz.

Tablo 4.13: DVM algoritması metrikleri.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8736	0.9313	0.9560	0.7527
Kesinlik	0.5	0.6521	0.75	0.3253
Duyarlılık	0.0217	0.9782	0.9782	0.8913
F-Ölçütü	0.0416	0.7826	0.8490	0.4767
Kappa Değeri	0.0314	0.7437	0.8238	0.3578

Doğruluk değeri Normal, SMOTE, ROS, RUS için sırasıyla 0.87, 0.93, 0.95 ve 0.75'tir. En yüksek doğruluk değerleri ROS algoritmalarına aittir.

Kesinlik değeri Normal, SMOTE, ROS, RUS için sırasıyla 0.5, 0.65, 0.75 ve 0.32'dir. En yüksek kesinlik değerleri ROS' aittir.

Duyarlılık oranı Normal verilerde 0.02, SMOTE ve ROS için 0.97 ve RUS için 0.89 olarak hesaplanmıştır. En yüksek duyarlılık değeri SMOTE, ROS algoritmalarına aittir.

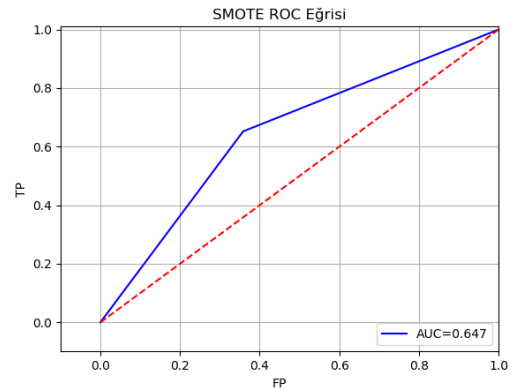
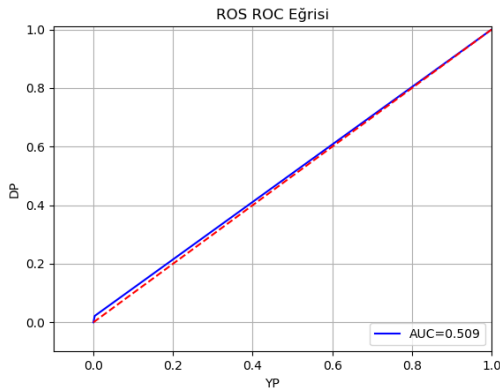
F-Ölçütü oranının Normal veri seti için 0.04, SMOTE için 0.78, ROS için 0.84 ve RUS için 0.47 olduğunu görmekteyiz. En yüksek F-Ölçütü değerleri ROS yöntemine aittir.

Kappa değeri Normal veri seti için 0.03, SMOTE için 0.74, ROS için 0.82 ve RUS için 0.35 olduğunu görmekteyiz. En yüksek F-Ölçütü değerleri ROS yöntemine aittir.

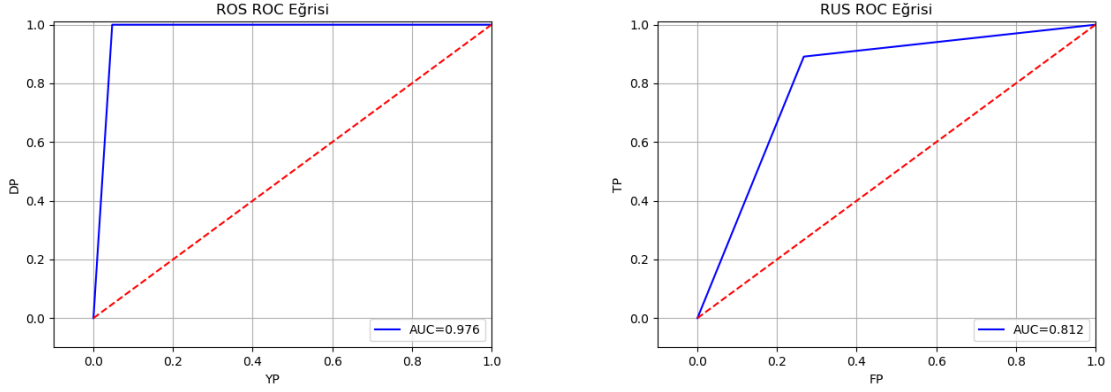
Tablo 4.14: DVM algoritması 5 kat çapraz geçерleme metrikleri.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8935	0.8919	0.9651	0.8806
Kesinlik	0.1	0.7541	0.8590	0.7374
Duyarlılık	0.0074	0.9314	0.9856	0.9295
F-Ölçütü	0.0137	0.8086	0.9115	0.7946

Doğruluk oranı Normal için 0.893, SMOTE için 0.8919, ROS için 0.96 ve RUS için 0.88'dir. Kesinlik oranı Normal için 0.1, SMOTE için 0.75, ROS için 0.85 ve RUS için 0.73'tür. Duyarlılık oranı Normal için 0.007, SMOTE için 0.93, ROS için 0.98 ve RUS için 0.92'dir. F-Ölçütü Normal için 0.01, SMOTE için 0.80, ROS için 0.91 ve RUS için 0.79'dur. En yüksek doğruluk ve F-Ölçütü ROS yöntemine aittir.



Şekil 4.15: DVM algoritması ROC eğrileri.



Şekil 4.16 (devam): DVM algoritması ROC eğrileri.

AUC değeri Normal veri seti için 0.50, SMOTE için 0.64, ROS için 0.97 ve RUS için 0.81'dir. En iyi performansı ROS yöntemi göstermektedir.

4.2.3. NAİVE BAYES ALGORİTMASI

Naive Bayes (NB) yöntemini uygulamak için Python'da Gauss Algoritmasına dayanan 'GaussianNB' fonksiyonu kullanılmıştır.

```
classifier=GaussianNB
```

Naive Bayes algoritması veri seti için için normal haliyle, SMOTE algoritması, Random Oversampling ve Random Undersampling algoritmaları ile uygulandı. Aşağıdaki tablolarda Karmaşıklık tabloları gösterilmektedir.

Tablo 4.15: NB algoritması normal karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	DN-288	YP-30
	Pozitif (AA-)	YN-40	DP-6

Verileri dengeleme yöntemleri kullanmadan normal Karmaşıklık tablosunu incelediğimizde DN=288, DP=6, YN=40 ve YP=30 değerlerini görmekteyiz. Doğru olarak sınıflanan hasta sayısı 294, gerçekte akut apandisit olmayan 40 hasta sınıflandırma sonucunda akut apandisit olarak etiketlendiğini, gerçekte akut apandisit olan 30 hasta sınıflandırma sonucunda akut apandisit olmayan olarak etiketlenmiş olduğunu görmekteyiz.

Tablo 4.16: NB algoritması SMOTE karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	191	127
	Pozitif (AA-)	13	33

Verilerden azınlık sınıfı yapay olarak çoğaltıp çoğunluk sınıfına yaklaştıran SMOTE algoritmasını kullandıktan sonra Karmaşıklık tablosunda DN=191, DP=33, YN=13 ve YP=127 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 224'tür. Gerçekte AA olan 127 hasta sınıflandırma sonucunda AA olmayan olarak, gerçekte AA olmayan 13 hasta sınıflandırma sonucunda AA olan olarak etiketlenmiş olduğunu görmekteyiz.

Tablo 4.17: NB algoritması ROS karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	223	95
	Pozitif (AA-)	21	25

Verileri rasgele seçip azınlık sınıfından çoğunluk sınıfına yaklaştıran yöntemlerden biri olan Random Oversampling yöntemini kullandıktan sonra Karmaşıklık tablosunda DN=223, DP=25, YN=21 ve YP=95 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 248'dir. 95 hastanın gerçekte AA olup sınıflandırma sonucunda AA olmayan olarak ve 21 hastanın gerçekte AA olmayan olup sınıflandırma sonucunda AA olan olarak etiketlendiğini görmekteyiz.

Tablo 4.18: NB algoritması RUS karmaşıklık tablosu.

		Tahmin Edilen	
		Pozitif (AA+)	Negatif (AA-)
Gerçek Sınıf	Pozitif (AA+)	221	97
	Negatif (AA-)	20	26

Verileri rasgele seçip çoğunluk sınıfından eksiltip azınlık sınıfına yaklaştıran yöntemlerden biri olan Random Undersampling yöntemini kullandıktan sonra Karmaşıklık tablosunda DN=221, DP=26, YN=20 ve YP=26 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 247'dir. 97 hastanın gerçekte AA olup sınıflandırma sonucunda akut apandisit olmayan olarak

etiketlendiğini görmekteyiz. 20 hastanın gerçekte AA olmayan olup sınıflandırma sonucunda akut apandisit olarak etiketlendiğini görmekteyiz.

Tablo 4.19: NB algoritması holdout metrikleri.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8076	0.6153	0.6840	0.6785
Kesinlik	0.1666	0.2062	0.2063	0.2113
Duyarlılık	0.1304	0.7173	0.5652	0.5652
F-Ölçütü	0.1463	0.3203	0.2857	0.3076
Kappa Değeri	0.0397	0.1543	0.1283	0.1516

Doğruluk değeri Normal, SMOTE, ROS, RUS için sırasıyla 0.80, 0.61, 0.68 ve 0.67'dir. En yüksek doğruluk değerleri Normal veri setine aittir.

Kesinlik değeri Normal, SMOTE, ROS, RUS ve Alvarado için sırasıyla 0.16, 0.20, 0.20 ve 0.21'dir. En yüksek kesinlik değerleri SMOTE ve ROS yöntemlerine aittir.

Duyarlılık oranı Normal verilerde 0.13 iken SMOTE için 0.71, ROS ve RUS için 0.56 olarak hesaplanmıştır. En yüksek duyarlılık değeri SMOTE yöntemine aittir.

F-Ölçütü oranının Normal veri seti için 0.14, SMOTE için 0.32, ROS için 0.28 ve RUS için 0.30 olduğunu görmekteyiz. En yüksek F-Ölçütü değeri SMOTE yöntemine aittir.

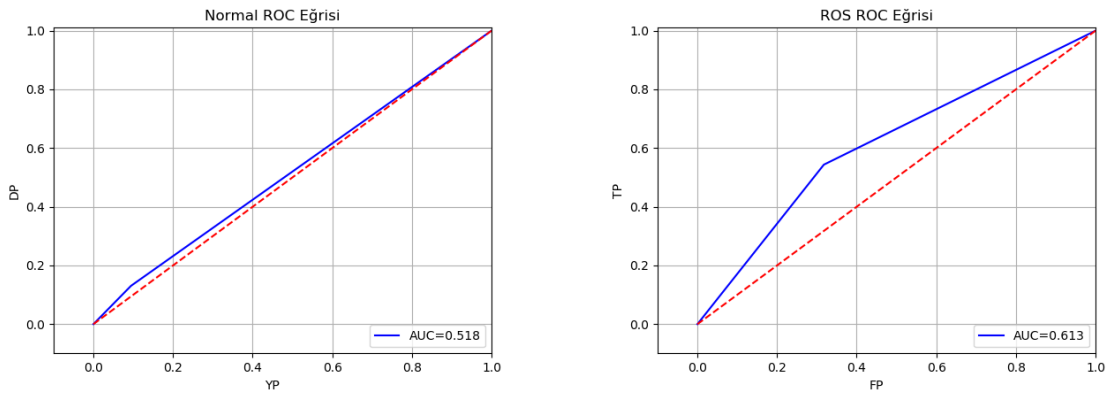
Kappa değeri Normal veri seti için 0.03, SMOTE için 0.15, ROS için 0.12 ve RUS yöntemi için 0.15 oranlarını görmekteyiz. Kappa değeri tüm yöntemler için oldukça düşüktür.

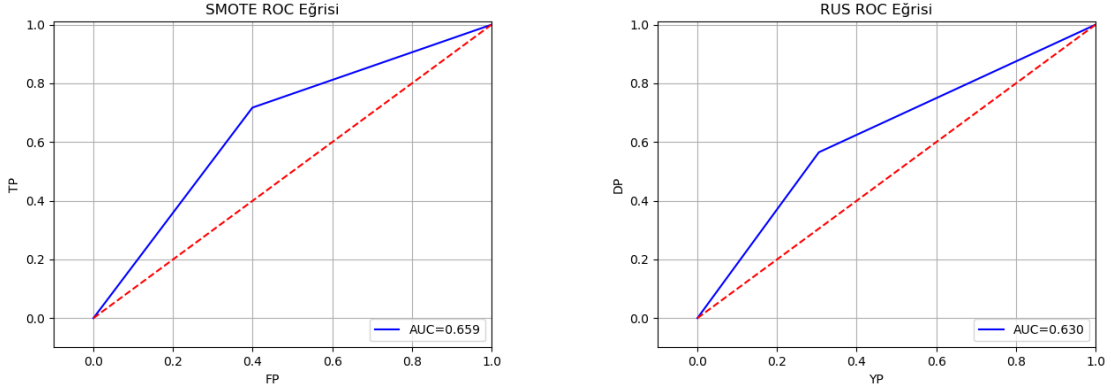
Tablo 4.20: NB algoritması 5 kat çaprazlama sonuçları.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8440	0.6477	0.7045	0.6926
Kesinlik	0.2724	0.1894	0.2149	0.2168
Duyarlılık	0.2946	0.7313	0.6762	0.5711
F-Ölçütü	0.2789	0.2992	0.3225	0.2603
AUC	0.6016	0.6847	0.6923	0.6410

Doğruluk oranı Normal için 0.84, SMOTE için 0.64, ROS için 0.70 ve RUS için 0.69'dur. Kesinlik oranı Normal için 0.27, SMOTE için 0.18, ROS için 0.214 ve RUS için 0.216'dır. Duyarlılık oranı Normal için 0.29, SMOTE için 0.73, ROS için 0.67 ve RUS için 0.57'dir. F-Ölçütü Normal için 0.27, SMOTE için 0.29, ROS için 0.32 ve RUS için 0.26'dır. En Yüksek F-Ölçütü ROS yöntemine ve doğruluk oranı Normal veri setine aittir.

Özellikle sağlık alanında yapılan modellerin değerlendirilmesinde sıklıkla kullanılan AUC kriteri sonuçları Şekil 4.16'da gösterilmiştir.

**Şekil 4.17:** Naive bayes algoritması ROC eğrileri.



Şekil 4.18 (devam): Naive bayes algoritması ROC eğrileri.

AUC değeri Normal veri seti için 0.51, SMOTE için 0.65, ROS için 0.61 ve RUS için 0.63'tür.

4.2.4.K- EN YAKIN KOMŞU (K-NEAREST NEIGHBOR) ALGORİTMASI

K-En Yakın Komşu (K-EY) algoritmasını uygulamak için Python'da 'KNeighborsClassifier' fonksiyonu için k değerleri sırasıyla 3, 5 ve 7 kullanılmış olup en iyi değeri sonucu veren k değeri seçilmiştir.

```
knn = KNeighborsClassifier(n_neighbors=5)
```

K-En Yakın Komşu algoritması veri seti için normal haliyle, SMOTE algoritması, Random Oversampling ve Random Undersampling algoritmaları ile uygulandı. Aşağıdaki tablolarda Karmaşıklık tabloları gösterilmektedir.

Tablo 4.21: K-EY algoritması normal karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	DN-312	YP-8
	Pozitif (AA-)	YN-44	DP-0

Verileri dengeleme yöntemleri kullanmadan normal Karmaşıklık tablosunu incelediğimizde DN=312, DP=0, YN=44 ve YP=8 değerlerini görmekteyiz. Doğru olarak sınıflanan hasta sayısı 312, gerçekte AA olmayan 44 hasta sınıflandırma sonucunda AA olarak etiketlendiğini ve gerçekte AA olan 8 hasta sınıflandırma sonucunda AA olmayan olarak etiketlendiğini görmekteyiz.

Tablo 4.22: K-EY algoritması SMOTE karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	231	87
	Pozitif (AA-)	2	44

Verilerden azınlık sınıfı yapay olarak çoğaltıp çoğunluk sınıfına yaklaştıran SMOTE algoritmasını kullandıktan sonra Karmaşıklık tablosunda DN=231, DP=44, YN=2 ve YP=87 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 275'tir. Gerçekte AA olan 87 hasta sınıflandırma sonucunda AA olmayan olarak, gerçekte AA olmayan 2 hastanın sınıflandırma sonucunda AA olan olarak etiketlenmiş olduğunu görmekteyiz.

Tablo 4.23: K-EY algoritması ROS karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	248	70
	Pozitif (AA-)	4	42

Verileri rasgele seçip azınlık sınıfından çoğunluk sınıfına yaklaştıran yöntemlerden biri olan Random Oversampling algoritmasını kullandıktan sonra Karmaşıklık tablosunda DN=248, DP=42, YN=4 ve YP=70 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 290'dır. Gerçekte AA olan 70 hasta sınıflandırma sonucunda AA olmayan olarak ve gerçekte AA olmayan 4 hasta sınıflandırma sonucunda AA olan olarak etiketlenmiş olduğunu görmekteyiz.

Tablo 4.24: K-EY algoritması RUS karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	193	125
	Pozitif (AA-)	12	34

Verileri rasgele seçip çoğunluk sınıfından eksiltip azınlık sınıfına yaklaştıran yöntemlerden biri olan Random Undersampling algoritmasını kullandıktan sonra Karmaşıklık tablosunda DN=193, DP=34, YN=12 ve YP=125 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 227'dir. 125 hastanın gerçekte AA olup sınıflandırma sonucunda AA olmayan olarak ve 12 hastanın gerçekte AA olmayan olup sınıflandırma sonucunda AA olan olarak etiketlendiğini görmekteyiz.

Tablo 4.25: K-EY algoritması holdut metrikleri.

		Doğruluk	Kesinlik	Duyarlılık	F-Ölçütü	Kappa Değeri
SMOTE	n=3	0.8324	0.4285	0.9782	0.596	0.5098
	n=5	0.7554	0.3358	0.9565	0.4971	0.3814
	n=7	0.728	0.312	0.9565	0.4705	0.3459
ROS	n=3	0.8708	0.4945	0.9782	0.6569	0.5877
	n=5	0.7967	0.375	0.913	0.5316	0.4294
	n=7	0.7391	0.3185	0.9347	0.4751	0.3532
RUS	n=3	0.6373	0.2133	0.6956	0.3265	0.165
	n=5	0.62362	0.2138	0.7391	0.3317	0.1687
	n=7	0.60366	0.1583	0.6512	0.2536	0.6242
NORMAL	n=3	0.8379	0.1578	0.0652	0.0923	0.0198
	n=5	0.8571	0.2	0.0434	0.0714	0.0275
	n=7	0.8708	0.3333	0.02173	0.0408	0.0257

K-EY algoritması n=3, 5 ve 7 için ayrı ayrı uygulanmış olup doğruluk, kesinlik, duyarlılık ve f-ölçütü değerlerinin en yüksek oranları ROS yönteminde n=3 olarak tespit edilmiştir. Doğruluk oranı 0.87, kesinlik oranı 0.49, duyarlılık oranı 0.97, f-ölçütü 0.65 ve kappa değeri 0.58 olduğu görülmektedir.

Doğruluk değeri SMOTE, ROS, RUS ve Normal için sırasıyla 0.8325, 0.87, 0.63 ve 0.8379'dur. En yüksek doğruluk değerleri Normal veri setine aittir.

Kesinlik değeri SMOTE, ROS, RUS ve Normal için sırasıyla 0.42, 0.49, 0.21 ve 0.15'tir. En yüksek kesinlik değerleri SMOTE yöntemine aittir.

Duyarlılık oranı Normal verilerde 0.06 iken SMOTE ve ROS için 0.97 ve RUS için 0.69 olarak hesaplanmıştır. En yüksek duyarlılık değeri SMOTE ve ROS yöntemlerine aittir.

F-Ölçütü oranının Normal veri seti için 0.14, SMOTE için 0.59, ROS için 0.65 ve RUS için 0.32 olduğunu görmekteyiz. En yüksek F-Ölçütü değeri ROS yöntemine aittir.

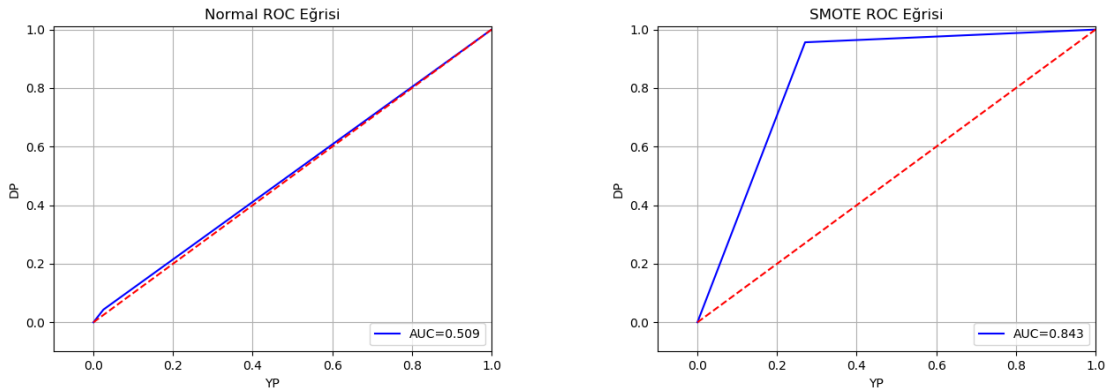
Kappa değeri Normal veri seti için 0.01, SMOTE için 0.50, ROS için 0.58 ve RUS yöntemi için 0.16 oranlarını görmekteyiz. Kappa değeri ROS için normal düzeydedir.

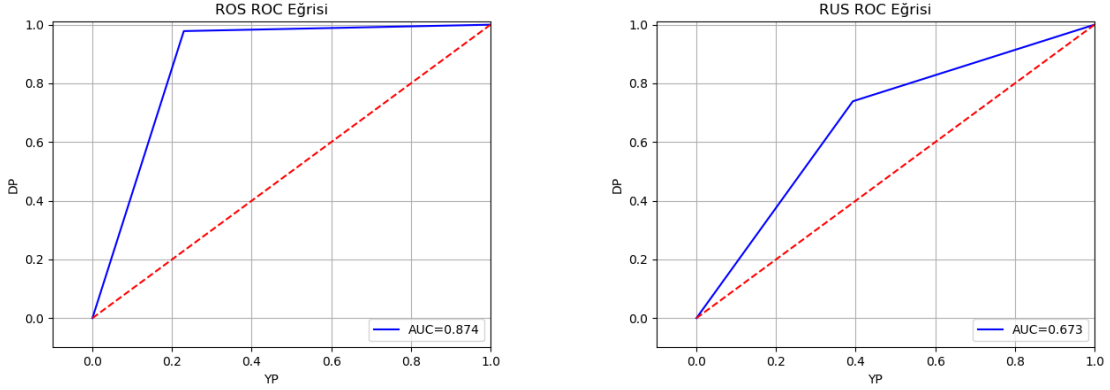
Tablo 4.26: K-EY algoritması 5 kat çapraz geçerleme sonuçları.

		Doğruluk	Kesinlik	Duyarlılık	F-Ölçütü	Kappa Değeri
SMOTE	n=3	0.7275	0.1854	0.4862	0.2651	0.6218
	n=5	0.7165	0.1955	0.5623	0.288	0.6489
	n=7	0.6871	0.1725	0.5316	0.2582	0.6188
ROS	n=3	0.7853	0.2132	0.3818	0.26847	0.6076
	n=5	0.7467	0.1981	0.4712	0.2753	0.6256
	n=7	0.6899	0.1682	0.4956	0.2486	0.6045
RUS	n=3	0.6321	0.1717	0.6629	0.272	0.6452
	n=5	0.5779	0.1538	0.6853	0.25	0.6254
	n=7	0.6119	0.1672	0.6921	0.2678	0.6478
NORMAL	n=3	0.8752	0.2599	0.1237	0.1641	0.5433
	n=5	0.8871	0.2833	0.0447	0.0751	0.5152
	n=7	0.8944	0.5	0.0309	0.0567	0.5129

K-EY algoritması 5 kat çapraz geçerleme sonucunda doğruluk değeri için normal veri seti için n=7, kesinlik değeri için normal veri seti ve n=7, duyarlılık değeri için RUS yöntemi ve n=7, f-ölçütü SMOTE yöntemi ve n=5 ve kappa değeri SMOTE yöntemi ve n=5 ile en yüksek oranlara sahip olmaktadır.

Özellikle sağlık alanında yapılan modellerin değerlendirilmesinde sıklıkla kullanılan AUC kriteri sonuçları Şekil 4.17’de gösterilmiştir.

**Şekil 4.19:** K-EY algoritması ROC eğrileri.



Şekil 4.20 (devam): K-EY algoritması ROC eğrileri.

4.2.5. LOJİSTİK REGRESYON

Lojistik Regresyon (LR) yöntemleriyle hangi değişkenin sınıflamaya ne kadar katkısının olduğunu anlayabilmek için `sag_alt_kadrana_agri_gocu`, `bulanti_kusma`, `istahsizlik`, `rebound`, `sag_alt_kadran_hassasiyet`, `ateş`, lökosit (WBC) ve NE% parametrelerinin katsayıları bulunarak modele verdikleri katkı çoktan aza doğru sıralanmıştır.

Dengeleme yöntemi uygulamadan niteliklerin katsayılarına bakıldığında rebound, lökosit ve ateş niteliklerinin, diğer niteliklere göre sınıflandırma üzerinde daha etkili olduğu sonucuna varılmıştır. Sağ alt kadrana ağrı göçü ve NE% niteliklerin sınıflandırma üzerinde pozitif etkisi bulunmakta, diğer parametreler ise negatif yönde etki etki oluşturmaktadır.

Yeni bir örnek sınıflandırmaya tanıtıldığında hangi sınıfa ait olduğunu belirlemek için aşağıdaki denklem belirleyici olacaktır.

$$\text{logit } \pi(x) = g(x) = -0.0140 -$$

$$0.0836\beta_{\text{sag_alt_kadrana_agri_gocu}} - 0.0849\beta_{\text{bulanti_kusma}} - 0.0474\beta_{\text{istahsizlik}} - 0.2210\beta_{\text{rebound}}$$

$$- 0.1096\beta_{\text{sag_alt_kadran_hassasiyet}} - 0.0966\beta_{\text{ates}} - 0.1702\beta_{\text{Lökosit}} + 0.0035\beta_{\text{NE\%}}$$

Yukarıdaki denkleme bakıldığında niteliklerin β katsayıları sınıflandırma üzerinde etki derecesine göre yukarıdan aşağı doğru sıralanmıştır:

1. Rebound
2. Lökosit (WBC)
3. Sağ alt kadranda hassasiyet
4. Ateş

5. Bulantı/Kusma
6. Sağ alt kadrana ağrı göçü
7. İştahsızlık
8. NE%

Niteliklerin üstünlükleri $Exp(\beta)$ 'lar: Odds ratio değerleri: [0.91978685, 0.91853805, 0.9536737, 0.80166754, 0.89613809, 0.90790402, 0.84347771, 1.0035376]

SMOTE algoritması uygulandıktan sonraki niteliklerin katsayılarına bakıldığında Rebound, ateş ve lökosit niteliklerinin, diğer niteliklere göre sınıflandırma üzerinde daha etkili olduğu sonucuna varılmıştır. Sağ alt kadrana ağrı göçü ve NE% niteliklerin sınıflandırma üzerinde pozitif etkisi bulunmakta, diğer parametreler ise negatif yönde etki etki oluşturmaktadır.

Yeni bir örnek sınıflandırmaya tanıtıldığında hangi sınıfa ait olduğunu belirlemek için aşağıdaki denklem belirleyici olacaktır.

$$\text{logit } \pi(x) = g(x) = 0.34168 +$$

$$0.0694\beta_{\text{sag_alt_kadrana_agri_gocu}} - 0.08394\beta_{\text{bulanti_kusma}} - 0.0905\beta_{\text{istahsizlik}} - 0.4393\beta_{\text{rebound}}$$

$$- 0.0047\beta_{\text{sag_alt_kadrana_hassasiyet}} - 0.3935\beta_{\text{ates}} - 0.1392\beta_{\text{Lökosit}} + 0.0216\beta_{\text{NE\%}}$$

Yukarıdaki denkleme bakıldığında niteliklerin β katsayıları sınıflandırma üzerinde etki derecesine göre yukarıdan aşağı doğru sıralanmıştır:

1. Rebound
2. Ateş
3. Lökosit (WBC)
4. İştahsızlık
5. Bulantı/Kusma
6. Sağ alt kadrana ağrı göçü
7. NE%
8. Sağ alt kadrana hassasiyet

Niteliklerin log üstünlüklerine bakarak etkilerini görmek yerine niteliklerin üstünlükleri $Exp(\beta)$ 'lar hesaplanarak ilişkiler rahat görülebilmektedir.

Odds ratio değerleri: [1.07191384, 0.91947993, 0.91345259, 0.64447121, 0.99525541, 0.67467932, 0.87000131, 1.02187071].

ROS algoritması uygulandıktan sonraki niteliklerin katsayılarına bakıldığında Rebound, ateş ve lökosit niteliklerinin, diğer niteliklere göre sınıflandırma üzerinde daha etkili olduğu sonucuna varılmıştır. Sağ alt kadrana ağrı göçü ve NE% niteliklerin sınıflandırma üzerinde pozitif etkisi bulunmakta, diğer parametreler ise negatif yönde etki etki oluşturmaktadır.

Yeni bir örnek sınıflandırmaya tanıtıldığında hangi sınıfa ait olduğunu belirlemek için aşağıdaki denklem belirleyici olacaktır.

$$\text{logit } \pi(x) = g(x) = 0.37987 +$$

$$0.0388\beta_{\text{sag_alt_kadrana_agri_gocu}} - 0.04032\beta_{\text{bulanti_kusma}} - 0.0789\beta_{\text{istahsizlik}} - 0.4154\beta_{\text{rebound}}$$

$$- 0.00551\beta_{\text{sag_alt_kadrana_hassasiyet}} - 0.3545\beta_{\text{ates}} - 0.1410\beta_{\text{Lökosit}} + 0.0208\beta_{\text{NE\%}}$$

Yukarıdaki denkleme bakıldığında niteliklerin β katsayıları sınıflandırma üzerinde etki derecesine göre yukarıdan aşağı doğru sıralanmıştır:

1. Rebound
2. Ateş
3. Lökosit (WBC)
4. İştahsızlık
5. Bulantı/Kusma
6. Sağ alt kadrana ağrı göçü
7. NE%
8. Sağ alt kadranda hassasiyet

niteliklerin üstünlükleri $\text{Exp}(\beta)$ 'lar: [1.03963363, 0.96047528, 0.92406994, 0.66007393, 1.0055311, 0.70146564, 0.86842001, 1.02108386]

RUS algoritması uygulandıktan sonraki katsayılarına bakıldığında lökosit, rebound ve ateş niteliklerinin, diğer niteliklere göre sınıflandırma üzerinde daha etkili olduğu sonucuna varılmıştır. Sağ alt kadrana ağrı göçü ve NE% niteliklerin sınıflandırma üzerinde pozitif etkisi bulunmakta, diğer parametreler ise negatif yönde etki etki oluşturmaktadır.

Yeni bir örnek sınıflandırmaya tanıtıldığında hangi sınıfa ait olduğunu belirlemek için aşağıdaki denklem belirleyici olacaktır.

$$\begin{aligned} \text{logit } \pi(x) = g(x) = & 0.0461 + \\ & 0.0259\beta_{\text{sag_alt_kadrana_agri_gocu}} - 0.00873\beta_{\text{bulanti_kusma}} + 0.00870\beta_{\text{istahsizlik}} - 0.09302 \\ & \beta_{\text{rebound}} + 0.01430\beta_{\text{sag_alt_kadran_hassasiyet}} - 0.0604\beta_{\text{ates}} - 0.1018\beta_{\text{Lökosit}} + 0.0152\beta_{\text{NE\%}} \end{aligned}$$

Yukarıdaki denkleme bakıldığında niteliklerin β katsayıları sınıflandırma üzerinde etki derecesine göre yukarıdan aşağı doğru sıralanmıştır:

1. Lökosit (WBC)
2. Rebound
3. Ateş
4. Sağ alt kadrana ağırlı göçü
5. NE%
6. Sağ alt kadranda hassasiyet
7. Bulantı/Kusma
8. İştahsızlık

Niteliklerin üstünlükleri $\text{Exp}(\beta)$ 'lar: 1.02628122, 0.99130381, 1.0087446, 0.91117273, 0.98579383, 0.9413337, 0.90320995, 1.01540658

Lojistik Regresyon yöntemini uygulamak için Python'da 'LogisticRegression' fonksiyonu kullanılmıştır.

classifier=LogisticRegression() fonksiyonu kullanılmıştır.

Logistik Regresyon algoritması veri seti için normal haliyle, SMOTE algoritması, Random Oversampling ve Random Undersampling algoritmaları ile uygulandı. Aşağıdaki tablolarda Karmaşıklık tabloları gösterilmektedir.

Tablo 4.27: LR normal karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	DN-318	YP-0
	Pozitif (AA-)	YN-46	DP-0

Verileri dengeleme yöntemleri kullanmadan normal Karmaşıklık tablosunu incelediğimizde DN=318, DP=0, YN=46 ve YP=0 değerlerini görmekteyiz. Doğru olarak sınıflanan hasta sayısı 318, gerçekte AA olmayan 46 hasta sınıflandırma sonucunda AA olarak etiketlendiğini görmekteyiz.

Tablo 4.28: LR algoritması SMOTE karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	217	101
	Pozitif (AA-)	20	26

Verilerden azınlık sınıfı yapay olarak çoğaltıp çoğunluk sınıfına yaklaştıran SMOTE algoritmasını kullandıktan sonra Karmaşıklık tablosunda DN=217, DP=26, YN=20 ve YP=101 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 243'dır. Gerçekte AA olan 101 hasta sınıflandırma sonucunda AA olmayan olarak ve gerçekte AA olan 20 hasta sınıflandırma sonucunda AA olmayan olarak etiketlenmiş olduğunu görmekteyiz.

Tablo 4.29: LR algoritması ROS karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	217	101
	Pozitif (AA-)	21	25

Verileri rasgele seçip azınlık sınıfından çoğunluk sınıfına yaklaştıran yöntemlerden biri olan Random Oversampling yöntemini kullandıktan sonra Karmaşıklık tablosunda DN=217, DP=25, YN=21 ve YP=101 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 242'dir. Gerçekte akut apandisit olan 101 hasta sınıflandırma sonucunda AA olmayan olarak ve gerçekte AA olan 21 hasta sınıflandırma sonucunda AA olan olarak etiketlenmiş olduğunu görmekteyiz.

Tablo 4.30: LR algoritması RUS karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	221	97
	Pozitif (AA-)	13	33

Verileri rasgele seçip çoğunluk sınıfından eksiltip azınlık sınıfına yaklaştıran yöntemlerden biri olan Random Undersampling algoritmasını kullandıktan sonra Karmaşıklık tablosunda DN=221, DP=33, YN=13 ve YP=97 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 254'tür. Gerçekte akut apandisit olan 97 hasta sınıflandırma sonucunda AA olmayan olarak ve gerçekte AA olan 13 hasta sınıflandırma sonucunda AA olan olarak etiketlenmiş olduğunu görmekteyiz.

Tablo 4.31: LR algoritması holdout metrikleri.

	Normal	SMOTE	ROS	RUS
Doğruluk	0.8770	0.6675	0.6648	0.6538
Kesinlik	0.0	0.2047	0.1984	0.1721
Duyarlılık	0.0	0.5652	0.5434	0.4565
F-Ölçütü	0.0	0.3005	0.2906	0.25
Kappa Değeri	0.0	0.1412	0.1295	0.0813

Doğruluk değeri Normal, SMOTE, ROS, RUS ve Alvarado için sırasıyla 0.87, 0.667, 0.664 ve 0.65'tir. En yüksek doğruluk değerleri dengesiz veri setine aittir.

Kesinlik değeri Normal, SMOTE, ROS, RUS ve Alvarado için sırasıyla 0.0, 0.56, 0.54 ve 0.45'dir. En yüksek kesinlik değeri SMOTE algoritmasına aittir.

Duyarlılık oranı dengesiz verilerde 0.0 iken SMOTE için 0.56, ROS için 0.54 ve RUS için 0.45 olarak hesaplanmıştır. En yüksek duyarlılık değeri SMOTE yöntemine aittir.

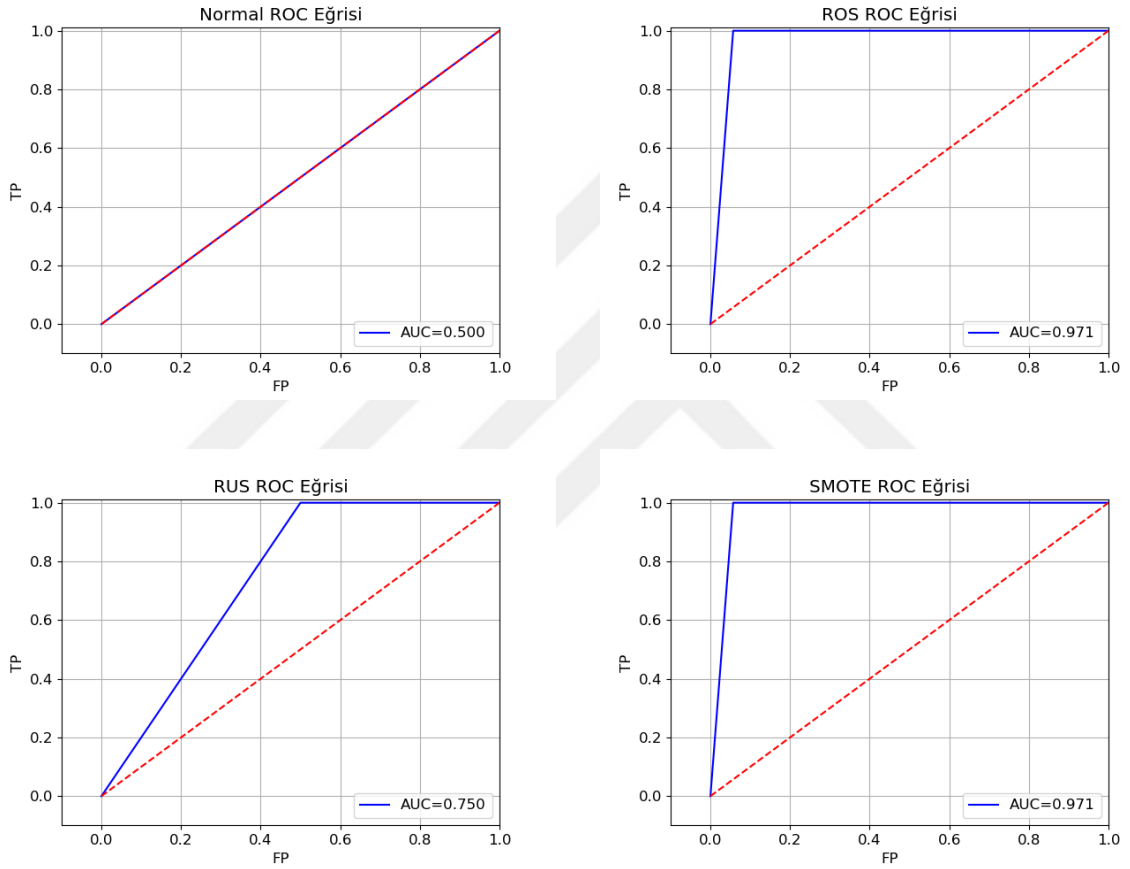
F-Ölçütü oranının Normal veri seti 0.0, SMOTE için 0.30, ROS için 0.29 ve RUS için 0.25 olduğunu görmekteyiz. En yüksek F-Ölçütü değeri SMOTE algoritmasına aittir.

Kappa değeri Normal veri seti için 0.0, SMOTE için 0.14, ROS için 0.12 ve RUS için 0.08 oranlarını görmekteyiz. Kappa değeri tüm yöntemler için düşük olarak yorumlanabilmektedir.

Tablo 4.32: LR algoritması 5 kat çapraz geçişleme sonuçları.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8954	0.6831	0.6733	0.67064
Kesinlik	0.0	0.2188	0.1964	0.19962
Duyarlılık	0.0	0.6631	0.6974	0.72282
F-Ölçütü	0.0	0.3275	0.3041	0.30902
AUC	0.5	0.6745	0.6843	0.69447

Doğruluk oranı Normal için 0.89, SMOTE için 0.68, ROS için 0.673 ve RUS için 0.67'dir. Kesinlik oranı Normal için 0.0, SMOTE için 0.21, ROS için 0.196 ve RUS için 0.199'dur. Duyarlılık oranı Normal için 0.0, SMOTE için 0.66, ROS için 0.69 ve RUS için 0.72'dir. F-Ölçütü Normal için 0.0, SMOTE için 0.32, ROS için 0.304 ve RUS için 0.309'dur. En Yüksek F-Ölçütü SMOTE yöntemine ve doğruluk oranı Normal veri setine aittir. AUC kriteri sonuçları Şekil 4.18'de gösterilmiştir.



Şekil 4.21: Lojistik regresyon ROC eğrileri.

4.2.6. YAPAY SİNİR AĞLARI

Bu çalışmada yapay sinir ağı çok katmanlı ileri beslemeli (multi layer feed forward) yapay sinir ağı mimarisi, ağı öğrenme algoritması olarak da tahmin problemlerinde en çok kullanılan Geri Yayımlı (Backpropagation) algoritması kullanılmıştır. Ağı yapısı 8 girdi, 2 çıktı verisi ve gizli katmanlardan oluşan çok katmanlı algılayıcı olarak tanımlanmıştır. 8 girdi seti; sag_alt_kadrana_agri_gocu, bulanti_kusma, istahsizlik, rebound, sag_alt_kadran_hassasiyet, aksiler_ates, wbc_binary, notrofilik_sola_kayma yer aldığı verilerdir. Çıktı seti ise akut

apandisit olması ve negatif akut apandisit verilerinden oluşmaktadır. Gizli katmandaki nöronların sayısı 8 ila 15 arasında olabilmektedir. Çıkış katmanında optimal sayıda nöron bulmak için deneme-yanılma yöntemini gizli katmandaki çeşitli sayılarla (8 ila 15 arasında değişen) nöronlar kullanılmıştır. Aktivasyon fonksiyonu olarak çıktı katmanı için Sigmoid ve gizli katmanlar için ReLu fonksiyonu kullanılmıştır. Maksimum yineleme sayısı deneme yapılarak 3000 olarak belirlenmiştir. YSA modellerinde diğer tüm parametreler Python yazılımında varsayılan değerler olarak ayarlanmıştır.

YSA algoritması yeniden örnekleme yapılmadan normal haliyle, SMOTE algoritması, Random Oversampling ve Random Undersampling algoritmaları ile uygulanmıştır. Aşağıda Karmaşıklık tabloları gösterilmektedir.

Tablo 4.33: YSA algoritması normal karmaşıklık tablosu.

		Tahmin Edilen	
		Hasta (AA+)	Sağlıklı (AA-)
Gerçek Sınıf	Hasta (AA+)	323	0
	Sağlıklı (AA-)	41	0

Verileri dengeleme yöntemleri kullanmadan normal karmaşıklık tablosunu incelediğimizde DN=323, DP=0, YN=41 ve YP=0 değerlerini görmekteyiz. Doğru olarak sınıflanan hasta sayısı 323, gerçekte akut apandisit olmayan 41 hasta sınıflandırma sonucunda akut apandisit olarak etiketlendiğini görmekteyiz.

Tablo 4.34: YSA algoritması SMOTE karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	DN-279	YP-44
	Pozitif (AA-)	YN-18	DP-23

Verilerden azınlık sınıfı yapay olarak çoğaltıp çoğunluk sınıfına yaklaştıran SMOTE yöntemini kullandıktan sonra Karmaşıklık tablosunda DN=279, DP=23, YN=18 ve YP=44 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 302'dir. 44 hastanın gerçekte AA olup sınıflandırma sonucunda AA olmayan olarak ve 18 hastanın gerçekte AA olmayıp sınıflandırma sonucu AA olarak etiketlendiğini görmekteyiz.

Tablo 4.35: YSA algoritması ROS karmaşıklık tablosu.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	DN-272	YP-52
	Pozitif (AA-)	YN-16	DP-24

Verileri rasgele seçip azınlık sınıfından çoğunluk sınıfına yaklaştıran yöntemlerden biri olan Random Oversampling algoritmasını kullandıktan sonra Karmaşıklık tablosunda DN=272, DP=24, YN=16 ve YP=52 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 296'dır. 52 hastanın gerçekte AA olup sınıflandırma sonucunda AA olmayan olarak ve 16 hastanın gerçekte AA olmayıp sınıflandırma sonucu AA olarak etiketlendiğini görmekteyiz.

Tablo 4.36: YSA algoritması RUS karmaşıklık tablosu.

		Tahmin Edilen	
		Pozitif (AA+)	Negatif (AA-)
Gerçek Sınıf	Pozitif (AA+)	DN-281	YP-43
	Negatif (AA-)	YN-27	DP-13

Verileri rasgele seçip çoğunluk sınıfından eksiltip azınlık sınıfına yaklaştıran yöntemlerden biri olan Random Undersampling yöntemini kullandıktan sonra karmaşıklık tablosunda DN=281, DP=13, YN=27 ve YP=43 olarak görülmektedir. Doğru olarak sınıflanan hasta sayısı 294'tür. Gerçekte AA olmayan 27 hasta sınıflandırma sonucunda AA olarak etiketlendiğini, gerçekte AA olan 43 hasta sınıflandırma sonucunda AA olmayan olarak etiketlenmiş olduğunu görmekteyiz.

Tablo 4.37: YSA algoritması holdout metrikleri.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.88736	0.82967	0.8131	0.80769
Kesinlik	0.0	0.34328	0.3157	0.23214
Duyarlılık	0.0	0.56097	0.6	0.325
F-Ölçütü	0.0	0.42592	0.4137	0.27083
Kappa Değeri	0.0	0.33266	0.3151	0.16360

Doğruluk değeri Normal, SMOTE, ROS ve RUS için sırasıyla 0.88, 0.82, 0.81 ve 0.80'dir. En yüksek doğruluk değerleri SMOTE algoritmasına aittir.

Kesinlik değeri Normal, SMOTE, ROS ve RUS için sırasıyla 0.0, 0.34, 0.31 ve 0.23'tür. En yüksek kesinlik değerleri SMOTE algoritmasına aittir.

Duyarlılık oranı Dengesiz verilerde 0.0 iken SMOTE için 0.56, ROS için 0.6 ve RUS için 0.32 olarak hesaplanmıştır. En yüksek duyarlılık değeri ROS algoritmasına aittir.

F-Ölçütü oranının Normal veri seti için 0.0, SMOTE için 0.42, ROS için 0.41 ve RUS için 0.27 olduğunu görmekteyiz. En yüksek F-Ölçütü değeri SMOTE algoritmasına aittir.

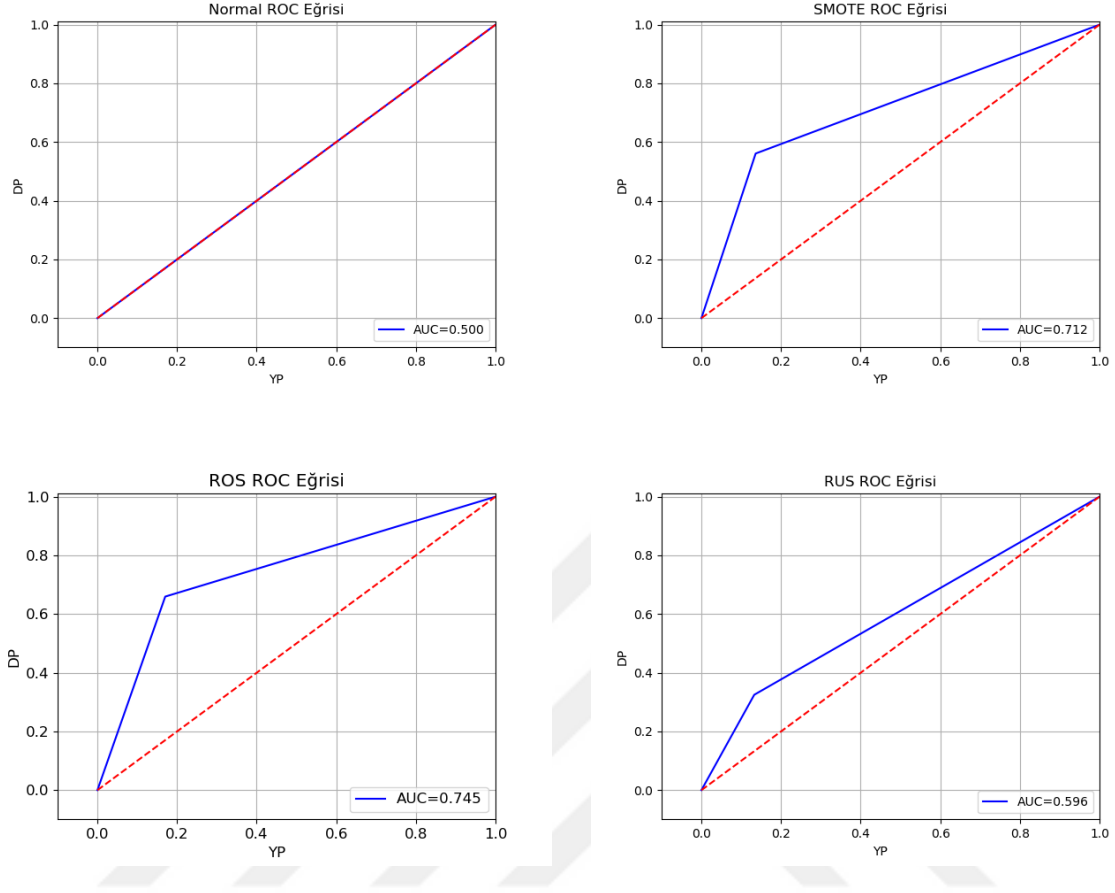
Kappa değeri Normal veri seti için 0.0, SMOTE için 0.33, ROS için 0.31 ve RUS algoritması için 0.16 oranlarını görmekteyiz.

Tablo 4.38: YSA algoritması 5 kat çapraz geçerleme sonuçları.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8917	0.7545	0.7994	0.8928
Kesinlik	0.0	0.2474	0.2876	0.3157
Duyarlılık	0.0	0.5714	0.5	0.1875
F-Ölçütü	0.0	0.3453	0.3652	0.2352
Kappa Değeri	0.0	0.2196	0.2562	0.1816

Doğruluk oranı Normal için 0.891, SMOTE için 0.75, ROS için 0.79 ve RUS için 0.89'dur. Kesinlik oranı Normal için 0.0, SMOTE için 0.24, ROS için 0.28 ve RUS için 0.31'dir. Duyarlılık oranı Normal için 0.0, SMOTE için 0.57, ROS için 0.5 ve RUS için 0.18'dir. F-Ölçütü Normal için 0.0, SMOTE için 0.34, ROS için 0.36 ve RUS için 0.23'dur. En Yüksek F-Ölçütü ROS yöntemine ve doğruluk oranı RUS yöntemine aittir.

Özellikle sağlık alanında yapılan modellerin değerlendirilmesinde sıklıkla kullanılan AUC kriteri sonuçları Şekil 4.19'da gösterilmiştir.



Şekil 4.22: YSA ROC eğrileri.

4.2.7.ALVARADO SKORU BAŞARI ÖLÇÜMÜ

Alvarado Skora göre ağrının sağ alt kadrana göçü 1, bulantı/kusma 1, iştahsızlık 1, sağ alt kadranda hassasiyet 2, ateş yüksekliği 1, Rebound 1, Lökositoz 2 ve NE% 1 olmak üzere toplam puan 10'dur. Bu katsayılar ile verilerin aldıkları değerler çarpılıp toplandığında toplam Alvarado skoru elde edilmiştir. Toplam skoru 7 ve üzeri olan hastalara akut apandisit teşhisi konup *cerrahi müdahale edilmeli* (0) , 7 altı olanlar ise *cerrahi müdahale edilmemeli* (1) olarak sınıflandırılmıştır. Alvarado skoru sonucunda elde edilen sonuçlar ile akut apandisit kesin tanısı için altın standart olarak kabul edilen histopatolojik değerlendirme sonuçları karşılaştırılıp 4.39 ve 4.40'da gösterilen sonuçlar elde edilmiştir.

Tablo 4.39: Alvarado karmaşıklık matrisi.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	118	42
	Pozitif (AA-)	553	741

Karmaşıklık tablosunu incelediğimizde DN=118, DP=741, YN=553 ve YP=42 değerlerini görmekteyiz. Doğru olarak sınıflanan hasta sayısı 859, gerçekte AA olmayan 553 hasta sınıflandırma sonucunda AA olarak ve gerçekte AA olan 42 hasta sınıflandırma sonucunda AA olmayan olarak etiketlendiğini görmekteyiz.

Tablo 4.40: Alvarado metrikleri.

ALVARADO	
Doğruluk	0.5907
Kesinlik	0.94
Duyarlılık	0.57
F-Ölçütü	0.70
Kappa Değeri	0.129

Alvarado Skorlama yönteminin performansını incelediğimizde 0.59 oranında doğruluk ve 0.70 F-Ölçütünü görmekteyiz. Makine öğrenmesi algoritmalarından en iyi performansı gösteren Destek Vektör Makinesi 1.0 doğruluk ve 1.0 F-Ölçütü ile Alvarado skorunun elde ettiği sonuçlardan daha başarılı olduğu görülmektedir.

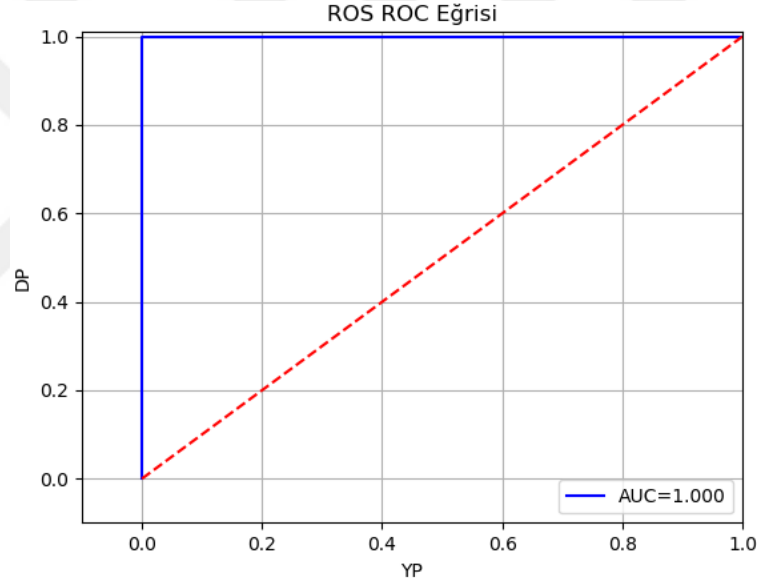
4.2.8. YAŞ VE CİNSİYET NİTELİKLİKLERİ EKLENEREK PERFORMANS BAŞARI ÖLÇÜMÜ

Alvarado Skorlama sisteminde olmayan cinsiyet ve yaş niteliklerinin değerleri verilere eklenerek sınıflandırma başarısını etkileyip etkilemediğinin analizi yapılmıştır. Doğruluk ve F-Ölçütüne bakıldığında yaş ve cinsiyet nitelikleri tüm sınıflandırma algoritmaları üzerinde performansı artırıcı bir etki oluşturduğu görülmektedir. DVM algoritmasında %100'e varan yüksek başarı elde ettiği görülmüştür. Yaş ve cinsiyet parametreleri akut apandisit teşhisinde tüm sınıflandırma algoritmaları için nitelik olarak eklenmesi tavsiye edilmektedir.

Tablo 4.41: DVM algoritması holdout yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8736	0.9890	1.0	0.9615
Kesinlik	0.0	0.9772	1.0	0.9444
Duyarlılık	0.0	0.9347	1.0	0.7391
F-Ölçütü	0.0	0.9555	1.0	0.8292
Kappa Değeri	0.0	0.9492	1.0	0.8079

Yaş ve cinsiyet nitelikler eklendikten sonra DVM algoritmasında parametre ayarlaması yapıp gamma değerini arttırdığımızda ROS yöntemi için doğruluk, kesinlik, duyarlılık, f-ölçütü ve kappa değeri %100 olarak tespit edilmiştir. En iyi performans göstermesi açısından DVM algoritması ve ROS yöntemi için Karar Destek Sistemi oluşturulmuştur.

**Şekil 4.23:** DVM yaş+cinsiyet nitelikleri eklendikten sonra ROC Eğrisi

DVM yaş+cinsiyet eklendikten sonraki ROC Eğrisine baktığımızda AUC değerini en iyi değer olan 1 olarak görmekteyiz.

Tablo 4.42: DVM yaş+cinsiyet nitelikleri eklendikten sonra karmaşıklık matrisi.

		Tahmin Edilen	
		Negatif (AA+)	Pozitif (AA-)
Gerçek Sınıf	Negatif (AA+)	318	0
	Pozitif (AA-)	0	46

DVM yaş+cinsiyet nitelikleri eklendikten sonraki Karmaşıklık tablosuna baktığımızda DN=318 ve DP=46 olmak üzere toplam doğru sınıflandırma sayısı 364 olup tüm sınıflandırma atamalarının %100 doğru yapıldığını görmekteyiz.

Tablo 4.43: DVM 5-fold yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8954	0.8128	0.9994	0.8328
Kesinlik	0.0	0.9054	0.9947	0.9192
Duyarlılık	0.0	0.7757	1.0	0.7834
F-Ölçütü	0.0	0.8159	0.9973	0.8329

Tablo 4.44: RO algoritması holdout yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8626	0.9587	0.9807	0.7335
Kesinlik	0.1666	0.8780	0.8679	0.3082
Duyarlılık	0.0217	0.7826	1.0	0.8913
F-Ölçütü	0.0384	0.8275	0.9292	0.4581
Kappa Değeri	0.0095	0.8042	0.9182	0.3328

Yaş ve cinsiyet nitelikler eklendikten sonra ROS yöntemi oldukça iyi sonuç vermiştir. Bu nitelikleri eklemekten önce doğruluk ve F-Ölçütü nispeten daha düşük oranda sonuç verdiğinden bu iki niteliği modele eklemeyi uygun görmekteyiz.

Tablo 4.45: RO algoritması 5-fold yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.

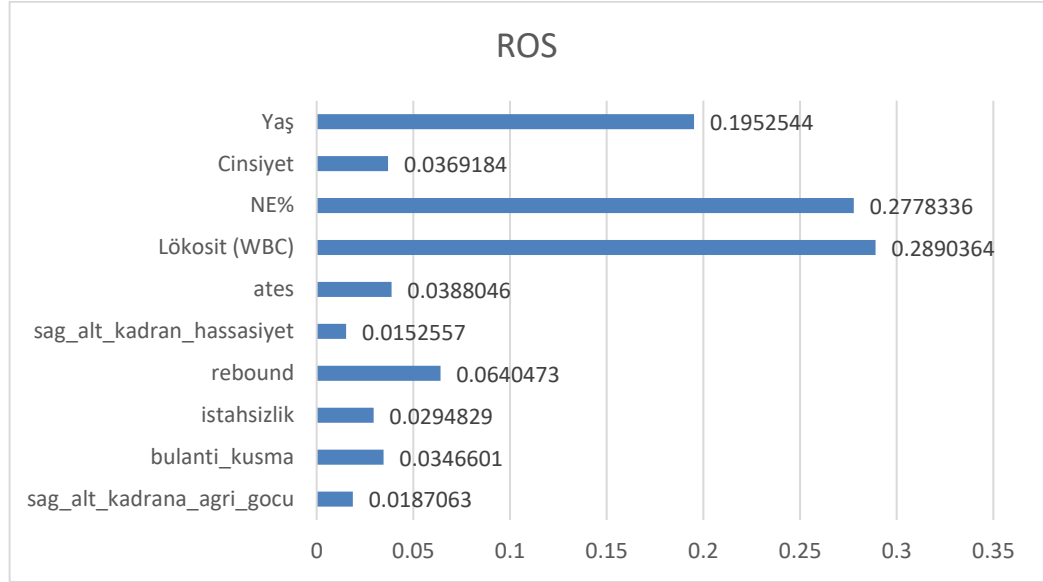
	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8926	0.9298	0.9595	0.5478
Kesinlik	0.3999	0.7436	0.8021	0.5238
Duyarlılık	0.0542	0.7264	1.0	0.3083
F-Ölçütü	0.0947	0.7346	0.8779	0.38195

RO algoritması 5-fold sonucunda ise ROS yöntemi doğruluk ve F-ölçütü için en iyi sonuç vermektedir.

Tablo 4.46: RO algoritması yaş+ cinsiyet nitelikleri eklenerek özellik önemi.

NİTELİK ÖNEMİ (FEATURE IMPORTANCE)				
	NORMAL	SMOTE	ROS	RUS
sag_alt_kadrana_agri_gocu	0.02569934	0.02223703	0.0187063	0.0312678
bulanti_kusma	0.02889119	0.08006385	0.0346601	0.02041033
İstahsızlık	0.02485439	0.07516372	0.0294829	0.03221523
Rebound	0.04658822	0.21882408	0.0640473	0.05772632
sag_alt_kadran_hassasiyet	0.0266775	0.04565695	0.0152557	0.01223601
Ateş	0.02327125	0.04425046	0.0388046	0.03825454
Lökosit (WBC)	0.29323361	0.1654375	0.2890364	0.28321206
NE%	0.27394121	0.12898945	0.2778336	0.27707265
Cinsiyet	0.03772602	0.10329329	0.0369184	0.0436664
Yaş	0.21911728	0.11608367	0.1952544	0.20393867

RO algoritması için en iyi performansı gösteren ROS yönteminin nitelik önemi grafiğine bakıldığında, Şekil 4.21’de RO ile yapılan sıralamaya göre Lökosit (WBC) değeri AA teşhisinde en önemli değişken (0.28), daha sonra NE% (0.27) ve yaş (0.19) en önemli 3 niteliktir. Rebound (0.06) ve cinsiyet (0.03) değişkenleri sınıflandırmaya normal katkı sağladığı ve en az katkıyı sağ alt kadranda hassasiyet (0.015) ve sağ alt kadrana ağrı göçü (0.018) değişkenleri sağlamaktadır.

**Şekil 4.24:**RO algoritması ROS yöntemi yaş+ cinsiyet nitelikleri eklenerek özellik önemi

Tablo 4.47: NB algoritması holdout yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8104	0.6758	0.7060	0.6895
Kesinlik	0.1891	0.2272	0.2201	0.2184
Duyarlılık	0.1521	0.6521	0.5217	0.5652
F-Ölçütü	0.1686	0.3370	0.3096	0.3151
Kappa Değeri	0.0631	0.1841	0.1604	0.1624

Tablo 4.48: NB algoritması 5 kat çapraz geçerleme yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8394	0.8394	0.7238	0.7247
Kesinlik	0.2718	0.2718	0.2273	0.2296
Duyarlılık	0.3177	0.3177	0.6961	0.7065
F-Ölçütü	0.2902	0.2902	0.3412	0.3453
Kappa Değeri	0.6091	0.6091	0.7116	0.71685

Tablo 4.49: K-EY algoritması holdout yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8571	0.7417	0.7554	0.5686
Kesinlik	0.25	0.3208	0.3358	0.1828
Duyarlılık	0.0652	0.9347	0.9565	0.6956
F-Ölçütü	0.1034	0.4777	0.4971	0.2895
Kappa Değeri	0.0539	0.3567	0.3814	0.1118

Tablo 4.50: K-EY algoritması 5-fold yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8862	0.6596	0.8862	0.5990
Kesinlik	0.2666	0.1507	0.2666	0.1472
Duyarlılık	0.0470	0.4919	0.0470	0.6031
F-Ölçütü	0.0784	0.2288	0.0784	0.2358
Kappa Değeri	0.5153	0.5862	0.5153	0.6010

Tablo 4.51: YSA algoritması holdout yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.9120	0.7967	0.8021	0.7939
Kesinlik	0.0	0.3	0.2361	0.2537
Duyarlılık	0.0	0.5714	0.5	0.4047
F-Ölçütü	0.0	0.3934	0.3207	0.3119
Kappa Değeri	0.0	0.2852	0.2220	0.1981

Tablo 4.52: LR algoritması holdout yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8736	0.6428	0.6565	0.6483
Kesinlik	0.0	0.1718	0.184	0.1796
Duyarlılık	0.0	0.4782	0.5	0.5
F-Ölçütü	0.0	0.2528	0.2690	0.2643
Kappa Değeri	0.0	0.0822	0.1033	0.0963

Tablo 4.53: LR algoritması 5-fold yaş+cinsiyet niteliklerinin eklenmesi sonucunda metrikler.

	NORMAL	SMOTE	ROS	RUS
Doğruluk	0.8954	0.6889	0.7036	0.6926
Kesinlik	0.0	0.1963	0.2195	0.2053
Duyarlılık	0.0	0.6446	0.7232	0.6870
F-Ölçütü	0.0	0.2999	0.3349	0.3144

Tablo 4.54: LR algoritması yaş+ cinsiyet eklendikten sonra katsayılar.

LOJISTİK REGRESYON KATSAYILARI				
	NORMAL	SMOTE	ROS	RUS
sag_alt_kadrana_agri_gocu	-0.06541271	0.00354498	0.07138812	0.02642333
bulanti_kusma	-0.08998713	-0.06633286	-0.1011655	-0.0111984
İstahsızlık	-0.03552248	-0.01354433	-0.0418338	0.0098373
Rebound	-0.19904541	-0.39454392	-0.4342898	-0.0851488
sag_alt_kadran_hassasiyet	-0.08535847	0.01772959	-0.0278374	-0.0129133
Ateş	-0.0996052	-0.36017704	-0.3750077	-0.0550096
Lökosit (WBC)	-0.16755368	-0.13505249	-0.1564029	-0.1014216
NE %	0.01097776	0.02273536	0.02604788	0.0193874
Cinsiyet	0.10299965	0.34993363	0.30695222	0.0641339
Yaş	-0.02182365	-0.00970441	-0.0094619	-0.0105298
Intercept	0.010836530	0.331912085	0.36348349	0.05004888

Yaş ve cinsiyet nitelikleri eklendikten sonra katsayılara baktığımızda Rebound en yüksek oranda sınıflandırma üzerinde tersi yönde etki etmektedir. Bunu Lökositoz takip etmektedir. Cinsiyet parametresi sınıflandırma üzerinde etkisi olduğu görülürken yaş parametresinin daha az katkıda bulunduğunu görmekteyiz.

ROS yöntemi için yeni bir örnek sınıflandırmaya tanıtıldığında hangi sınıfa ait olduğunu belirlemek için aşağıdaki denklem belirleyici olacaktır.

$$\begin{aligned} \text{logit } \pi(x) = g(x) = & 0.363483 + 0.0713\beta_{\text{sag_alt_kadrana_agri_gocu}} - 0.1011\beta_{\text{bulanti_kusma}} - \\ & 0.0418\beta_{\text{istahsızlık}} - 0.4342\beta_{\text{rebound}} - 0.0278\beta_{\text{sag_alt_kadran_hassasiyet}} - 0.3750\beta_{\text{ates}} - \\ & 0.1564\beta_{\text{Lökosit}} + 0.0260\beta_{\text{NE\%}} + 0.3069\beta_{\text{Cinsiyet}} - 0.0094\beta_{\text{Yaş}} \end{aligned}$$

Yukarıdaki denkleme bakıldığında niteliklerin β katsayıları sınıflandırma üzerinde etki derecesine göre yukarıdan aşağı doğru sıralanmıştır:

1. Rebound
2. Ateş
3. Cinsiyet
4. Lökosit (WBC)
5. Bulantı/Kusma
6. Sağ alt kadrana ağrı göçü
7. İştahsızlık
8. Sağ alt kadranda hassasiyet
9. NE%
10. Yaş

4.3. SINIFLANDIRMA MODELLERİNİN PERFORMANSLARININ KARŞILAŞTIRILMASI

Sınıflandırma algoritmalarından elde edilen sonuçlar Tablo 4.55 ve Tablo 4.56’da gösterilmektedir. En iyi performansı gösteren algoritmaların sonuçları daha belirgin hale getirilmiştir.

Tablo 4.55: Analizlerin genel performans değerlendirmesi (holdout).

SINIFLANDIRMA YÖNTEMLERİ	METRİKLER	ALGORİTMALAR			
		Normal Veri Seti	SMOTE	ROS	RUS
RASGELE ORMAN (RANDOM FOREST-RF)	Doğruluk	0.86538	0.93406	0.97252	0.74175
	Kesinlik	0.28571	0.76190	0.82142	0.31818
	Duyarlılık	0.04347	0.69565	1.0	0.91304
	F-Ölçütü	0.07547	0.72727	0.90196	0.47191

Tablo 4.55 (devam): Analizlerin genel performans deęerlendirmesi (holdout).

SINIFLANDIRMA YÖNTEMLERİ	METRİKLER	Normal Veri Seti	SMOTE	ROS	RUS
DESTEK VEKTÖR MAKİNESİ (DVM)	Doęruluk	0.8736	0.9313	0.9560	0.7527
	Kesinlik	0.5	0.6521	0.75	0.3253
	Duyarlılık	0.0217	0.9782	0.9782	0.8913
	F-Ölçütü	0.0416	0.7826	0.8490	0.4767
NAIVE BAYES ALGORİTMASI	Doęruluk	0.8076	0.6153	0.6840	0.6785
	Kesinlik	0.1666	0.2062	0.2063	0.2113
	Duyarlılık	0.1304	0.7173	0.5652	0.5652
	F-Ölçütü	0.1463	0.3203	0.2857	0.3076
K- EN YAKIN KOMŞU) ALGORİTMASI	Doęruluk	0.85714	0.75549	0.79670	0.62362
	Kesinlik	0.2	0.33587	0.375	0.21383
	Duyarlılık	0.0434	0.95652	0.9130	0.73913
	F-Ölçütü	0.0714	0.49717	0.53164	0.33170
LOJİSTİK REGRESYON	Doęruluk	0.8770	0.6675	0.6648	0.6538
	Kesinlik	0.0	0.2047	0.1984	0.1721
	Duyarlılık	0.0	0.5652	0.5434	0.4565
	F-Ölçütü	0.0	0.3005	0.2906	0.25
YAPAY SİNİR AĞLARI (YSA)	Doęruluk	0.88736	0.82967	0.8131	0.80769
	Kesinlik	0.0	0.34328	0.3157	0.23214
	Duyarlılık	0.0	0.56097	0.6	0.325
	F-Ölçütü	0.0	0.42592	0.4137	0.27083

Tablo 4.56: Tabakalı 5-kat çapraz geçirme ile analizlerin performans değerlendirmesi.

SINIFLANDIRMA YÖNTEMLERİ	METRİKLER	ALGORİTMALAR			
		Normal Veri Seti	SMOTE	ROS	RUS
RASGELE ORMAN (RANDOM FOREST- RF)	Doğruluk	0.87889	0.90731	0.94318	0.5564
	Kesinlik	0.2238	0.7205	0.75748	0.5222
	Duyarlılık	0.06606	0.78419	0.99534	0.2872
	F-Ölçütü	0.10048	0.74507	0.8401	0.3673
DESTEK VEKTÖR MAKİNESİ (DVM)	Doğruluk	0.8935	0.8919	0.9651	0.8806
	Kesinlik	0.1	0.7541	0.859	0.7374
	Duyarlılık	0.0074	0.9314	0.9856	0.9295
	F-Ölçütü	0.0137	0.8086	0.9115	0.7946
NAIVE BAYES ALGORİTMASI	Doğruluk	0.844	0.6477	0.7045	0.6926
	Kesinlik	0.2724	0.1894	0.2149	0.2168
	Duyarlılık	0.2946	0.7313	0.6762	0.5711
	F-Ölçütü	0.2789	0.2992	0.3225	0.2603
K- EN YAKIN KOMŞU) ALGORİTMASI	Doğruluk	0.8871	0.7165	0.7467	0.5779
	Kesinlik	0.2833	0.1955	0.1981	0.1538
	Duyarlılık	0.0447	0.5623	0.4712	0.6853
	F-Ölçütü	0.0751	0.288	0.2753	0.25
LOJİSTİK REGRESYON	Doğruluk	0.8954	0.6831	0.6733	0.67064
	Kesinlik	0	0.2188	0.1964	0.19962
	Duyarlılık	0	0.6631	0.6974	0.72282
	F-Ölçütü	0	0.3275	0.3041	0.30902
YAPAY SİNİR AĞLARI	Doğruluk	0.8917	0.7545	0.7994	0.8928
	Kesinlik	0	0.2474	0.2876	0.3157
	Duyarlılık	0	0.5714	0.5	0.1875
	F-Ölçütü	0	0.3453	0.3652	0.2352

Sınıflandırma sonuçlarından elde edilen bulgulara göre hastalık teşhisinde en iyi performansı gösteren algoritma Destek Vektör Makinesi Algoritmasıdır. Daha sonra Rasgele Orman, K-En Yakın Komşu ve Yapay Sinir Ağları izlemektedir. Performans değerlendirilirken F-ölçütü ve doğruluk oranı göz önünde bulundurulmuştur.

Kullanılan makine öğrenimi yöntemleriyle hangi niteliğin sınıflamaya ne kadar katkısının olduğu da anlaşılabilmektedir. Tablo 4.57 ve 4.58’de analizlerde kullanılan niteliklerin modele verdikleri katkılara göre sıralaması yer almaktadır. Logistik Regresyon analizi ile β katsayılarına bakılmış olup en çok katkı sağlayan niteliklerin sıralanması yapılmıştır. Normal,

SMOTE, ROS ve RUS algoritmaları incelendiğinde ilk sırayı Rebound almaktadır. Ateş ve WBC bunu takip etmektedir.

Bu çalışma için performansı en yüksek modellemelerle (RO 'da nestimator=100 ve Lojistik Regresyon'da radial) değişken önemliliği analizi yapılmıştır. RO yönteminde WBC (lökosit), NE% ve Rebound en önemli üç nitelik olarak karşımıza çıkmaktadır. SMOTE ve ROS yöntemlerinde sağ alt kadrana ağrı göçü sınıflamaya çok az katkı sağladığı görülmektedir.

Tablo 4.57: RO ile niteliklerin modele verdikleri katkılara göre sıralanması.

	RASGELE ORMAN			
	NORMAL	SMOTE	ROS	RUS
1	Lökosit(WBC)	Lökosit(WBC)	Lökosit(WBC)	Lökosit(WBC)
2	NE%	NE%	NE%	NE%
3	Rebound	Rebound	Rebound	Rebound
4	İştahsızlık	Sağ alt kardandan hassasiyet	Ateş	Bulantı/kusma
5	Sağ alt kardandan hassasiyet	Ateş	Bulantı/kusma	Sağ alt kadrana ağrı göçü
6	Bulantı/kusma	Bulantı/kusma	İştahsızlık	Ateş
7	Sağ alt kadrana ağrı göçü	İştahsızlık	Sağ alt kadrana ağrı göçü	İştahsızlık
8	Ateş	Sağ alt kadrana ağrı göçü	Sağ alt kardandan hassasiyet	Sağ alt kardandan hassasiyet

Tablo 4.58: Lojistik regresyon ile niteliklerin modele verdikleri katkılara göre sıralanması.

	LOJİSTİK REGRESYON			
	NORMAL	SMOTE	ROS	RUS
1	Rebound	Rebound	Rebound	Lökosit (WBC)
2	Lökosit (WBC)	Ateş	Ateş	Rebound
3	Sağ alt kadranda hassasiyet	Lökosit (WBC)	Lökosit (WBC)	Ateş
4	Ateş	İştahsızlık	İştahsızlık	Sağ alt kadrana ağrı göçü
5	Bulantı/Kusma	Bulantı/Kusma	Bulantı/Kusma	NE%
6	Sağ alt kadrana ağrı göçü	Sağ alt kadrana ağrı göçü	Sağ alt kadrana ağrı göçü	Sağ alt kadranda hassasiyet
7	İştahsızlık	NE%	NE%	Bulantı/Kusma
8	NE%	Sağ alt kadranda hassasiyet	Sağ alt kadranda hassasiyet	İştahsızlık

Tablo 4.59: Yaş+cinsiyet niteliklerinin eklenmesinden sonraki genel analizler (holdout).

SINIFLANDIRMA YÖNTEMLERİ	METRİKLER	ALGORİTMALAR			
		Normal Veri Seti	SMOTE	ROS	RUS
RASGELE ORMAN (RANDOM FOREST- RF)	Doğruluk	0.8626	0.9587	0.9807	0.7335
	Kesinlik	0.1666	0.878	0.8679	0.3082
	Duyarlılık	0.0217	0.7826	1	0.8913
	F-Ölçütü	0.0384	0.8275	0.9292	0.4581
DESTEK VEKTÖR MAKİNESİ (DVM)	Doğruluk	0.8736	0.989	1	0.9615
	Kesinlik	0	0.9772	1	0.9444
	Duyarlılık	0	0.9347	1	0.7391
	F-Ölçütü	0	0.9555	1	0.8292
NAIVE BAYES ALGORİTMASI	Doğruluk	0.8104	0.6758	0.706	0.6895
	Kesinlik	0.1891	0.2272	0.2201	0.2184
	Duyarlılık	0.1521	0.6521	0.5217	0.5652
	F-Ölçütü	0.1686	0.337	0.3096	0.3151
K- EN YAKIN KOMŞU) ALGORİTMASI	Doğruluk	0.8571	0.7417	0.7554	0.5686
	Kesinlik	0.25	0.3208	0.3358	0.1828
	Duyarlılık	0.0652	0.9347	0.9565	0.6956
	F-Ölçütü	0.1034	0.4777	0.4971	0.2895
LOJİSTİK REGRESYON	Doğruluk	0.8736	0.6428	0.6565	0.6483
	Kesinlik	0	0.1718	0.184	0.1796
	Duyarlılık	0	0.4782	0.5	0.5
	F-Ölçütü	0	0.2528	0.269	0.2643
YAPAY SİNİR AĞLARI	Doğruluk	0.912	0.7967	0.8021	0.7939
	Kesinlik	0	0.3	0.2361	0.2537
	Duyarlılık	0	0.5714	0.5	0.4047
	F-Ölçütü	0	0.3934	0.3207	0.3119

Doğruluk ve F- Ölçütüne bakıldığında yaş ve cinsiyet nitelikleri tüm sınıflandırma algoritmaları üzerinde performansı arttırıcı bir etki oluşturduğu görülmektedir. DVM algoritmasında %100'e varan yüksek başarı elde ettiği görülmüştür. Yaş ve cinsiyet parametreleri akut apandisit teşhisinde tüm sınıflandırma algoritmaları için nitelik olarak eklenmesi tavsiye edilmektedir.

Tablo 4.60: Yaş+cinsiyet nitelikleri ile Tabakalı 5-kat çapraz geçerleme genel analizler.

SINIFLANDIRMA YÖNTEMLERİ	METRİKLER	ALGORİTMALAR			
		Normal Veri Seti	SMOTE	ROS	RUS
RASGELE ORMAN (RANDOM FOREST- RF)	Doğruluk	0.8926	0.9298	0.9595	0.5478
	Kesinlik	0.3999	0.7436	0.8021	0.5238
	Duyarlılık	0.0542	0.7264	1	0.3083
	F-Ölçütü	0.0947	0.7346	0.8779	0.38195
DESTEK VEKTÖR MAKİNESİ (DVM)	Doğruluk	0.8954	0.8128	0.9994	0.8328
	Kesinlik	0	0.9054	0.9947	0.9192
	Duyarlılık	0	0.7757	1	0.7834
	F-Ölçütü	0	0.8159	0.9973	0.8329
NAIVE BAYES ALGORİTMASI	Doğruluk	0.8104	0.6758	0.706	0.6895
	Kesinlik	0.1891	0.2272	0.2201	0.2184
	Duyarlılık	0.1521	0.6521	0.5217	0.5652
	F-Ölçütü	0.1686	0.337	0.3096	0.3151
K- EN YAKIN KOMŞU) ALGORİTMASI	Doğruluk	0.8571	0.7417	0.7554	0.5686
	Kesinlik	0.25	0.3208	0.3358	0.1828
	Duyarlılık	0.0652	0.9347	0.9565	0.6956
	F-Ölçütü	0.1034	0.4777	0.4971	0.2895
LOJİSTİK REGRESYON	Doğruluk	0.8736	0.6428	0.6565	0.6483
	Kesinlik	0	0.1718	0.184	0.1796
	Duyarlılık	0	0.4782	0.5	0.5
	F-Ölçütü	0	0.2528	0.269	0.2643
YAPAY SİNİR AĞLARI	Doğruluk	0.912	0.7967	0.8021	0.7939
	Kesinlik	0	0.3	0.2361	0.2537
	Duyarlılık	0	0.5714	0.5	0.4047
	F-Ölçütü	0	0.3934	0.3207	0.3119

Tablo 4.61: Lojistik regresyon ile niteliklerin modele verdikleri katkılara göre sıralanması.

	LOJİSTİK REGRESYON			
	NORMAL	SMOTE	ROS	RUS
1	Rebound	Rebound	Rebound	Lökosit
2	Lökosit(WBC)	Ateş	Ateş	Rebound
3	Cinsiyet	Cinsiyet	Cinsiyet	Cinsiyet
4	Ateş	Lökosit (WBC)	Lökosit	Ateş
5	Bulantı/Kusma	Bulantı/Kusma	Bulantı/Kusma	Sağ Alt Kadrana Ağrı Göçü
6	Sağ Alt Kadranda Hassasiyet	NE%	Sağ Alt Kadrana Ağrı Göçü	NE%
7	Sağ Alt Kadrana Ağrı Göçü	Sağ Alt Kadranda Hassasiyet	İştahsızlık	Sağ Alt Kadranda Hassasiyet
8	İştahsızlık	İştahsızlık	Sağ Alt Kadranda Hassasiyet	Bulantı/Kusma
9	Yaş	Yaş	NE%	Yaş
10	NE%	Sağ Alt Kadrana Ağrı Göçü	Yaş	İştahsızlık

4.4. KARAR DESTEK SİSTEMİ UYGULAMASI

Klinik Karar Destek Sistemi (KKDS) klinisyenlere klinik bilgileri ve hastayla ilgili bilgileri sağlamak, akılcıca filtrelenmek ve hasta bakımını arttırmak için uygun zamanlarda sunulmasıdır (LaRosa, J. ve diğ., 2012).

KDDS sistemleri klinik bakımda yaygın olarak kullanılır, ancak düşük kullanılabilirlik nedeniyle sıklıkla etkisizdir (Genes, N., ve diğ., 2016). İnsan faktörleri, kullanılabilirlik ve insan bilgisayar arayüzü ilkeleri KDDS'nin başarısı için kritik öneme sahiptir. KDDS'yi insan faktörleri açısından değerlendirerek, kullanıcı düşünülerek tasarlanmış araçlar oluşturmak için hayal kırıklıkları ve hataya açık koşullar en aza indirilebilir. Özellikle grafiksel kullanıcı arayüzleri tasarlamaya odaklanan kurallar gibi sistem tasarım kılavuzları, klinik kullanılabilirliğe odaklanan özel kurallarla ilgili daha iyi tasarım ve işlevsel özellikler gerektirir. Bu etkileşimli bilgisayar sistemlerinin tasarımını insan faktörleri mühendisliği perspektifinden değerlendirmek, klinisyenler ve sistem arasındaki etkileşimi optimize edebilir. Sonunda hem hasta hem de klinisyen sonuçlarını iyileştirebilir.

Sağlık alanında Yapay Sinir Ağları, Bayes Ağları, Bulanık Mantık, Kural Tabanlı Yaklaşım algoritmaları teşhis desteği veren sistemler geliştirilirken en çok tercih edilen algoritmalarlardır. (Khoong, C.,M., 1995).

Sınıflandırma sonucunda en iyi performans gösteren Destek Vektör Makinesi algoritması alt yapıda çalışacak şekilde Python GUI arayüzüne entegre edilen bir Klinik Karar Destek Sistemi oluşturulmuştur. Özellikle hekimlerin kullanımına uygun olan bu uygulamada hekim nitelikleri girdiği zaman ona karar vermesinde yardımcı olan *Akut Apandisit-Müdahale edilmeli* veya *Gözlemlenmeli* uyarısı vermektedir. Şekil 4.25 ve 4.26'da Karar Destek Sisteminin arayüzü gösterilmektedir.

HASTA VERİ GİRİŞİ-Lütfen aşağıdaki alanları doldurunuz.	
Sağ Alt Kadrana Ağrı Göçü	1
Bulanti/Kusma	0
İştahsızlık	0
Rebound Ağrı	0
Sağ Alt Kadranda Hassasiyet	1
Ateş	0
Lökosit (WBC)	14.15
NE%	76.2
Cinsiyet	0
Yaş	44
SONUÇ:	Akut Apandisit-Müdahale Edilmeli
Tahmin Et	

Şekil 4.25: KDS- AA olan hasta örneği.

Şekil 4.25'te hastanın öyküsü, fiziki muayenesi ve kan biyokimya parametreleri ile akut apandisit teşhisinde, hekime yardımcı olacak ve kolaylık sağlayabilecek bir karar destek sistemi oluşturulmuştur. Sistemin işleyişi veri madenciliği tekniklerinden Destek Vektör Makinesi yöntemi ile sağlanmaktadır. Sisteme giriş olarak, hastanın öykü ve fiziksel muayenesi parametreleri olan sağ alt kadrana ağrı göçü, sağ alt kadranda hassasiyet, rebound ağrı,

iştahsızlık, bulantı/kusma, ateş, biyokimya parametrelerinden akut apandisit hastalığı için temel belirleyiciler olan WBC Lökositve Netrofil Yüzdesi (NE%) değerlendirmelerde bulunulmuştur. Hastananın öykü ve fiziksel muayenesi 0 (yok) ve 1 (var) olarak girilmelidir. WBC ve NE% değerleri sayısal değerler olarak girilmelidir.

HASTA VERİ GİRİŞİ-Lütfen aşağıdaki alanları doldurunuz.	
Sağ Alt Kadrana Ağrı Göçü	1
Bulantı/Kusma	0
İştahsızlık	1
Rebound Ağrı	0
Sağ Alt Kadranda Hassasiyet	1
Ateş	0
Lökosit (WBC)	8.76
NE%	58.6
Cinsiyet	0
Yaş	50
SONUÇ:	Gözlemlenmeli

Tahmin Et

Şekil 4.26: KDS- AA olmayan hasta örneği.

Şekil 4.26'da girilen değerlerin Tahmin et butonuna basıldıktan sonra *Gözlemlenmeli* sonucu görülmektedir.

5. TARTIŞMA VE SONUÇ

Sağlık alanında karın ağrısı şikâyetiyle hastaneye başvuran hastalarda akut apandisit (AA) en sık görülen acil durumlardan biridir. Ülkemiz çapında sık rastlanılan acil durumlardan biri olan akut apandisit klinik tanı koymasını hekim açısından kolaylaştırmak ve gereksiz ameliyatların önüne geçerek sağlık hizmetlerinde israfı azaltmak ve insan sağlığını gereksiz yere tehlikeye atılmasını önlemek amacıyla uzman hekimlerle görüşülmüş ve onların önerileri doğrultusunda bu çalışma ortaya çıkmıştır. Akut apandisit tanısını koymak oldukça güçtür. Akut batın, akut karın ağrısı, bağırsak tıkanıkları, gastroenterit vb ile akut apandisit belirtileri çok fazla karıştırıldığı için bu çalışma genel cerrahide hekimlerin karar vermesinde kolaylık sağlayacaktır.

Son yıllarda, hastaların ameliyat edilip edilmeme kararı hala %10 ile %25 arasında negatif apendektomi oranları nedeniyle onlarca gereksiz operasyona ve hastaların acı çekmesine neden olan bir zorluk oluşturmaktadır (Hale ve diğ.,1997; Colson ve diğ., 1997). Negatif apendektomi oranının % 20 ila % 30'lara ulaşması ne denli zor bir karar olduğunu kanıtlamaktadır (Yang ve diğ., 2006). Literatürde AA için negatif laparotomi (ameliyat) oranının % 4 ila %75 arasında değiştiği bildirilmiştir (Eskelinen ve diğ., 1992).

Sağlık alanında sıkça görülen sınıf dengesizliği problemini azaltmak için bu çalışmada SMOTE, Random Oversampling ve Random Undersampling algoritmaları önerilmiştir. Veri setlerindeki dengesizlik problemi azaltılıp veriler dengeli hale getirildikten sonra AA teşhisinin tahmini için Alvarado Skora ait risk faktörlerinden yararlanılmıştır. Bu amaçla Alvarado skoru hesaplanarak akut apandisit tanısında altın standart olarak kabul edilen histopatolojik rapor sonucu ile karşılaştırılmıştır.

Bu çalışma kapsamında, sınıflandırma algoritmalarından Lojistik Regresyon, Rasgele Orman, Yapay Sinir Ağları, Naive Bayes, k-En Yakın Komşu Algoritması ve Karar Ağacı kullanılarak en iyi performansı göstermeyi amaçlayan analizler yapılmıştır. Bu modellerin sonuçları F-Ölçütü ve doğruluk oranları göz önünde bulundurularak karşılaştırılmış ve en iyi performansı gösteren Destek Vektör Makinesi ve Rasgele Orman Algoritması kullanılması tavsiye edilmiştir. Verileri dengelemeyi sağlayan yöntemleri ve verilerin analizleri için Python kodlama dili kullanılmıştır.

Akut apandisit teşhisine yönelik makine öğrenmesi sınıflandırma algoritmaları aracılığı ile gerçekleştirilen bu çalışmada en iyi performansı gösterdiği tespit edilen Destek Vektör Makinesi algoritması için arayüz oluşturularak hekimlerin karar verme mekanizmasına katkı sunabilecek Karar Destek Sistemi uygulaması geliştirilmiştir.

Bu çalışmanın katkıları aşağıda sıralanmıştır:

- Akut Apandisit teşhisi için kullanılan Alvarado Skor ile Makine Öğrenmesi Algoritmaları karşılaştırılmış ve karın ağrısıyla gelen hastaların akut apandisit olup olmadığına karar vermede yardımcı olan en iyi sınıflandırmayı yapan algoritmanın Destek Vektör Makinesi olduğu tespit edilmiştir. Parametre ayarlaması yapılarak ve Alvarado Skorundaki parametrelere ek olarak yaş ve cinsiyet parametreleri de eklenerek 1454 hasta verisi ile %100 doğruluk bulunmuştur.
- Dengesiz veri setlerinde en çok kullanılan yöntemlerden Random undersampling (RUS), Random oversampling (ROS) ve SMOTE algoritmaları kullanılarak veri seti dengeli hale getirilmiştir.
- Bu çalışmada en iyi performansın tespit edildiği Destek Vektör Makinesi algoritması için arayüz oluşturularak hekimlerin karar verme mekanizmasına katkı sunabilecek Karar Destek Sistemi uygulaması geliştirilmiştir.
- Yaş ve cinsiyet parametreleri de eklenerek bu iki parametrenin akut apandisit teşhisi için önemli olup olmadığının tespiti yapılmıştır.
- Niteliklerin katsayılarına ve niteliklerin önemine bakılarak modele verdikleri katkı sıralanmıştır.

Çalışmamızda Destek Vektör Makineleri; Rastgele Orman, Yapay Sinir Ağları, Lojistik Regresyon ve Alvarado skorlama sisteminden anlamlı olarak daha doğru sonuçlar vererek daha iyi performans göstermiştir. Yapay Sinir Ağları, Lojistik Regresyon ve Alvarado arasında anlamlı fark görülmemiştir. Rastgele Orman modeli Alvarado skorundaki nitelikler ile tahmin yaptığında Destek Vektör makinelerine göre daha iyi sonuç vermektedir. Rastgele Ormanda, yalnızca bir anahtar parametre (ağaç sayısı) ayarlanırken; Destek Vektör Makineleri modelleri en az 4-5 parametre ayarlamak gerekmektedir. Ancak yaş ve cinsiyet parametreleri eklendiğinde Destek Vektör Makineleri daha iyi sonuç verdiği gözlemlenmektedir. Model

yapımının kolaylığı göz önüne alındığında, Rastgele Orman ve iyi performans sağlaması açısından ile Destek Vektör Makineleri AA tanısında klinik kullanım için daha iyi bir modeldir.

Hekime destek olması açısından Rasgele Orman, Makine öğrenmesi için ise Destek Vektör Makineleri ile birbirini desteklemelidir.

- Bu çalışmada, Alvarado skorlarda belirtilen parametrelere farklı bir bakış açısı getirilmiştir. Klinik bulguların yanısıra WBC, NE% çeşitli durumlarda çeşitli değerlerle durumu değiştirmektedir. WBC ve NE% kritik değerlere bağlı olduğu tespit edilmiştir. Bu değerler belli koşullarda hastalara farklı etki ettiği görülmüştür. Alvarado'dan farklı değerler bulunmuştur.
- Logistik Regresyon analizi ile modele katkı sağlayan niteliklerin β katsayıları sıralanmış ve hangi niteliğe öncelikle bakılması gerektiği tavsiyesinde bulunulmuştur. Normal, SMOTE, ROS ve RUS algoritmaları incelendiğinde ilk sırayı Rebound almaktadır ateş ve WBC bunu takip etmektedir.

Bu çalışma için performansı en yüksek modellemelerle (RO 'da nestimator=100 ve Lojistik Regresyon'da radial) değişken önemliliği analizi yapılmıştır. RO yönteminde WBC (lökosit), NE% ve Rebound en önemli üç nitelik olarak karşımıza çıkmaktadır. SMOTE ve ROS yöntemlerinde sağ alt kadrana ağrı göçü sınıflamaya çok az katkı sağladığı görülmektedir.

Doğruluk ve F- Ölçütüne bakıldığında yaş ve cinsiyet nitelikleri tüm sınıflandırma algoritmaları üzerinde performansı arttırıcı bir etki oluşturduğu görülmektedir. DVM algoritmasında %100'e varan yüksek başarı elde ettiği görülmüştür. Yaş ve cinsiyet parametreleri akut apandisit teşhisinde tüm sınıflandırma algoritmaları için nitelik olarak eklenmesi tavsiye edilmektedir.

Gelecekte yapılması tavsiye edilen çalışmalar aşağıdaki gibidir:

- Veri sayısı arttırılarak sınıflandırma algoritmaları uygulanabilir. Daha çok verilerle içe sinilecek bir model oluşturulabilir ve anlamlı veri üretilir.
- Alvarado Skoruna bağımlılığı ortadan kaldırmak için kümeleme yapılabilir.
- Birliktelik kuralı ile AA olanların hangi özellikleri bir arada görüldüğünün tespiti yapılabilir.
- Farklı yeniden örnekleme yöntemleri kullanılabilir.

- Farklı sınıflandırma algoritmaları kullanılabilir.
- Karar Destek Sisteminin hekimler tarafından kullanılması sağlanıp karar destek sisteminin sonuçları, doktorun verdiği kararlarla tamamen örtüşüp örtüşmediği araştırılabilir.
- Çalışmaya USG ve BT, CRP değerleri eklenerek sınıflandırma yapılabilir.
- Bu çalışma daha sonra hekimlerin ihtiyaç duyduğu farklı klinik karar verme alanlarında yeni algoritmalarla ülkemiz gerçeklerine uygun olarak geliştirilmesi sağlanabilir.

AA ve nadir görülen hastalıkların veri setleri dengesiz veri setlerinden oluşmaktadır. Bu çalışmada uygulanan adımlar diğer hastalıkların sınıflandırması için örnek model teşkil etmektedir. Öncelikle veriler dengeli hale getirilmeli, performans ölçümlerinden doğruluk değeri yanında diğer ölçümlere ve bu çalışma da özellikler f-ölçütüne bakılıp değerlendirme yapılmıştır.

Sonuç olarak, sağlık sektöründe problem olan AA teşhisinin en iyi bir şekilde yapılmasını amaçlayan tez çalışmamızda en iyi tahmin yapan sınıflandırma algoritması belirlenerek bilime katkı sağlanması amaçlanmıştır. Oluşturulan karar destek sisteminin hekimler tarafından kullanılıp insan sağlığına olumlu katkıda bulunması en büyük isteğimizdir.

KAYNAKLAR

- Abraham, A., 2004, Meta-Learning Evolutionary Artificial Neural Networks, *Neurocomputing Journal*, 56, 36-37.
- Adams, I. D., Chan, M., Clifford, P. C., Cooke, W. M., Dallos, V., de Dombal, F. T., Edwards, 1986, Computer-aided diagnosis of acute abdominal pain: a multicentre study, *British Medical Journal*, 293, 800-804.
- Akay, M., 1992, Noninvasive diagnosis of coronary artery disease using a neural network algorithm, *Biological Cybernetics*, 67, 361-367.
- Alvarado, A., 1986, A practical score for the early diagnosis of acute appendicitis, *Annals of Emergency Medicine*, 15 (5), 557-564.
- Andersson, R. E., 2007, The natural history and traditional management of appendicitis revisited: spontaneous resolution and predominance of prehospital perforations imply that a correct diagnosis is more important than an early diagnosis, *World Journal of Surgery*, 31(1), 86-92.
- Archana, S. ve diğ., 2014, Survey of Classification Techniques in Data Mining, *International Journal of Computer Science and Mobile Applications*, 2, 65-71.
- Baesens, B., Vlasselaer, V. V. ve Verbeke, W., 2015, *Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques*, John Wiley & Sons, USA, 31.
- Baxt, W. G., 1995, Application of artificial neural networks to clinical medicine, *Lancet*, 346(8983), 8.
- Bergeron E., 2006, Clinical judgement remains of great value in the diagnosis of acute appendicitis, *J Can Chir*, 49, 96-100.
- Binnebösel, M., Otto, J., Stumpf, M., Mahnken, A., H., Gassler, N. ve Schumpelick, V., 2009, Acute appendicitis Modern diagnostics-surgical ultrasound, *Chirurg*, 80, 579-587.
- Blagus R. ve Lusa, L., 2017, Gradient boosting for high-dimensional prediction of rare events, *Computational Statistics and Data Analysis*, 13, 19-37.
- Boser, B. E., Guyon, I. M. ve Vapnik, V. N., 1992, A training algorithm for optimal margin classifiers, *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ACM, 44-152.
- Bhangu, A., Soreide, K., Di Saverio, S., Assasson, J.H. ve Drake, F.T., 2015, Acute appendicitis: modern understanding o pathogenesis, diagnosis, and management, *Lancet*, 386, 1278-1287.
- Breiman, L., Random forests, 2001, *Machine Learning* , 45, 5-32.

- Birnbaum, B. A., Wilson, S. R., 2000, Appendicitis at the millennium, *Radiology*, 215(2), 337-48.
- Birnbaum, B. A. ve Jeffrey, R. B, Jr., 1998, CT and sonographic evaluation of acute right lower quadrant abdominal pain, *AJR Am J Roentgenol*, 170, 361-71
- Bounds, D. G. ve Lloyd, P. J., 1990, A comparison of neural network and other pattern recognition approaches to the diagnosis of low back disorders, *Neural Networks*, 3, 583-591.
- Brereton, R.G. ve Lloyd, G.R., 2010, Support vector machines for classification and regression, *Analyst*, 135 (2), 230–267.
- Canu, S., Grandvalet, Y., Guigue, V. ve Rakotomamonjy, A., 2008, *SVM and Kernel Methods Matlab Toolbox*, <http://asi.insa-rouen.fr/enseignants/~arakotom/toolbox/index.html>, [Ziyaret Tarihi: 20 Mayıs 2018].
- Cartwright, H., 2015, *Artificial Neural Networks*, Springer Humana, 2nd ed., New York, 195-197.
- Chawla, N. V., Japkowicz, N. ve Kotcz, A., 2004, Special Issue on Learning from Imbalanced Data Sets, *Sigkdd Explorations*, 6(1), 1–6.
- Chawla, N. V., Bowyer, K. W., Hall, L. O. ve Kegelmeyer, W. P., 2002, SMOTE: Synthetic Minority Over-sampling Technique, *J Artif Intell Res*, 16, 321–357.
- Chalazonitis, A. N., Tzovara, I., Sammouti, E., Ptohis, N., Sotiropoulou, E., Protopappa, E., 2008, CT in appendicitis, *Diagn. Interventional Radiol.*, 14 (1), 19-25.
- Cortes, C. ve Vapnik, V., 1995, Support-vector networks, *Machine Learning*, 20, 273-97.
- Cieslak, D., Chawla, N., ve Striegel, A., *Combating imbalance in network intrusion datasets*, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.874.9431&rep=rep1&type=pdf>, [Ziyaret Tarihi: 25.09.2018].
- Cleeve, S., Jones, N., Joshi, A., Misra, D., Phelps, S. ve Ward H., 2011, Trends in Childhood appendicitis, *British Association of Paediatric Surgeons*, 58th International Conference Belfast.
- Colson, M., Skinner, K. A. ve Dunnington, G., 1997, High negative appendectomy rates are no longer acceptable , *Am J Surg*, 174, 723.
- Cooper, G.F. ve Herskovits, E., 1992, A bayesian method for the induction of probabilistic networks from data, *Machine Learning*, 9, 309–347.
- Coyne, J. D., 2007, Lengthy appendices, *J Clin Pathol*, 60(5), 584.

- Cross, S. S., Harrison, R.F. ve Kennedy, R. L., 1995, Introduction to neural networks, *Lancet*, 346(8982), 1075.
- Cusheiri, A, Grace, P. A., Darzi, A ve ark., 2003, *Disorders of small intestine and vermiform appendix*, Clinical Surgery, 2nd ed, Blackwell Publishing Ltd, UK, 405.
- Çatak, F., Ö., 2015, *Bulut Bilişim Sistemlerinde Eşle/İndirge Yöntemi Uygulanarak Veri Madenciliği Yazılım Çatısının Geliştirilmesi*, Doktora Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü.
- Choy, K. L., Lee, W. B., ve Lo, V., 2003, Design of an intelligent supplier relationship management system: A hybrid case based neural network approach, *Expert Systems with Applications*, 24(2), 225–237.
- Dağ, Z., 2012, *Avuç İçi Terlemesi ile Kandaki Glikoz HbA1C Parametreleri Arasındaki İlişkinin Yapay Sinir Ağları Kullanılarak Belirlenmesi*, Doktora Tezi, Sakarya Üniversitesi Fen Bilimleri Enstitüsü.
- Dayhoff, J. E. ve DeLeo, J. M., 2001, Artificial Neural Networks-Opening the black box, *Cancer*, 91, 1615–1635.
- Dekhtyar, A., 2009, *CSC 466: Knowledge Discovery from Data - Distance/Similarity Measures*, [htDP://users.csc.calpoly.edu/~dekhtyar/560-Fall2009/lectures/lec09.466.pdf](http://users.csc.calpoly.edu/~dekhtyar/560-Fall2009/lectures/lec09.466.pdf), [Ziyaret Tarihi: 20 Mayıs 2018].
- DeKoning, E., P., 2016, Acute appendicitis, *Tintinalli's Emergency Medicine*, 8, 532-535.
- Deshmukh, S., Verde, F., Johnson, P. T., Fishman, E., K. ve Macura, K., J., 2014, Anatomical variants and pathologies of the vermiform appendix, *Emergency Radiology*, 21(5), 543-552.
- Dreiseitl, S. ve Ohno-Machado, L., 2002, Logistic regression and artificial neural network classification models: a methodology review, *J Biomed Inform*, 35, 352-359.
- Doğan, S., Bulut A. Ş., Karamürsel, B., S., 2012, Akut apandisitinin nadir ayırıcı tanısı: appendisk endometriozisi, *Turkish Journal of Surgery*, 28, 159-161.
- Domingos, P., 2012, A few useful things to know about machine learning, *Communications of the ACM*, 55 (10), 78–87.
- de Dombal, F., Leaper, D., Staniland, J., McCann, A. ve Horrocks, J., 1972, Computer-aided diagnosis of acute abdominal pain, *British Medical Journal*, 2, 9-13.
- Drew, P. J. ve Monson, J. R., 2000, Artificial neural networks, *Surgery*, 127, 3-11.
- Duda, R.O., Hart, P.E. ve Stork, D.G., 2000, *Pattern Classification*, 2nd edition., WileyInterscience, USA, ISBN: 978-0-471-05669-0.

- Eberhart, R., Dobbins, R. ve Hutton, L., 1991, Neural network paradigm comparisons for appendicitis diagnoses, *Computerbased medical systems, proceedings of the Fourth Annual IEEE Symposium*, Baltimore, 298-304.
- Ellis, H., 1986, The 100th birthday of appendicitis, *British Medical Journal*, 293(6562), 1617-1618.
- Escribá, A., Gamell, A. M., Fernández, Y., Quintillá, J. M. ve Cubells, C. L., 2011, Prospective validation of two systems of classification for the diagnosis of acute appendicitis, *Pediatric Emergency Care*, 27, 165-9.
- Eskelinen, M., Ikonen, J. ve Lipponen, P., 1992, Clinical diagnosis of acute appendicitis. A prospective study of patients with acute abdominal pain, *Theoretical Surgery*, 7, 81-85.
- Fenyő, G., 1987, Routine use of a scoring system for decisionmaking in suspected acute appendicitis in adults, *Acta Chir Scand*, 153, 545-551.
- Flach, P., 2012, *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*, 1st, Cambridge University Press Glasgow, UK, ISBN: 978-1-107-09639-4.
- Fleiss, J.L. Levin, B. ve Paik, M.C., 2013, Statistical methods for rates and proportions, *John Wiley & Sons*.
- Flum, D. R., Koepsell, T., 2008, The clinical and economic correlates of misdiagnosed appendicitis: nationwide analysis, *Archives of Surgery*, 137, 799-804.
- Friedman, N., Geiger, D. ve Goldszmidt, M., 1997, Bayesian network classifiers, *Machine Learning*, 29(2-3), 131-163.
- Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M. ve Haussler, D., 2000, Support vector machine classification and validation of cancer tissue samples using microarray expression data, *Bioinformatics*, 16 (10), 906-914.
- Fyfe, C., 2000, *Artificial neural networks and information theory*, <http://index-of.co.uk/InformationTheory/Neural%20Networks%20And%20Information%20Theory%20-%20Colin%20Fyfe.pdf>, [Ziyaret Tarihi: 02 Haziran 2018].
- Galar, M., Fernández, A., Tartas, E., B., Sola, H., B. ve Herrera, F., A., 2012, Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and HybridBased Approaches, *IEEE Trans Systems, Man, and Cybernetics*, 42(4), 463-484.
- Geng, Haijiang, Lu, Tao, Lin, Xiao, Liu, Yu, Yan, Fangrong., 2015, Prediction of proteinprotein interaction sites based on naïve bayes classifier, *Biochem Res Int*, 2015, 1-7.
- Genes, N, Kim, M., S., Thum, F., L., 2015, Usability evaluation of a clinical decisionsupport system for geriatric ED pain treatment, *Applied Clinic Informatics*, 7 (1) , 128-130.
- Ghorbani, A., Forouzesh, M. ve Kazemifar, A. M., 2014, *Variation in Anatomical Position of Vermiform Appendix among Iranian Population: An Old Issue Which Has Not Lost Its*

Importance, <https://www.hindawi.com/journals/ari/2014/313575.pdf>, [Ziyaret Tarihi: 25 Haziran 2018].

- Gopal, M. ve Jaffrey, B., 2011, The consequences of ignoring evidence base in management of acute appendicitis in children, *British Association of Paediatric Surgeons, 58th International Conference Belfast*.
- Guidry, S. P., Poole, G. V., 1994, The anatomy of appendicitis, *The American Surgeon*, 60, 68–71.
- Guo, H., Li, Y., Shang, J., Gu, M., Huang, Y. ve Gong, B., 2017, Learning from class-imbalanced data: review of methods and applications, *Expert Systems with Application*, 73, 220–239.
- Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002, Gene selection for cancer classification using support vector machine, *Machine Learning*, 46, 389–422.
- Hale, D. A., Molloy, M., Pearl, R. H., Schutt, D. C. ve Jaques, D. P., 1997, Appendectomy: a contemporary appraisal, *Annals of Surgery*, 225, 252–61.
- Han, J. ve Kamber, M., 2006, *Data mining: concepts and techniques*, Morgan Kaufmann, USA.
- Han, J., Kamber, M. ve Pei, J., 2012, *Data Mining Concepts and Techniques*, 3rd (Ed), Morgan Kaufmann Publishers, USA.
- Han Wu, Shengqi Yang, Zhangqin Huang, Jian He, Xiaoyi Wang, 2018, Type 2 diabetes mellitus prediction model based on data mining, *Informatics in Medicine Unlocked*, 10, 100–107.
- Hanif, M., S., Tahir, T. H., Sheikh, I. A., Ranjha, M. Z., 2010, Acute appendicitis: gaining time in mass casualty scenario, *Pak Armed Forces Med J.*, 3, 1–6.
- Haykin, S., 1999, *Neural Networks, A Comprehensive Foundation*, Prentic-Hall, Inc., New Jersey.
- Haykin, S. O. ,2008, *Neural Networks and Learning Machines*, 3rd ed., Pearson, New York.
- He, H. ve Garcia, E. A., 2009, Learning from Imbalanced Data. Knowledge and Data Engineering, *IEEE Transactions*, 21(9), 1263–1284.
- Hindi, K.E., 2014, Fine tuning the naive Bayesian learning algorithm, *AI Commun*, 27(2), 133–141.
- Holte, C., Drummond, R. C., 2003, C4.5, class imbalance, and cost sensitivity: why UnderSampling beats Over-Sampling, *Workshop on Learning from Imbalanced Datasets II, ICML, Washington DC*, 1–8.
- Ho, T. K., 1995, Random Decision Forests, *IEEE Computer Society*, 1, 278.

- Hosmer, D. W., Lemeshow, S. ve Sturdivant, R. X., 2013, *Applied logistic regression*, 3rd ed., Wiley Hoboken, New Jersey, ISBN: 978-0-470-58247-3.
- Hsieh, C. H., , Lu, R. H., Lee, N. H., Chiu, W.T, Hsu, M. H. ve Li, Y.C., 2011, Novel solutions for an old disease: diagnosis of acute appendicitis with random forest, support vector machines, and artificial neural networks, *Surgery*, 149, 87-93.
- Huang, Y. M., Hung, C. M., ve Jiau, H. C., 2006, Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem, *Nonlinear Anal. R. World Appl.*, 7, 720–747.
- Humes, J., Simpson, J., 2006, Acute appendicitis, *BMJ*, 333, 530-534.
- Jaffe, B. M. ve Berger, D. H., 2010, *The Appendix*, Schwartz' s Principles of Surgery, 9th ed. New York, The McGraw-Hill Companies, 1073-89.
- Jiang, L., Zhang, L., Li, C. ve Wu, J., 2019, A correlation-based feature weighting filter for naive Bayes, *IEEE Trans. Knowl. Data Eng.*, 31, 201-213.
- Jiang, L., Zhang H. ve Cai, Z., 2009, A novel Bayes model: hidden naive Bayes, *IEEE Trans. Knowl. Data Eng.* 21, 1361–1371.
- Jock A., Blackard J. ve Dean D.,1999, Comparative Accuracies of Artificial Neural Networks and Discriminant Analysis in Predicting Forest Cover Types From Cartographic Variables, *Computers and Electronics in Agriculture*, 24, 131-151.
- Jones, P. F., 2001, Suspected acute appendicitis: trends in management over 30 years, *Br J Surg*, 88, 1570-1577.
- Kanumba, E.S., Mabula, J.B., Rambau, P., Chalya, P.L., 2011, Modified Alvarado scoring system as a diagnostic tool for acute appendicitis at Bugando Medical Centre, *BMC Surgery*, 11, 1–5.
- Karaağaoğlu, E., Karakaya, J. ve Kılıçkap, M., 2016, *Tanı Testlerinin Değerlendirilmesinde İstatistiksel Yöntemler*, Detay Yayıncılık, Ankara.
- Kartal, E., 2015, *Sınıflandırmaya Dayalı Makine Öğrenmesi Teknikleri ve Kardiyolojik Risk Değerlendirmesine İlişkin Bir Uygulama*, Doktora Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü.
- Kavic, M. S., Kavic, S. M. ve Kavic, S. M., 2010, Laparoscopic Appendectomy, *Society of Laparoendoscopic Surgeons*, 3, 239-48.
- Kessler R., C., Stein M., B., P, M., V., Bliese P., Bossarte R., M., Bromet E., J., 2016, Predicting suicides after ouDPatient mental health visits in the army study to assess risk and resilience in servicemembers (army starrrs), *Mol Psychiatry*, 22, 544–51.

- Khan, I. ve Rehman, A.U., 2005, Application of Alvarado scoring system in diagnosis of acute appendicitis, *J Ayub Med Coll Abbotabad*, 17, 41-44.
- Khan, S., 2006, Evaluation of hyperbilirubinemia in acute inflammation of appendix: a prospective study of 45 cases, *Kathmandu Univ Med J(KUMJ)*, 4, 281-9.
- Khoong, C. M., 1995, Decision support systems: an extended research agenda, *Omega*, 23(2), 221-229.
- Kong, G., Xu, D. L., Yang, J. B., Yin, X., Wang, T., Jiang, B. ve Hu, Y., 2016, Belief rule-based inference for predicting trauma outcome, *Knowledge Based System*, 95, 35-44.
- Krishnaveni, C. ve Sobharani, T., 2011, On the Classification of Imbalanced Datasets, *International Journal of Computer Science and Technology*, 2(1), 145- 148.
- Kwok, My., Kim, M.K. ve Gorelick, M.H., 2004, Evidence based approach to the diagnosis of appendicitis in children, *Pediatric Emergency Care*, 20, 690-698.
- Kulkarni, V.Y., Sinha, P.K. ve Petare, M.C., 2016, Weighted hybrid decision tree model for random forest classifier, *J. Inst. Eng.*, 97, 209-217.
- Kulkarni, S. ve Harman, G., 2011, *An Elementary Introduction to Statistical Learning Theory*, 1st ed., John Wiley & Sons, Inc., Singapore, ISBN: 978-1-118- 02343-3.
- Landis, J.R. ve Koch, G.G., 1977, The measurement of observer agreement for categorical data, *Biometrics*, 33, 159-174.
- Langley, P., Iba, W. ve Thomas, K., 1992, An analysis of Bayesian classifiers, *Proceedings of the Tenth National Conference of Artificial Intelligence AAAI Press*, 223-228.
- Larose, D.T., 2014, *Discovering knowledge in data: an introduction to data mining*, 2nd ed., John Wiley & Sons, Canada.
- LaRosa, J., Ahmad, N., Feinberg M., Shah, M., DiBrienza, R. ve Studer, S., 2012, The use of an early alert system to improve compliance with sepsis bundles and to assess impact on mortality, *Crit. Care Res. Pract.*, 1-8.
- Lewis, D.D., 1998, Naive (Bayes) at forty: the independence assumption in information retrieval, *Lect. Notes Comput. Sci.*, 1398 (15), 4-15.
- Li, Y., Liu, L., Chiu, W. ve Jian, W. S., 2000, Neural network modeling for surgical decisions on traumatic brain injury patients, *Int J Med Inform*, 57, 1-9.
- Liang, M. K., Andersson, R. E., Jaffe, B. M. ve Berger, D. H., 2014, The Appendix, *Schwartz's Principles of Surgery*, 10, 1241-1262.
- Limpawattanasiri, C., 2011, Alvarado score for the acute appendicitis in a provincial hospital, *J Med Assoc Thai.*, 94, 441-448.

- Lin, C. C., Wang, Y. C., Chen, J. Y., Liou, Y. J., Bai, Y. M. ve Lai, I. C., ve ark., 2008, Artificial neural network prediction of clozapine response with combined pharmacogenetic and clinical data, *Comput Methods Programs Biomed*, 91, 91-99.
- Lin, C. S., Chiu, J. S., Hsieh, M. H., Mok, M. S., Li, Y. C. ve Chiu, H. W., 2008, Predicting hypotensive episodes during spinal anesthesia with the application of artificial neural networks, *Comput Methods Programs Biomed*, 92, 193-197.
- Liu, X. Y., Wu, J. ve Zhou, Z.H., 2009, Exploratory undersampling for class-imbalance learning, *IEEE Transactions on Systems, Man, and Cybernetics*, 39(2), 539–550.
- Lu, W. Z., ve Wang, D., 2008, Ground-level ozone prediction by support vector machine approach with a cost-sensitive classification scheme, *Sci. Total. Environ.*, 395, 109–116.
- Lunardon, N., Menardi, G., Torelli, N., 2014, ROSE: a Package for Binary Imbalanced Learning, *R Journal*, 6(1), 82–92.
- Macklin, P. S., Dempsey, J., Brooks, J. ve Rand, J., 1991, Using neural networks to diagnose cancer, *J Med Sys*, 15, 11-19.
- Mardan, M. A., Mufti, T. S., Khattak, I. U., ve diğ., 2007, Role of ultrasound in acute appendicitis, *J Ayub Med Coll Abbottabad*, 19, 72-78.
- Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A., ve Tourassi, G. D., 2008, Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance, *Neural Netw.*, 21, 427–436.
- McGreevy, J. M., Finlayson, S. R., Alvarado, R., Laycock, W. S., Birkmeyer, C. M. ve Birkmeyer, J.D., 2002, Laparoscopy may be lowering the threshold to operate on patients with suspected appendicitis, *Surg Endosc*, 16, 1046-1049.
- Menardi, G. ve Torelli, N., 2014, Training and Assessing Classification Rules with Imbalanced Data, *Data Min Knowl Discov.*, 28(1), 92–122.
- Miranda, Eka, Irwansyah, Edy, Alowisius, Y, Amelga, Marco M, Maribondang Mulyadi, ve diğ., 2016, Detection of cardiovascular disease risk's level for adults using naive Bayes classifier, *Healthc Inform Res*, 22(3), 196–205.
- Mishra, V. K., Joshi, P, Shah, JV, Agrawal, C, Sharma, D, Aggarwal, K ve Amyand's Hernia, 2013, A Case of an Unusual Inguinal Herniace, *Indian J Surg*, 75(1), 469-71.
- Mitchell, T. M., 1997, *Machine Learning*, 1st Edition, McGraw-Hill Science/Engineering/Math, ISBN: 0-07-042807-7.
- Mohri, M., Rostamizadeh, A. ve Talwalkar, A., 2012, *Foundations of Machine Learning*, The MIT Press, ISBN: 0-262-01825-X.

- Munir, K., Iqbal, J., Mushtaq, U., Ishaque, I., Mudassar, J., Khalid, A., 2008, Modified Alvarado scoring system in the diagnosis of acute appendicitis, *APMC*, 2, 91-94.
- Nauck, D. ve Kruse, R., 1999, Obtaining interpretable fuzzy classification rules from medical data, *Artif. Intell. Med.*, 16, 149–169.
- Noble, W.S., 2006, What is a Support vector machine, *Nat Biotechnol*, 24(12), 1564-1567.
- O’Connel, P.R., 2008, *The vermiform appendix*, in: N.S. Williams, C.J. Bulstrode, P.R. O’Connel (Eds.), 25th ed., Bailey and Love’s Short Practice of Surgery, Edward Arnold, London, 1204-1218.
- Ohmann, C., Yang, Q. ve Franke, C., 1995, Diagnostic scores for acute appendicitis, *Abdominal Pain Study Group. Eur J Surg*, 161, 273-81.
- Owen, T., D., Williams, H., Stiff, G., Jenkinson, L. R. ve Rees, B., I., 1992, Evaluation of Alvarado scoring in acute appendicitis, *J R Soc Med.*, 85, 87-88.
- Öztemel, E., 2003, *Yapay Sinir Ağları*, Papatya yayıncılık, İstanbul, 187- 196.
- Park, H. ve Kwon, H.C., 2011, Improved Gini-Index Algorithm to correct feature-selection bias in text classification, *IEICE Trans. Inf. Syst.*, 173, 855–865.
- Paterson, B., S., 2003, Acute appendicitis, *Hamilton Bailey’s Emergency Surgery*, 13, 399.
- Pepe, M. S., 2003, *The Statistical Evaluation of Medical Tests for Classification and Prediction*, Oxford Statistical Sciences Series, UK, 978-0198565826.
- Pesonen, E., Juhola, M., Ikonen, J. ve Eskelinen, M., 1994, A tentative automatic diagnosis of acute abdomen using neural networks, *Finnish Artificial Intelligence Society*, 59-66.
- Phillips, S., Wiles, J ve Schwartz, S, 1991, A comparison of three classification algorithms on the diagnosis of abdominal pains, *In: Proceedings of the Second Australian Conference of Neural Networks (ACNN 91)*, Sydney University Press, 283-287.
- Pieper R, Keger, L ve Niisman, P., 1982, Acute appendicitis: A clinical study of 1018 cases of emergency appendectomy, *Acta Chir Stand*, 148, 51-62.
- Poole David, L. ve Mackworth Alan, K., 2010, *Artificial Intelligence: foundations of computational agents*, Cambridge University Press, USA.
- Powers, D. M. W., 2007, *Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation*. Adelaide, Australia: School of Informatics and Engineering, <https://csem.flinders.edu.au/research/techreps/SIE07001.pdf>, [Ziyaret Tarihi: 05 Temmuz 2018].

- Prabhudesai, S., G., Gould, S., Rekhraj, S., Tekkis, P. P., Glazer, G. ve Ziprin, P., 2008, Artificial neural networks: useful aid in diagnosing acute appendicitis, *World J Surg*, 32, 305-9
- Quinlan, J.R., 1986, Induction on decision tree, *Mach. Learn.* 1, 81–106.
- Rai, P., 2011, *Model Selection and Feature Selection* (CS5350/6350: Machine Learning), <http://www.cs.utah.edu/~piyush/teaching/22-9-print.pdf>, [Ziyaret Tarihi: 02 Haziran 2018].
- Reynolds, S.L., Jaffe, D.M., 1992, Diagnosing abdominal pain in a paediatric emergency department, *Pediatr. Emerg. Care*, 8,126–128.
- Richards, J. A. ve Jia, X., 2006, *Remote Sensing Digital Image Analysis*, Springer, Berlin.
- Rioux, M., 1992, Sonographic detection of the normal and abnormal appendix, *AJR Am J Roentgenol*, 158, 773-8.
- Rosenblatt, F., 1962, *Principles in neurodynamics*, Springer, ISBN: 978-3-642-70913-5, 245-248.
- Sakai, S., Kobayashi, K., Toyabe, S. , Mandai, N., Kanda, T. ve Akazawa, K., 2007, Comparison of the levels of accuracy of an artificial neural network model and alogistic regression model for the diagnosis of acute appendicitis, *J Med Syst*, 31, 357-64.
- Sammalkorpi, H. E., Mentula, P., Savolainen, H., Leppäniemi, A., 2016, The Introduction Of Adult Appendicitis Score Reduced Negative Appendectomy Rate, *Scandinavian Journal of Surgery*, 106(3), 196-201.
- Sargent, D. J., 2001, Comparison of artificial neural networks with other statistical approaches- results from medical data sets, *Cancer*, 91(8), 1636-1642.
- Schapire, R., 2008, *COS 511: Theoretical Machine Learning*, http://www.cs.princeton.edu/courses/archive/spr08/cos511/scribe_notes/0204.pdf, [Ziyaret Tarihi: 22.04.2018].
- Scholer, S., Pituch, K., Orr, D., ve ark., 1996, Clinical outcomes of children with acute abdominal pain, *Pediatrics*, 98, 680–685.
- Seetahal, S., A., Bolorunduro, O., B., Sookdeo, T., C., ve diğ., 2011, Negative appendectomy: a 10-year review of a nationally representative sample, *American journal of surgery*, 201(4), 433-437.
- Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y. ve Wang, Z., 2007, A novel feature selection algorithm for text categorization, *Expert Syst. Appl.* , 33, 1-5.

- da Silva, I.N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L.H.B. ve dos Reis Alves, S.F., 2017, *Artificial Neural Network Architectures and Training Processes, Artificial Neural Networks*, 21, 21-29.
- Singh, K., Gupta, S. ve Parga, P., 2008, Application of Alvarado scoring system in the diagnosis of acute appendicitis, *JK Sci.*, 10, 84-86.
- Smith, P. H., 1965, The diagnosis of appendicitis, *Postgrad Med J*, 41, 2-5.
- Sokolova, M., Japkowicz, N. ve Szpakowicz, S., 2006, Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation, *Australasian Joint Conference on Artificial Intelligence*, 1021.
- Soucy, P., Mineau, G.W., 2001, A simple knn algorithm for text categorization, in: *Data Mining, ICDM*, 647-648.
- Steven L., L., Alicia, J., W. ve Hung, S., H., 2001, Computed tomography and ultrasonography do not improve and may delay the diagnosis and treatment of acute appendicitis, *Arch Surg*, 136, 556- 562.
- Sun, J., Lang, J, Fujita, H., Li, H., 2018, Imbalanced Enterprise evaluation with DTE-SBD: Decision Tree Ensemble Based on SMOTE and Bagging with Differentiated Sampling Rates, *Information Sciences*, 78-79.
- Suzuki K., 2013, *Artificial Neural Networks, Architectures and Applications*, Rijeka, InTech Janeza Trdine, ISBN 978-953-51-0935-8.
- Tan, P.N., Steinbach, M. ve Kumar, V., 2006, *Introduction to Data Mining*, <https://www-users.cs.umn.edu/~kumar001/dmbook/sol.pdf>, [Ziyaret Tarihi: 05 Şubat 2019].
- Thai-Nghe, N., GanDNeer, Z., Schmidt-Thieme, L., 2011, A new evaluation measure for learning from imbalanced data, In: *Neural Networks (IJCNN), The 2011 International Joint Conference*, 537–542.
- Testi, D., Cappello, A., Chiari, L., Viceconti, M. ve Gnudi, S., 2001, Comparison of logistic and bayesian classifiers for evaluating the risk of femoral neck fracture in osteoporotic patients, *Med Biol Eng Comput*, 39(6), 633–7.
- Todd, B. S ve Stamper R, 1994, The relative accuracy of a variety of medical diagnostic programs, *Meth Med*, 33, 402-415.
- Turhan, A. N. ve Kapan, S., 2000, Akut Apandisit, *Acil Cerrahi*, 2, 316.
- Vapnik, V. N., 1995, *The nature of statistical learning theory*, 2nd ed., Springer-Verlag New York, USA.

- Vassilis Z. A, Dimitris M. P. ve Vassilis G. A., Athanasios V. A., 2019, Solar radiation estimation methods using ANN and empirical models, *Computers and Electronics in Agriculture*, 160, 160–167.
- Vellido, A., Martin-Guerrero, J. D. ve Lisboa, P. J. G., 2012, *Making machine learning models interpretable*, Louvain-La-Neuve, ISBN: 978-2-87419-049-0.
- Wei, J. T., Zhang, Z., Barnhill, S. D., Madyastha, R., Zhang, H ve Oesterling, J. E, 1998, Understanding Artificial Neural Networks and exploring their potential application for the practicing urologist, *Urology*, 52, 161–352.
- Widrow, B. ve Stearns, S. D., 2003, Adaptive signal processing, *European Urology* ,43, 596-603.
- Widrow, B., Hoff, M., 1960, Adaptive switching circuits, *August IRE WESCON Convention Record*, 4, 96–104.
- Williams, G. R., 1983, Presidential Address: a history of appendicitis. With anecdotes illustrating its importance, *Ann Surg.*, 197(5), 495-506.
- Witten, I. H., Frank, E., Hall, M.A. ve Pal, C.J., 2016, *Data Mining: Practical machine learning tools and techniques*, 2nd ed., Morgan Kaufmann.
- Yang, H. R., Wang, Y. C., Chung, P. K., Chen, W. K., Jeng, L. B, Chen, R. J., 2006, Laboratory tests in patients with acute appendicitis, *ANZ J Surg*, 76, 71-7.
- Young, P., 2014, Appendicitis and its history, *Rev Med Chil*, 142(5), 667-72.
- Yegane, R., Peyvandi, H., Hajinasrollah, Salehei, N. ve Ahmadei M., 2008, Evaluation of modified Alvarado score in acute appendicitis among Iranian patients, *Acta Medica Iranica*, 46, 501-506.
- Yu, F.W., Ho, W.T., Ho, Chan, K.T., Sit, Chan R.K.Y., 2018, Logistic regression-based optimal control for aircooled Chiller, *International journal of refrigeration*, 85, 200–212
- Zhang, J. ve Mani, I., 2003, KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction, *In: Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets*, 42–48.
- Zhang, H. ve Su, J., 2004, Naive Bayesian classifiers for ranking. Machine Learning, *ECML*, 3201, 501–512.
- Zhang, S., 2019, *Cost-Sensitive KNN Classification*, Neurocomputing, <https://doi.org/10.1016/j.neucom.2018.11.101>, [Ziyaret Tarihi: 05 Şubat 2019].
- Zhang, X., 2014, *Machine Learning: Model Selection*, https://www.lri.fr/~xlzhang/KAUST/CS229_slides/c14_model_selection.pdf, [Ziyaret Tarihi: 02 Haziran 2018].

EKLER

EK 1. YSA kodlar ve çıktıları.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_recall_curve, auc, precision_score, recall_score, f1_score, roc_auc_score, n
import matplotlib.pyplot as plt
from IPython.display import Image
from keras.models import Sequential, Model
from keras.layers import Dense, Input
from keras.utils import plot_model
import numpy as np
from sklearn.pipeline import make_pipeline
from imblearn.pipeline import make_pipeline as make_pipeline_imb
df = pd.read_csv('C:/Users/kevser/Desktop/dataNewCY.csv', delimiter=';')
data = df.values
X = data[:,0:10]
y = data[:,10]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)
from collections import Counter
from imblearn.over_sampling import SMOTE
sm = SMOTE(ratio='minority')
X_sm, y_sm = sm.fit_sample(X, y)
print('Original dataset shape %s' % Counter(y))
print('SMOTE Resampled dataset dağılımı %s' % Counter(y_sm))
model = Sequential()
model.add(Dense(12, activation='relu', kernel_initializer='uniform', input_dim=10 ))
model.add(Dense(15, activation='relu', kernel_initializer='uniform'))
model.add(Dense(8, activation='relu', kernel_initializer='uniform'))
model.add(Dense(10, activation='relu', kernel_initializer='uniform' ))
model.add(Dense(1, activation='sigmoid', kernel_initializer='uniform'))
model.compile(loss='binary_crossentropy', optimizer='Adam', metrics=['accuracy'])
history = model.fit(X_sm, y_sm, epochs=500, batch_size=1024, verbose=1, validation_split=0.25)
y_pred = model.predict_classes(X_test)
classification_report(y_test, y_pred)
accuracy_score(y_test, y_pred)
precision_score(y_test, y_pred)
recall_score(y_test, y_pred)
f1_score(y_test, y_pred)
confusion_matrix(y_test, y_pred)

print("accuracy: {}".format(accuracy_score(y_test, y_pred)))
print("precision: {}".format(precision_score(y_test, y_pred)))
```

```

y_hat = model.predict(X_test)
model.add(Dense(1000, input_dim=8, activation='relu', kernel_initializer='uniform' ))
model.add(Dense(1, activation='sigmoid', kernel_initializer='uniform'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

history = model.fit(X_sm, y_sm, epochs=3000, batch_size=1024, verbose=1, validation_split=0.25)

y_hat = model.predict(X_test)
classification_report(y_test, y_pred)
print("classification: {}".format(classification_report(y_test, y_pred)))

accuracy_score(y_test, y_pred)
print("accuracy: {}".format(accuracy_score(y_test, y_pred)))
precision_score(y_test, y_pred)
print("precision: {}".format(precision_score(y_test, y_pred)))
recall_score(y_test, y_pred)
print("recall: {}".format(recall_score(y_test, y_pred)))
f1_score(y_test, y_pred)
print("f1_score: {}".format(f1_score(y_test, y_pred)))
confusion_matrix(y_test, y_pred)
print("cohen kappa: {}".format(cohen_kappa_score(y_test, y_pred)))
print("Confusion Matrix: {}".format(confusion_matrix(y_test, y_pred)))

#Eğitim tarihçesi
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.ylabel('dogruluk', fontsize=14)
plt.xlabel('ağa gösterim sayısı', fontsize=14)
plt.legend(['eğitim', 'test'], loc='upper left')
plt.show()

#Kayıp fonksiyon değişimi
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])

plt.ylabel('kayıp', fontsize=14)
plt.xlabel('ağa gösterim sayısı', fontsize=14)
plt.legend(['eğitim', 'test'], loc='upper left')

```

EK 2. Rasgele orman algoritması ROS yöntemi kodları.

```
def print_results(headline, true_value, pred):
    print(headline)
    print("accuracy: {}".format(accuracy_score(true_value,pred)))
    print("precision: {}".format(precision_score(true_value,pred)))
    print("recall: {}".format(recall_score(true_value,pred)))
    print("f1_score: {}".format(f1_score(true_value,pred)))
    print("cohen kappa: {}".format(cohen_kappa_score(true_value,pred)))

classifier=RandomForestClassifier(n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_le

df = pd.read_csv('C:/Users/kevse/Desktop/dataNewCV.csv',delimiter=';')
#matrise dönüştürme
data = df.values
X = data[:,0:10]
y = data[:,10]

x_train, x_test, y_train, y_test=train_test_split(X,y,test_size=0.25,random_state=2)

#normal model oluşturma
pipeline=make_pipeline(classifier)
model=pipeline.fit(x_train, y_train)
prediction=model.predict(x_test)

#ROS ile model oluşturma
ros_pipeline=make_pipeline_imb(RandomOverSampler(random_state=4),classifier)
ros_model=ros_pipeline.fit(x_train,y_train)
ros_prediction=ros_model.predict(x_test)
#veri artırdık
ros = RandomOverSampler()
X_ros, y_ros = ros.fit_sample(X, y)
print('Orjinal dataset dağılımı %s' % Counter(y))
print('RANDOM OVERSAMPLE Resampled dataset dağılımı %s' % Counter(y_ros))
#orjinal veri kümesi ile train ettik
clf = classifier
clf.fit(X=x_train,y=y_train)
#print(clf.feature_importances_)
#orjinal veri kümesinin test örnekleri ile conf matr
y_pred = clf.predict(x_test)
cm = confusion_matrix(y_test,y_pred)
print("Normal Conf Matr")
print(cm)
```

```

x_train_ros, x_test_ros, y_train_ros, y_test_ros=train_test_split(X_ros,y_ros,test_size=0.25,random_state=2)

# Arttırılmış over sample edilmiş veri kümesi eğitimi
clf_ros = classifier
clf_ros.fit(X=x_train_ros,y=y_train_ros)

#orjinal veri kümesinden alınmış olan test veri kümesinin oversample edilmiş veri ile eğitilen sınıflandırma
ros_prediction = clf_ros.predict(x_test)
print(classification_report(y_test,ros_prediction))
cm = confusion_matrix(y_test,ros_prediction)
print(cm)

#modelleri yazma
print()
print("normal data dağılımı: {}".format(Counter(y)))
x_ros, y_ros= RandomOverSampler().fit_sample(X,y)
print("Random OverSampling data dağılımı: {}".format(Counter(y_ros)))

#sınıflandırma raporu
print("Orjinal veri")
print(classification_report(y_test,prediction))
print("Random OverSampling")
print(classification_report_imbalanced(y_test,ros_prediction))

print()
print('Normal Pipeline score {}'.format(pipeline.score(x_test,y_test)))
print('Random OverSampling Pipeline Score {}'.format(ros_pipeline.score(x_test,y_test)))

print()
print_results("normal classification", y_test, prediction)
print()
print_results("Random OverSampling classification", y_test, ros_prediction)
print()

print("ROS Feature Importance")
importances_ros=clf_ros.feature_importances_
print(importances_ros)

```

EK 3. DVM algoritması için GUI kodları.

```

example_negative=[[21,2,1,1,1,0,1,0,1,1,8,5]]
result_negative=clf_smote.predict(example_negative)
print("Result positive",result_negative[0])

example_positive=[[20,2,1,1,1,0,1,1,1,1,9,1]]
result_positive=clf_smote.predict(example_positive)
print("Result positive",result_positive[0])

column_titles=["Yaş","Cinsiyet","Sağ Alt Kadran Ağrısı","Bulantı/Kusma",
               "İştahsızlık","Rebound Ağrı","Sağ Alt Kadranda Hassasiyet","Aksiler Ateş",
               "WBC","Nötrofilik Sola Kayma","Alvorado","AA Çeşit"]
column_count=len(column_titles)
data_record=[0] * column_count
data_vars=[]
data_entries=[]
main_window = tk.Tk()
main_window.title("Akut Apandisit")
main_window.geometry("360x480")
result_value = tk.StringVar()
result_label=tk.Label(main_window,textvariable=result_value)

def handle_find():
    print("Calculate")

    for i in range(column_count):
        data_record[i]=float(data_entries[i].get());
        print("data_record[i]", data_record[i])
    data=[data_record]
    result=clf_smote.predict(data)
    print(result[0])
    if(result[0]==0):
        result_label.config(bg='#00CC00')
        result_value.set("Negative")

    else:
        result_label.config(bg='red')
        result_value.set("Positive")

frame=tk.Frame(main_window)

```

```

title_value = tk.StringVar()
title_label=tk.Label(main_window,textvariable=title_value)
title_label.grid(row=0, column=1,columnspan=2,padx=3, pady=3)
title_value.set("HASTA VERİ GİRİŞİ")
title_label.config(font=('Helvetica 10 bold'))

for i in range(column_count):
    data_label=tk.Label(main_window,text=column_titles[i])
    data_label.grid(row=i+1, column=1,padx=3, pady=3)
    data_vars.append(tk.StringVar());
    data_label.config(anchor = tk.E,justify=tk.LEFT,font=('Helvetica 10 bold'))
    data_entries.append(tk.Entry(main_window,textvariable=data_vars[i]))
    data_vars[i].set(data_record[i])
    data_entries[i].grid(row=i+1, column=2,padx=3, pady=3)

explanation_value = tk.StringVar()
explanation_label=tk.Label(main_window,textvariable=explanation_value)
explanation_label.grid(row=column_count+2,column=1,padx=3, pady=3)

explanation_label.config(font=('Helvetica 10 bold italic'))
explanation_value.set("SONUÇ: ")

result_label.grid(row=column_count+2, column=2,padx=3, pady=3)

button_insert = tk.Button(main_window, text="Tahmin Et",command=handle_find, bg="#9999ff")
button_insert.grid(row=column_count+3, column=1,columnspan=2,padx=3, pady=3)

main_window.mainloop()

```

EK 4. DVM parametre ayarlaması (parameter tuning) kodları.

```

sm = SMOTE(random_state=42)
X_res, y_res = sm.fit_resample(X, y)
print('Orjinal dataset dağılımı %s' % Counter(y))
print('Smote Resampled dataset dağılımı %s' % Counter(y_res))

x_train_res, x_test_res, y_train_res, y_test_res=train_test_split(X_res,y_res,test_size=0.25,random_state=2)

from sklearn.preprocessing import StandardScaler
sc=StandardScaler()

x_train_res=sc.fit_transform(x_train_res)
x_test_res=sc.transform(x_test_res)

classifier=SVC(kernel='linear',random_state=0)
classifier.fit(x_train_res, y_train_res)

y_pred=classifier.predict(x_test_res)
cm=confusion_matrix(y_test_res,y_pred)

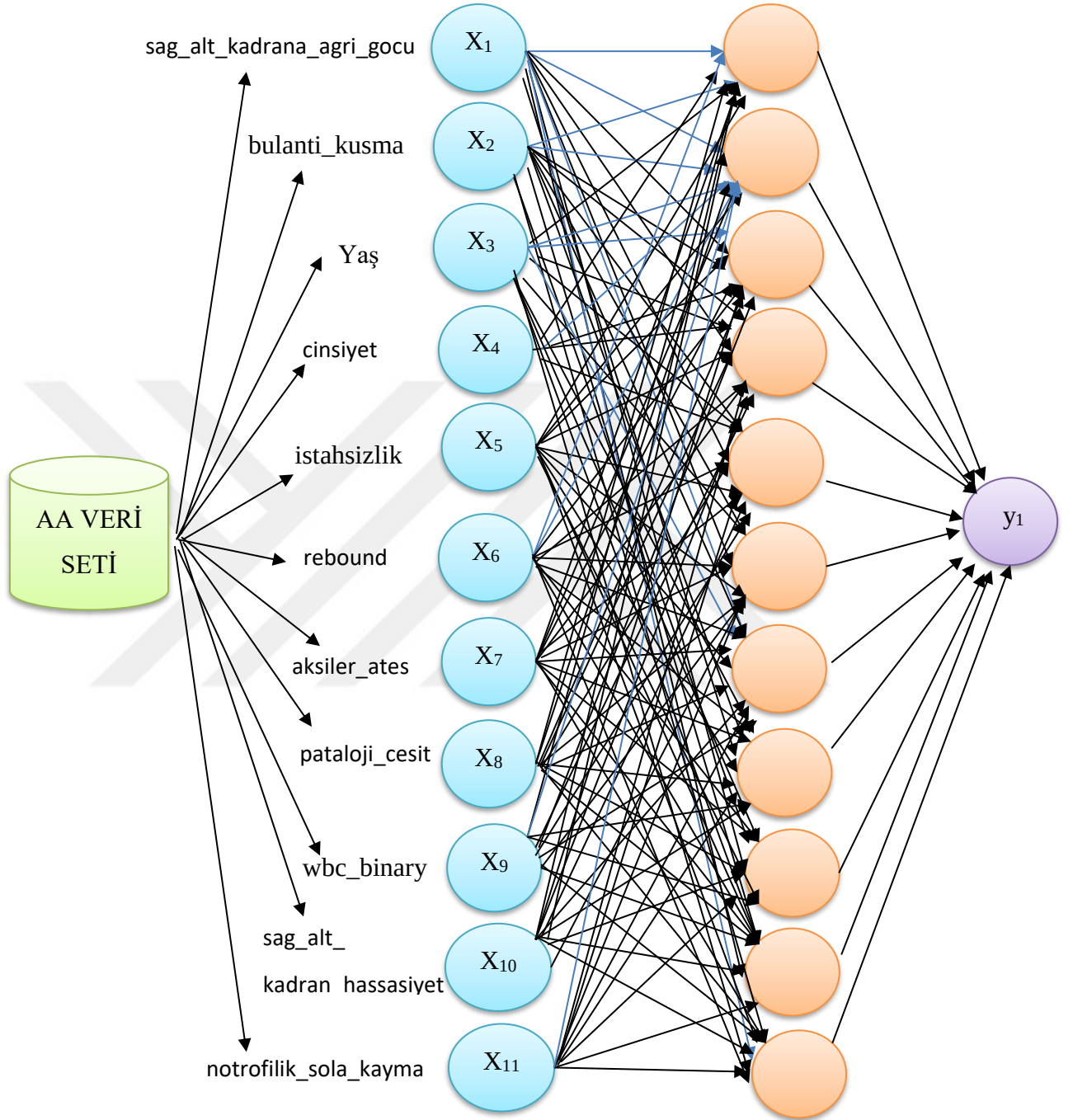
accuracy=accuracy_score(y_test_res,y_pred)
print(accuracy)

parameters = [
    {'C': [1, 10,100, 1000], 'kernel': ['linear']},
    {'C': [1, 10,100, 1000], 'gamma': [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9], 'kernel': ['rbf']},
    {'C': [1, 10,100, 1000], 'gamma': [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9], 'kernel': ['sigmoid']},
    {'C': [1, 10,100, 1000], 'gamma': [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9], 'degree':[1, 2, 3], 'kernel': ['poly']}
]
# Create a classifier object with the classifier and parameter candidates
clf_smote = GridSearchCV(estimator=svm.SVC(), param_grid=parameters, cv=5, n_jobs=-1)

# Train the classifier on data1's feature and target data
clf_smote.fit(X=X_res,y=y_res)
print('Best score for data1:', clf_smote.best_score_)
print('Best C:',clf_smote.best_estimator_.C)
print('Best Kernel:',clf_smote.best_estimator_.kernel)
print('Best Gamma:',clf_smote.best_estimator_.gamma)
print('Best Degree:', clf_smote.best_estimator_.degree)

```

EK 5. YSA bağlantıları örneği.



EK 6. T.C. medipol üniversitesi etik kurul kararı.

İSTANBUL MEDİPOL ÜNİVERSİTESİ
SOSYAL BİLİMLER BİLİMSEL ARAŞTIRMALAR ETİK KURULU
ETİK KURULU KARAR FORMU

BAŞVURU BİLGİLERİ	ARAŞTIRMANIN AÇIK ADI	Acil Sağlık Hizmetlerinde Konu Modelleme Tabanlı Karar Destek Sistem Önerisi			
	KOORDİNATÖR/SORUMLU ARAŞTIRMACI UNVANI/ADU/SOYADI	Kevser Şahinbaş			
	KOORDİNATÖR/SORUMLU ARAŞTIRMACININ UZMANLIK ALANI	Öğr. Gör.			
	KOORDİNATÖR/SORUMLU ARAŞTIRMACININ BULUNDUĞU MERKEZ	İstanbul			
	DESTEKLEYİCİ	-			
	ARAŞTIRMAYA KATILAN MERKEZLER	TEK MERKEZ ☐	ÇOK MERKEZLİ ☒	ULUSAL ☒	ULUSLARARASI ☐

İSTANBUL MEDİPOL ÜNİVERSİTESİ
SOSYAL BİLİMLER BİLİMSEL ARAŞTIRMALAR ETİK KURULU
ETİK KURULU KARAR FORMU

Değerlendirilen Belgeler	Belge Adı	Tarihi	Versiyon Numarası	Dili		
	ARAŞTIRMA PROTOKOLÜ/PLANI			Türkçe ☒	İngilizce ☐	Diğer ☐
	BİLGİLENDİRİLMİŞ GÖNÜLLÜ OLUR FORMU			Türkçe ☒	İngilizce ☐	Diğer ☐
Karar Bilgileri	Karar No: 46	Tarih: 29/07/2019				
	Yukarıda bilgileri verilen Sosyal Bilimler Bilimsel Araştırmalar Etik Kurulu başvuru dosyası ile ilgili belgeler araştırmanın gerekçe, amaç, yaklaşım ve yöntemleri dikkate alınarak incelenmiş ve araştırmanın etik ve bilimsel yönden uygun olduğuna "oybirliği" ile karar verilmiştir.					

İSTANBUL MEDİPOL ÜNİVERSİTESİ SOSYAL BİLİMLER BİLİMSEL ARAŞTIRMALAR ETİK KURULU	
BAŞKANIN UNVANI / ADI / SOYADI	Prof. Dr. Ali BÜYÜKASLAN

Unvanı/Adı/Soyadı	Uzmanlık Alanı	Kurumu	Cinsiyet		Araştırma ile İlişki		Katılım *		İmza
Prof. Dr. Ali BÜYÜKASLAN	İletişim Fakültesi	İstanbul Medipol Üniversitesi	E ☒	K ☐	E ☐	H ☒	E ☒	H ☐	
Doç. Dr. Mevlüt TATLIYER	İşletme ve Yönetim Bilimleri Fakültesi	İstanbul Medipol Üniversitesi	E ☒	K ☐	E ☐	H ☒	E ☒	H ☐	
Doç. Dr. Nurullah GÜR	İşletme ve Yönetim Bilimleri Fakültesi	İstanbul Medipol Üniversitesi	E ☒	K ☐	E ☐	H ☒	E ☒	H ☐	
Dr. Öğr. Üyesi İhsan EKEN	Medya ve Reklam Araştırmaları	İstanbul Medipol Üniversitesi	E ☒	K ☐	E ☐	H ☒	E ☒	H ☐	
Dr. Öğr. Üyesi Erol YILDIRIM	İnsan ve Toplum Bilimleri Fakültesi	İstanbul Medipol Üniversitesi	E ☒	K ☐	E ☐	H ☒	E ☒	H ☐	

* :Toplantıda Bulunma

EK 7. İstanbul esenyurt necmi kadioğlu devlet hastanesi izin yazısı.

26.07.2019

İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENFORMATİK BÖLÜM BAŞKANLIĞI'NA

İstanbul Esenyurt Necmi Kadioğlu Devlet Hastanesi başhekimi olduğum dönemde Klinik Karar Destek Sistemi hazırlığı için hastane sistemi veri tabanından retrospektif taranarak kayıt altına aldığım hasta verilerinden bir bölümü olan akut apandisit ön tanısıyla ameliyat olan hasta verilerinin İstanbul Üniversitesi Enformatik Bölümü doktora tezi hazırlayan Kevser Şahinbaş'ın "Acil Sağlık Hizmetlerinde Konu Modelleme Tabanlı Karar Destek Sistem Önerisi" adlı doktora tezi için kullanma izni uygundur.

Bilgilerinize arz ederim.


Dr. Öğr. Üyesi Ahmet Atilla YILMAZ
Beylikdüzü Devlet Hastanesi
Op. Dr. Ahmet Atilla YILMAZ
Genel Cerrahi Uzmanı
Diploma Tescil No: 49161

EK 8. Başkent üniversitesi tıp fakültesi genel cerrahi anabilim dalı uzmanlık tezi verileri için izin yazısı.

29.07.2019

İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENFORMATİK ANABİLİM DALI BAŞKANLIĞI'NA

Başkent Üniversitesi Tıp Fakültesi Genel Cerrahi Anabilim Dalı'nda Laparoskopik ve Tek İnsizyon Laparoskopik Apendektominin Prospektif Karşılaştırma Sonuçları adlı uzmanlık tezim için akut apandisit ön tanısı ile ameliyat edilen 74 hastanın verileri hastane sistemi veri tabanından retrospektif taranarak kayıt altına alındı. İstanbul Üniversitesi Enformatik Bölümü doktora tezi hazırlayan Kevser Şahinbaş'ın bu verileri "Acil Sağlık Hizmetlerinde Konu Modelleme Tabanlı Karar Destek Sistem Önerisi" adlı doktora tezi için kullanması uygundur.

Bilgilerinize arz ederim.

Doç.Dr.İlker Murat ARER

BAŞKENT ÜNİVERSİTESİ
ADANA DR. TURGUT NOYAN
UYGULAMA VE ARASTIRMA MERKEZİ
DOÇ.DR. İLKER M. ARER
DİP.TES. NO: 132367
GENEL CERRAHI UZMANI

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Kevser ŞAHİNBAŞ
Doğum Yeri	Çorum-Mecitözü
Doğum Tarihi	23.04.1982
Uyruğu	☒ T.C. ☐ Diğer:
Telefon	5333737641
E-Posta Adresi	kevsers@beykent.edu.tr
Web Adresi	



Eğitim Bilgileri	
Lisans	
Üniversite	Beykent Üniversitesi
Fakülte	İktisadi ve İdari Bilimler Fakültesi
Bölümü	Yönetim Bilişim Sistemleri
Mezuniyet Yılı	20.06.2019

Yüksek Lisans	
Üniversite	Beykent Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Bilgisayar Mühendisliği
Programı	Bilgisayar Mühendisliği

Doktora	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Enformatik Anabilim Dalı
Programı	Enformatik Programı

Makale ve Bildiriler
Assesment Approach with Mahara and Moodle in E-Learning, 5th International FutureLearning Conference on Innovations in Learning for the Future 2014 e-learning, May 5-7, İstanbul.
7th International Strategic Conference - A Performance Evaluation of The Turkish Banking Sector After The Global Crisis Via Camels Ratios 3.
7th International Strategic Conference - The Significance of Emotional Intelligence on The Innovative Work Behaviour of Managers As Strategic Decision-Makers
Akademik Bilişim Konferansı Şahinbaş, K., Güzelyazıcı, Ö. (2009). Bazı Kamu Kurumlarında Elektronik İmza Uygulaması ve Karşılaşılan Sorunlar: Eminönü Belediyesi Örneği, 601-604

Sanal Sınıf Uygulamalarının Karşılaştırılması, 26. Bilişim Kurultayı, İSTANBUL, 2009

