

**BEBEK AĞLAMALARININ MAKİNE ÖĞRENMESİ
ALGORİTMALARIYLA SINIFLANDIRILMASI**

ADEM EKİNCİ

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DANIŞMAN
DR. ÖĞR. ÜYESİ ENVER KÜÇÜKKÜLAHLI**

DÜZCE, 2023

T.C.
DÜZCE ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BEBEK AĞLAMALARININ MAKİNE ÖĞRENMESİ
ALGORİTMALARIYLA SINIFLANDIRILMASI

Âdem EKİNCİ tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Dr. Öğr. Üyesi Enver KÜÇÜKKÜLAHLI

Düzce Üniversitesi

Jüri Üyeleri

Dr. Öğr. Üyesi Enver KÜÇÜKKÜLAHLI

Düzce Üniversitesi

Doç. Dr. Nihat DALDAL

Bolu Abant İzzet Baysal Üniversitesi

Dr. Öğr. Üyesi Ahmet ALBAYRAK

Düzce Üniversitesi

Tez Savunma Tarihi: 24/07/2023

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

24 Temmuz 2023

Adem EKİNCİ



TEŐEKKÜR

Yüksek lisans öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı çok değerli danışman hocam Dr. Öğr. Üyesi Enver KÜÇÜKKÜLAHLI'ya en içten dileklerle teşekkür ederim.

Hayatım boyunca her zaman bana inanan, güvenen ve yanımda olan sevgili aileme, yüksek lisans eğitimim boyunca desteğini ve inancımı benden esirgemeyen kıymetli eşime ve canım oğluma çok teşekkür ederim.

24 Temmuz 2023

Adem EKİNCİ



İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ.....	vii
ÇİZELGE LİSTESİ.....	ix
KISALTMALAR.....	x
ÖZET	xi
ABSTRACT	xii
1. GİRİŞ.....	1
2. LİTERATÜR ÖZETİ.....	3
3. MATERYAL	9
3.1. VERİ SETİ	9
3.2. SES İŞLEME VE ÖZİNİTELİK ÇIKARMA.....	9
3.3. MAKİNE ÖĞRENMESİ ALGORİTMALARI	13
3.3.1. Random Forest (RF)	14
3.3.2. Destek Vektör Makinesi (Support Vector Machine - SVM)	16
3.3.3. Çok Katmanlı Algılayıcı (Multilayer Perceptron - MLP)	19
3.3.4. k-Nearest Neighbours (k-NN)	22
3.3.5. Extreme Gradient Boosting (XGBoost)	23
3.3.6. Lojistik Regresyon (LR)	25
3.3.7. K-Katlı Çapraz Doğrulama (K-Fold Cross Validation).....	26
3.4. PERFORMANS SKORLARI	28
3.4.1. Karışıklık Matrisi (Confusion Matrix).....	28
3.4.2. Doğruluk (Accuracy)	29
3.4.3. Duyarlılık (Recall).....	29
3.4.4. Kesinlik (Precision)	29

3.4.5. F1	30
3.4.6. ROC Eğrisi ve AUC Skoru.....	30
4. YÖNTEM	32
5. BULGULAR.....	36
5.1. XGBOOST ALGORİTMASI İLE ELDE EDİLEN BULGULAR	36
5.2. K-NN ALGORİTMASI İLE ELDE EDİLEN BULGULAR	39
5.3. MLP ALGORİTMASI İLE ELDE EDİLEN BULGULAR.....	42
5.4. RF ALGORİTMASI İLE ELDE EDİLEN BULGULAR	45
5.5. LR ALGORİTMASI İLE ELDE EDİLEN BULGULAR	48
5.6. SVM ALGORİTMASI İLE ELDE EDİLEN BULGULAR	51
5.7. ALGORİTMALARIN KARŞILAŞTIRILMASI.....	54
6. TARTIŞMA VE SONUÇ	55
7. KAYNAKLAR.....	58
ÖZGEÇMİŞ.....	63

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 3.1. Açlık sınıfına ait bir verinin sinyal ve mel-spectrogram görüntüsü	12
Şekil 3.2. MFCC akış şeması [25]	13
Şekil 3.3. Makine öğrenmesi metodolojisi	14
Şekil 3.4. Basit bir RF modeli [29]	16
Şekil 3.5. Hiper düzlem kullanarak sınıflandırma [30]	17
Şekil 3.6. Algılayıcı modeli [31]	20
Şekil 3.7. Basit bir MLP modeli [31]	21
Şekil 3.8. k-NN Modeli [33]	23
Şekil 3.9. K-Katlı Çapraz Doğrulama akış şeması [37]	28
Şekil 3.10. Genel olarak bir Karışıklık Matrisi	29
Şekil 3.11. ROC Eğrisi ve AUC skoru [38]	31
Şekil 4.1. Yöntemin Akış Şeması	32
Şekil 4.2. (a) karın ağrısı, (b) geğirme, (c) huzursuzluk, (d) açlık ve (e) yorgunluk sınıflarına ait birer spektogram görüntüsü	34
Şekil 5.1. Tüm veri setleri üzerinde XGBoost ile hesaplanan doğruluk skorları	37
Şekil 5.2. İki veri seti üzerinde XGBoost ile oluşturulan Karışıklık Matrislerinin karşılaştırması	38
Şekil 5.3. XGBoost ile (a) Ana veri Seti (b) vs13 veri seti kullanılarak elde edilen ROC Eğrileri ve AUC skorları	38
Şekil 5.4. Tüm veri setleri üzerinde k-NN ile hesaplanan doğruluk skorları	40
Şekil 5.5. İki veri seti üzerinde k-NN ile oluşturulan Karışıklık Matrislerinin karşılaştırması	41
Şekil 5.6:k-NN ile (a) Ana veri Seti (b) vs13 veri seti kullanılarak elde edilen ROC Eğrileri ve AUC skorları	41
Şekil 5.7. Tüm veri setleri üzerinde MLP ile hesaplanan doğruluk skorları	43

Şekil 5.8. İki veri seti üzerinde MLP ile oluşturulan Karışıklık Matrislerinin karşılaştırması.....	44
Şekil 5.9. MLP ile (a) Ana veri Seti (b) vs13 veri seti kullanılarak elde edilen ROC Eğrileri ve AUC skorları	44
Şekil 5.10. Tüm veri setleri üzerinde RF ile hesaplanan doğruluk skorları.....	46
Şekil 5.11. İki veri seti üzerinde RF ile oluşturulan Karışıklık Matrislerinin karşılaştırması.....	47
Şekil 5.12. RF ile (a) Ana veri Seti (b) vs15 veri seti kullanılarak elde edilen ROC Eğrileri ve AUC skorları	47
Şekil 5.13. Tüm veri setleri üzerinde LR ile hesaplanan doğruluk skorları	49
Şekil 5.14. İki veri seti üzerinde LR ile oluşturulan Karışıklık Matrislerinin karşılaştırması.....	50
Şekil 5.15. LR ile (a) Ana veri Seti (b) vs11 veri seti kullanılarak elde edilen ROC Eğrileri ve AUC skorları	50
Şekil 5.16. Tüm veri setleri üzerinde SVM ile hesaplanan doğruluk skorları.....	52
Şekil 5.17. İki veri seti üzerinde SVM ile oluşturulan Karışıklık Matrislerinin karşılaştırması.....	53
Şekil 5.18. SVM ile (a) Ana veri Seti (b) vs15 veri seti kullanılarak elde edilen ROC Eğrileri ve AUC skorları	53
Şekil 5.19. Kullanılan algoritmaların karşılaştırılması	54

ÇİZELGE LİSTESİ

	<u>Sayfa No</u>
Çizelge 3.1. Donate-a-cry Veri Setinin Sınıflara Dağılımı.....	9
Çizelge 4.1. Parçalara bölme yöntemi ile oluşan veri setleri ve veri sayıları	33
Çizelge 4.2. Ana veri seti ve vs02 veri seti için oluşan tensör	34
Çizelge 4.3. Sınıflandırma algoritmalarının bazı önemli parametreleri	35
Çizelge 5.1. İki veri seti üzerinde XGBoost ile performans skorlarının karşılaştırılması	37
Çizelge 5.2. İki veri seti üzerinde k-NN ile performans skorlarının karşılaştırılması	40
Çizelge 5.3. İki veri seti üzerinde MLP ile performans skorlarının karşılaştırılması	43
Çizelge 5.4. İki veri seti üzerinde RF ile performans skorlarının karşılaştırılması	46
Çizelge 5.5. İki veri seti üzerinde LR ile performans skorlarının karşılaştırılması	49
Çizelge 5.6. İki veri seti üzerinde SVM performans skorlarının karşılaştırılması	52

KISALTMALAR

AÇ	Açlık
GE	Geğirme
HU	Huzursuzluk
KA	Karın Ağrısı
k-NN	K-Nearest Neighbor
LPC	Linear Predictive Coding
LPCC	Linear Predictive Cepstral Coefficients
LR	Lojistic regresyon
MFCC	Mel Frequency Cepstral Coefficients
MLP	Multilayer Perceptron
RF	Random Forest
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
XGBoost	Extreme Gradient Boosting
YO	Yorgunluk

ÖZET

BEBEK AĞLAMALARININ MAKİNE ÖĞRENMESİ ALGORİTMALARIYLA SINIFLANDIRILMASI

Adem EKİNCİ

Düzce Üniversitesi
Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi
Danışman: Dr. Öğr. Üyesi Enver KÜÇÜKKÜLAHLI
Temmuz 2023, 62 sayfa

İnsanlar sürekli olarak birbirleriyle iletişim halindedirler ve bunu genellikle dil aracılığıyla gerçekleştirirler. Yeni doğan bir bebeğin bu beceriyi kazanana kadar kendini ifade etmek için en etkili yolu ağlamaktır. Bebek ağlamaları genellikle yetişkinler tarafından rahatsız edici ve anlamsız olarak algılanabilirken, aslında birçok bilgi içerebilirler. Bu bilgiler doğru yöntemler kullanılarak analiz edildiğinde bebeğin ağlama nedenini ortaya koyabilir. Bebeğin sağlığı ve mutluluğu için ağlama nedeninin anlaşılması çok önemlidir. Bu tez çalışmasında, bebek ağlaması sinyallerindeki bilgi ses işleme yöntemleri kullanılarak anlamlandırılmaya ve makine öğrenimi algoritmalarıyla sınıflandırılmaya çalışılmıştır. Bu doğrultuda beş ayrı sınıfa ayrılmış bebek ağlaması ses sinyalleri içeren bir veri seti kullanılmış, veri seti üzerinde verileri eşit parçalara bölme yöntemi uygulanarak her bölme işlemi için ayrı veri seti oluşturulmuştur. Daha sonra her veri seti üzerinde öznitelik çıkarma işlemleri uygulanmış ve sınıflandırma algoritmaları ile performans değerleri ölçülmüştür. Bu ölçüm sonuçları incelenmiş ve kullanılan veri artırma yöntemi ile oluşturulan veri setleri üzerinde daha yüksek performans değerleri elde edildiği görülmüştür. Kullanılan altı algoritma oluşturulan veri setleri ile modellenmiş ve en yüksek performans skoru Extreme Gradient Boosting (XGBoost) algoritması ile %99,51 olarak bulunmuştur.

Anahtar Sözcükler: Bebek ağlamaları, Ses İşleme, MFCC, Makine Öğrenmesi, XGBoost

ABSTRACT

CLASSIFICATION OF BABY CRIES USING MACHINE LEARNING ALGORITHMS

Adem EKİNCİ
Düzce University
Graduate School, Department of Computer Engineering
Master Thesis
Supervisor: Assist. Prof. Dr. Enver KÜÇÜKKÜLAHLI
July 2023, 62 pages

People are constantly in communication with each other, and they generally do this through language. Until a newborn baby acquires this skill, crying is the most effective way for them to express themselves. While baby cries may be perceived as disturbing and meaningless by adults, they actually contain a lot of information. When these pieces of information are analyzed using appropriate methods, the reason behind the baby's crying can be revealed. Understanding the cause of the baby's cries is crucial for their health and happiness. In this thesis study, an attempt has been made to interpret baby cries using signal processing techniques and classify them with machine learning algorithms. For this purpose, a dataset containing baby cry audio signals categorized into five classes was used. The dataset was divided into equal parts for each partitioning process, and a separate dataset was created for each part. Then, feature extraction processes were applied to each dataset, and performance values were measured using classification algorithms. These measurement results were examined, and it was observed that higher performance values were achieved on the datasets generated using the data augmentation method. Six algorithms were modeled with the generated datasets, and the highest performance score of 99.51% was obtained with the Extreme Gradient Boosting (XGBoost) algorithm.

Keywords: Baby cries, Sound processing, MFCC, Machine learning, XGBoost

1. GİRİŞ

Ağlamak, tüm bebeklerde görülen, doğuştan gelen bir tepkidir ve onlar için etkili bir iletişim biçimidir. Açlıklarını, huzursuzluklarını, uykusuzluklarını veya bir şeylerin yanlış gittiğini bu yolla ifade edebilirler. Bu nedenle bebeklerin ağlaması, bakımından sorumlu kişilerin (ebeveynler, bakıcılar, pediatristler, sağlık personelleri vb.) dikkat etmesi gereken en önemli işaretlerden biridir. İşaret ettikleri durum, doğru anlaşılır ve sorunları giderilirse daha mutlu olurlar. Bu da sağlıklı büyümelerine olumlu etki yapar [1]. Bebeğin bakımından sorumlu kişiler, tecrübe ve bilgilerini kullanarak işitsel duyu algılarına dayanarak bebeğin neden ağladığını doğru tahmin edebilirler, ancak her zaman tecrübeli ve bilgili olamayabilirler. Ayrıca bu tamamen subjektif bir değerlendirmedir [2]. Bu gibi önemli bir konuda, daha objektif ve hata payı çok az olan yöntemler kullanılmalıdır; bebeğin ağlayarak ifade etmek istediği ihtiyaçlarının karşılanmaması durumunda ağlamanın süresi ve şiddeti artabilir. Bu durum, bebeği birçok açıdan olumsuz etkileyebilir. Uzun süre ağlayan bebeklerde kortizol seviyesi artar ve bu durum beyin ve bilişsel işlevler üzerinde olumsuz etkilere neden olabilir [3]. Bu çalışmanın temel amacı bebek ağlamalarının sınıflandırılmasına katkıda bulunmak ve olumsuz etkilerini azaltmaktır. Bu amaç doğrultusunda bebek ağlamasının nedenini daha yüksek bir doğrulukla belirlemektir. Bebek ağlamasının bir ses sinyali olduğu düşünüldüğünde, ses işleme bu bağlamda daha etkili sonuçlar sağlayabilir.

Ses mütemediyen, zamanla değişen tek boyutlu bir sinyal olarak kabul edilir ve bu sinyal kulak kanalındaki hava basıncını temsil eder [4]. İnsan kulağı, 20Hz ile 20.000Hz arasındaki tüm sesleri algılayabilir, ancak bunların tamamını anlamlandıramaz. Bir bebek sesi, bebeğin durumuyla ilgili bilgiler içerebilir; ancak çoğu insan için bebek ağlaması anlamsızdır. Bebeğin ağlamasının altında yatan nedeni anlamak, bu ağlamayı bir ses sinyali olarak değerlendirmeyi gerektirir. Çünkü her ses sinyali, karakteristik yapılarıyla belirli özelliklere sahiptir. Frekans, zaman, genlik, ton, güç vb. gibi faktörler, seslerin ayrımını sağlar ve sesi daha iyi tanımlamamıza yardımcı olur. Ses işleme, bu ses sinyallerinin analiz edilmesi, özniteliklerinin çıkarılması ve anlamlandırılması için kullanılan bir disiplindir [5]. Öznitelik çıkartma, bir veri kümesindeki verilerin önemli özelliklerini ayırt etme sürecidir. Öznitelikler, ses verilerinin zaman, frekans ve diğer

özelliklerindeki farklılıklardan ortaya çıkar. Öznitelik çıkarma işlemi için Mel-frequency cepstral (MFCC), Linear Predictive Coding (LPC), Short-Time Fourier Transform (STFT) gibi farklı yöntemler kullanılabilir [6], [7]. Bu işlem sonrası ses içeren veri, makine öğrenmesi algoritmaları ile uyumlu hale gelir.

Son dönemde, hemen hemen tüm teknolojik gelişmelerde yapay zeka kavramı sıkça karşımıza çıkmaktadır. Bu bağlamda, makine öğrenmesi yapay zekanın önemli bir alanını oluşturur. Makine öğrenmesi, verileri veya deneyimleri analiz ederek öğrenme yeteneği kazanan sistemlerin geliştirilmesi ve ilerletilmesi bilimidir. Makine öğrenmesi algoritmaları, veri kümeleri üzerinde eğitilir ve bu verilerin niteliği, çeşitliliği, boyutu ve doğruluğu, algoritmaların performansını doğrudan etkiler. Bu nedenle, makine öğrenmesi için temel bir dayanak veri kümeleridir [8].

Veri kümelerinde yeterli veri olmaması, aşırı uyum (overfitting) veya modelin genelleştirme yeteneğinde düşüşe neden olabilir. Bu gibi durumların önlenmesi amacı ile veri artırma yöntemleri kullanılabilir. Veri artırma yöntemleri, hali hazırda elde bulunan veriler üzerinde oynama yaparak yeni verilerin oluşturulması sürecidir [8]. En bilinen veri artırma yöntemleri arasında; rastgele kesme (Random cropping), rastgele kaydırma (Random shifting), aynalama (Mirroring), döndürme (Rotation), zaman değiştirme (Time stretching) gibi yöntemler sayılabilir [9]. Bu çalışmada ise müzik parçalarının kısa bir kesitinden tanınması için uygulanan verileri eşit parçalara bölme yöntemi kullanılmıştır [10].

Bebeğin ağlaması temelde bir ses sinyalidir. Diğer ses sinyalleri gibi belli bir frekansa, tona ve şiddete sahiptir. Bu ses sinyalleri üzerinde ses işlemede kullanılan her işlem uygulanabilir. Öznitelik çıkartma da bu işlemlerden biridir. Öznitelikleri çıkartılan bebek ağlama sinyalleri makine öğrenmesiyle işlenebilir. Böylece yeterli veri ve doğru yöntemle eğitilen bir makine öğrenmesi modeli, bebeklerin neden ağladığını objektif bir değerlendirme ve yüksek doğrulukla tespit edebilir.

Makine öğrenmesi bilgisayar sistemlerinin veriye dayalı olarak öğrenme yeteneğiyle görevleri gerçekleştirmesini sağlayan bir alandır. Bu sistemler, verilerdeki desenleri ve ilişkileri tanıyarak gelecekteki verileri tahmin etmek, sınıflandırmak veya kararlar vermek gibi görevleri gerçekleştirebilirler. Denetimli, denetimsiz ve pekiştirmeli öğrenme olmak üzere farklı yaklaşımlar kullanılır.

2. LİTERATÜR ÖZETİ

Literatürde bebek ağlama seslerinin sınıflandırılması ile ilgili birçok çalışma bulunmaktadır. Özseven'in çalışmasında, bebek ağlamalarının sınıflandırılması için elle girilmiş özellikler ve spektral görüntülerin kullanılması incelenmiştir. Deneyler, bir boyutlu Convolutional Neural Network (CNN) modeli, aktarım öğrenme ve elle girilmiş özelliklerin sınıflandırma performansını değerlendirmiştir. Sonuçlar, elle girilmiş özelliklerin ve spektrogram/skalogram görüntülerinin yüksek başarı sağladığını göstermiştir. Ayrıca, 1D CNN modelinin diğer yöntemlere göre daha düşük başarı gösterdiği belirlenmiştir. Sonuçlar, literatürde yaygın olarak kullanılan iki veri seti üzerinde karşılaştırılmış %97,60 ve %95,20 doğruluk oranı elde edilmiştir. Bu sonuçlar hem sinyal işleme yöntemlerinin hem de spektrogram ve skalogram görüntülerinin bebek ağlama sınıflandırma çalışmalarında başarıyla kullanılabileceğini göstermektedir [1].

Rezaee ve arkadaşlarının yaptığı çalışmada, prematüre ve zamanında doğmuş bebeklerin ağlama seslerinin analiz edilmesi için bir yöntem önerilmiştir. Çalışmada donate-a-cry veri seti kullanılmış. Fraktal tanımlayıcı kullanarak özellik çıkarma ve Deep Support Vector Machine (deepSVM) ile sınıflandırma yapılmıştır. Farklı özelliklere dayalı beş tür ağlama algılaması sınıflandırmasına göre, tüm ağlama algılamalarının %98,34'ü tanınmıştır. Elde edilen sonuçlar, önerilen yöntemin bebek ağlama dilini analiz etmede diğer yöntemlere kıyasla daha yüksek bir doğruluk sağladığını göstermektedir [2].

Khan ve Neware, yenidoğan ve bebeklerde ağlama analizi ve ağrı değerlendirmesi için otomatik bir sistem geliştirmeyi amaçlayan bir çalışma sunmuşlardır. Bu çalışmada, bebeklerin ağlama sesleri, yüz ifadeleri, vücut hareketleri ve hayati işaretleri gibi çeşitli ağrı göstergeleri incelenmiş ve bunların birleştirilmesiyle oluşan bir çoklu ortam ağrı değerlendirme sistemi önerilmiştir. Donate-a-cry veri seti kullanılan çalışmada ağlama seslerinin seviyesini tahmin etmede %88 civarında bir doğruluk elde edildiği belirtilmiştir. Ayrıca, ağlama sesleri ile yüz ifadesi, vücut hareketi ve hayati işaretleri birleştirilerek bebeklerin duygusal durumlarını ağrısız veya şiddetli ağrılı olarak sınıflandırmada %96,6 doğruluk elde edildiği rapor edilmiştir. Bu sonuçlar, bebeklerde ağrı değerlendirmesi için çoklu ortam göstergelerinin entegrasyonu ile otomatik bir

sistemin geliştirilmesinin mümkün olduğunu göstermektedir [3].

Ting ve arkadaşlarının yaptığı çalışma, bebek ağlama seslerini kullanarak asfiksi (boğulma) durumunu belirlemeye çalışan bir araştırmadır. Çalışmada, ağlama seslerinden hem sesin yüksekliği ve tonu gibi akustik özellikler hem de sesin ritmi ve vurgusu gibi prosodik özellikler elde edilmiştir. Akustik özellikler, MFCC ve Linear Predictive Cepstral Coefficients (LPCC) ile elde edilmiştir. Prosodik özellikler ise temel frekans (F0), enerji, formant frekansları ve spektral eğim gibi parametreler ile hesaplanmıştır. Bu özellikler bir araya getirilerek bir matris oluşturulmuştur. Bu matris, derin öğrenme modelleri ile işlenerek asfiksili ve normal bebek ağlama sesleri arasında ayırım yapılabilmektedir. Makalenin bulgularına göre, bu yöntem, asfiksili bebek ağlama seslerini yüksek doğrulukla tanımlayabilmektedir [4].

Ashwini ve arkadaşlarının yaptığı çalışmada, ses sinyalleri spektrogram görüntüsüne dönüştürülerek ve AlexNet adlı bir derin evrişimli sinir ağı kullanılarak bebek ağlamaları analiz edilmiştir. Ağ, son uçtan son uca öğrenme süreçleri gerçekleştirerek, ses sinyali analizindeki karmaşıklığı azaltıyor ve optimizasyon tekniği kullanarak performansı artırıyor. çalışmada, ağın etkinliğini artırmak için SGDM (Stokastik Gradyan İnişi ile Momentum) adlı bir algoritma kullanılmıştır. Çalışma, bu yöntemin açlık, uykulu, ağrılı ve rahatsız gibi farklı ağlama türlerini %95'e varan bir doğrulukla ayırt edebildiğini ortaya koymuştur [5].

Chang ve arkadaşlarının yaptığı çalışmada, yeni doğan bebeklerin ağlamalarını açlık, uyku ve rahatsızlık olmak üzere üç grupta sınıflandırmak için yeni bir yöntem önerilmiştir. Ağlama çerçevelerinden çıkarılan akustik özellikler ve Random Forest (RF) ile yapılan değişken seçimi kullanılarak ağlamalar analiz edilmiştir. Ardından, XGBoost destekli grouped-support-vector network (XGB-GSVN) yöntemiyle bebek ağlamaları sınıflandırılmıştır. Yapılan deneyler, önerilen yöntemin neonatal ağlamaları etkili bir şekilde üç farklı gruba sınıflandırabildiğini göstermektedir. En iyi deneysel sonuçlarda, çoğu senaryoda yaklaşık %91,00 ortalama doğruluk elde edilmiştir, bu da önerilen yöntemin neonatal ağlamaların sınıflandırılmasında büyük potansiyele sahip olduğunu göstermektedir. Ayrıca, duygusal ağlamaların tanımlanmasında önerilen yöntem hızlı bir tanıma oranına sahiptir ve sadece 27 saniye sürede sonuç verebilmektedir [6].

Mohite ve Jadhav çalışmalarında, sağır ebeveynler için bebek ağlamasını tanıyabilen ve uygun uyarılar gönderen bir sistem tasarlanmıştır. Bu sistemde, bebek ağlamasının sesi

bir mikrofonla alınmış ve Raspberry Pi işlemcisine gönderilmiştir. İşlemci, sesi Yapay Sinir Ağı (Artificial Neural Network – ANN) ve MFCC yöntemleriyle işleyerek ağlama türünü belirlemiştir. Eğer ağlama tespit edilirse, GSM Modem aracılığıyla ebeveynlerin cep telefonuna önceden tanımlanmış bir mesaj gönderilmiştir. Çalışmada Donate-a-cry veri seti gerçek bebek sesleri için ve Environmental-Sound Classification (ESC50) veri seti ise negatif sesler için kullanılmıştır. MFCC ve ANN kullanılarak alınan performans değeri %85'tir. Çalışmada tasarlanan sistem ile duyma engelli ebeveynlerin bebeklerinin durumlarını takip etmeleri amaçlanmıştır [7].

Ashwini K ve arkadaşlarının yaptığı çalışmada, yeni doğan bebeklerin ağlama sinyallerini ağrı, açlık ve uykusuzluk olarak sınıflandırmak için derin öğrenme ve makine öğrenimi tekniklerinin birleştirilmesi önerilmiştir. Ağlama ses sinyalleri spektrogram görüntüsüne dönüştürülmüş ve CNN ile öznelik çıkarımı yapılmıştır. Elde edilen öznelikler SupSVM ile sınıflandırılmış ve %88,89 performans değeri elde edilmiştir. Bu yöntem, orta sayıda veri örneğiyle bile yüksek doğruluk sağlamıştır. Bu çalışma, yeni doğan bebeklerin ağlama nedenlerini anlamak için etkili bir yöntem sunmaktadır [8].

Kulkarni ve arkadaşlarının yaptığı çalışmada, çocuk ağlamalarının nedenlerini belirlemek için konuşma ve işitsel işleme yoluyla çıkarılan farklı özellikleri kullanarak sınıflandırma yapılmıştır. Çalışmada, Lojistik Regresyon, SVM, (k-Nearest Neighbor) k-NN ve Rastgele Orman gibi farklı makine öğrenimi modellerinin performansları karşılaştırılmış, en iyi sınıflandırma sonucunun Rastgele Orman modeli ile %93,2 doğruluk oranına ulaşılmıştır. MFCC, spektrogram, temel frekans ve ses şiddeti gibi özelliklerin birleştirilmesinin etkili olduğu ortaya koyulmuştur. Ayrıca çalışmada, çocuk ağlamalarının sınıflandırılması için yeni yöntemler ve veri setleri önerilmiştir [9].

Sutanto ve arkadaşlarının yaptığı çalışmada, bebek kuvözünün gerçek bebek durumuna göre tepki vermesini sağlamak için Internet of Things (IoT) ve Akıllı Cihaz teknolojisinin kullanılma olasılığı araştırılmıştır. Bu çalışmada, Dunstan Baby Language ve Donate a cry veri setleri kullanılarak bebek ağlamalarının sınıflandırılması için bir yöntem sunulmuştur. Yaklaşım, WAV dosyalarından sesin gücünün analiziyle başlamış ve makine öğrenmesi için özelliklere sahip 2D desene ilerlemiştir. Bu çalışmadan, yaklaşık %85 doğruluk elde edilmiştir. Sensörlerle birlikte, bu önerilen yapılandırmanın kuvöz inovasyonu için yararlı olacağı belirtilmiştir [10].

Ji ve arkadaşlarının yazdığı makalede, bebek ağlama sinyali analizi ve sınıflandırma

işlemlerini kapsayan güncel çalışmalar incelenmiştir. Veri toplama, çapraz alan sinyal işleme teknikleri ve makine öğrenmesi sınıflandırma yöntemleri gibi yönlerden geniş bir literatür taraması yapılmıştır. Ön işleme yaklaşımları tanıtılmış ve MFCC, spektrogram ve temel frekans gibi çeşitli özellikler açıklanmıştır. Geleneksel makine öğrenmesi sınıflandırıcıları olan k-Nearest Neighbor (k-NN), SVM ve Gaussian Mixture Model (GMM)'nin yanı sıra, CNN ve (Recurrent Neural Network) RNN gibi yeni geliştirilen sinir ağı mimarileri de bebek ağlama araştırmalarında uyguladığı gözlemlenmiştir. Tipik veri tabanlarıyla patolojik ağlama tanımlama, ağlama nedeni sınıflandırma ve ağlama sesi tespiti üzerine bazı önemli deneysel sonuçlar incelenmiştir. Bu çalışma, yapıldığı yıla kadar olan bebek ağlamasıyla ilgili tüm alakalı alanlardaki önceki araştırmaları sistematik olarak incelemekte ve bebek ağlama sinyali analizi ve sınıflandırmasındaki güncel çalışmalara bir bakış sunmaktadır. Ayrıca, bebek ağlama sinyallerini daha iyi anlamak, yorumlamak ve işlemek için veri işleme, özellik çıkarma ve sinir ağı sınıflandırma alanlarında gelecekte yapılabilecek çalışmalara değinilmiştir [11].

Widhyanti ve Juniati'nin yaptığı çalışmada, Higuchi Fraktal boyutu kullanılarak bebek ağlama seslerinin sınıflandırılması amaçlanmıştır. Toplamda 80 veriden oluşan 4 sınıflı bir veri kümesi kullanılmıştır. Fraktal boyut değerinin her ses sinyali için hesaplanmasından sonra, veri gruplama işlemi gerçekleştirilmiştir. Veri dağıtımı için k-katlı çapraz doğrulama yöntemi kullanılmış, sınıflandırma işlemi ise k-NN ve SVM yöntemleri ile yapılmıştır. k-NN ile %78,75 ve SVM ile %80 doğruluk oranı bulunmuştur [12].

Jiang ve arkadaşlarının yaptığı çalışmada, işitsel bir modelden ağlama seslerinin özellikleri belirlenmiştir. Veri seti olarak donate-a-cry kullanılan çalışmada sınıflandırma içinde "Gaussian Mixture Model-Universal Background Model" adlı bir yöntem kullanılmıştır. Kullanılan bu yöntem sayesinde kanal dengesizliği ve bozulmuş sinyale karşı dayanıklılık sağlanmıştır. Mevcut bir özellik olan High Order Spectral Feature (HOS) ile karşılaştırıldığında, tanıma doğruluğu temiz ve bozulmuş sinyal altında sırasıyla %14,6 ve %13,7 oranında artışı gözlenmiştir. Temiz sinyallerde %93,71 f1, bozulmuş sinyallerde %82,77 f1 puanı elde edilmiştir [13].

Bashiri ve Hosseinkhani'nin yaptığı çalışmada, bebeklerin çevreyle iletişim kurmak için kullandıkları tek yol olan ağlamanın nedenlerini belirlemek amacıyla yapay sinir ağı ve genetik algoritma kullanarak sınıflandırma yapılmıştır. Çalışmada, ağlama sinyallerinin analizi ve özellik çıkarımı için LPC ve MFCC gibi yöntemler kullanılmıştır. Ayrıca, en

iyi özellikleri seçmek amacıyla genetik algoritma ve sınıflandırmada yapay sinir ağı kullanılmıştır. Çalışmada, önerilen yöntemin diğer yöntemlere kıyasla daha üstün olduğu vurgulanmış ve yeni doğanların ağlamalarını %93,2 doğruluk oranıyla sınıflandırmıştır [14].

Maghfira ve arkadaşlarının yaptığı çalışmada, bebek ağlamalarının nedenlerini belirlemek için derin öğrenme yöntemleri kullanılmıştır. Çalışmada, ağlama sinyalleri spektrograma dönüştürülerek görüntü olarak işlenmiştir. Ağlama sinyallerinin özellikleri, CNN ile çıkarılmış ve tekrarlayan sinir ağı (RNN) ile sınıflandırılmıştır. Çalışma, CNN-RNN yönteminin, geleneksel makine öğrenimi yöntemlerine göre daha yüksek doğruluk sağladığını ortaya koymuştur [15].

Le ve arkadaşlarının yaptığı çalışmada, bebeklerin ağlamalarını analiz ederek farklı türlerini sınıflandırmak için derin öğrenme, makine öğrenimi ve karar birleştirme yöntemleri kullanılmıştır. Çalışmada, bebeklerin ağlamalarının sağlık durumları ve ihtiyaçları hakkında değerli bilgiler taşıdığı ve erken teşhis edilmesi halinde bazı hastalıkların önüne geçilebileceği belirtilmiştir. Ağlama ses dosyalarının spektrogram görüntüleri kullanılarak işlenmiş ve önceden eğitilmiş bir CNN olan ResNet50 ile özellik çıkarımı yapılmıştır. Ayrıca, Support Vector Machine (SVM) ile sınıflandırma yapılmış ve CNN ve SVM'in tahminlerini birleştirmek için karar birleştirme yöntemi olan oylama kullanılmıştır. Baby Chillanto veri tabanından alınan ağlama örnekleri üzerinde modeller eğitilmiş ve 10 katlı Çapraz Doğrulama (Cross Validation) ile %90 üzerinde doğruluk elde edilmiştir [16].

Dewi ve arkadaşlarının yaptığı çalışmada, bebeklerin ağlamalarını analiz ederek duygu durumlarını belirlemek için LFCC ve k-NN yöntemlerini kullanılmıştır. LFCC yönteminde, ön vurgulama, filtre bankası sayısı ve kepsral sayısı değişikliklerinin analizi yapılmıştır. Uygulanan filtre bankası değerinin uygulanan kepsral değerinden büyük olması gerektiği vurgulanmıştır. Kepsral değerleri daha iyi doğruluk elde etmek için ayarlanmıştır. Sistem 8 kepsral değeri ve 3 en yakın komşu değeri kullandığında en yüksek doğruluk yüzdesi olan %90 elde edilmiştir ve tüm kurallar test sonuçlarına göre en iyi değer olarak kabul edilmiştir [17].

Tuduce ve arkadaşlarının yaptığı çalışma, bebek ağlamalarının farklı ihtiyaçları nasıl yansıttığını otomatik olarak tanımlayabilecek bir sistem geliştirmeyi amaçlamıştır. Çalışmada, bebek ağlamalarından çeşitli paralinguistik özellikler çıkarılmış ve bu

özellikler kullanılarak çeşitli kural tabanlı veya istatistiksel sınıflandırıcılar eğitilmiştir. Çalışmanın sonuçları, değerlendirme veri kümesinde %70'e kadar doğruluk elde edildiğini göstermiştir. Çalışma, bebek ağlamalarının anlaşılmasına katkıda bulunmayı ve ebeveynlere yardımcı olmayı hedeflemiştir [18].

Franti ve arkadaşlarının çalışması literatürdeki diğer bebek ağlamasının sınıflandırılması çalışmalarından oldukça farklıdır. Bu çalışma, 3 aydan küçük bebeklerin ihtiyaçlarını ağlamadan önce iletmek için kullandıkları iddia edilen beş “kelime” veya ses biriminden oluşan "Dunstan bebek dili"nin tanınması için CNN uygulamasını sunmuştur. Çalışmada, CNN mimarisi, yazarların ses temelli duygu tespiti için başarıyla kullandıkları bir mimariden türetilmiştir. Sinir ağının girdisi, bebek seslerinin ses kayıtlarından elde edilen spektrogramdır ve iki boyutlu bir görüntü olarak işlenmiştir. Mimarisi, 250 küçük süreli kayıt kümesi için eğitilmiş ve diğer 65 kayıt için %89 tanıma oranı ile test edilmiştir. Tüm ses dosyalarının uzunluğu 1 saniyeden azdır; kayıtlar sertifikalı Dunstan dil kayıtlarından çıkarılmıştır. Çalışmanın en önemli orijinal katkısı, gerçek "bebek kelimeleri"nin tanınmasıdır (daha önce yapıldığı gibi bebek ağlaması değil). Bu mimarisi, bebek dilinin evrensel olduğu hipotezinin doğrulanması için etkili bir araç sunmaktadır. Bu hipoteze göre, bebeklerin dili kültür, aile vb. bağımlı değildir [19].

Orlandi ve arkadaşlarının çalışması, tam zamanlı ve erken doğan bebeklerin ağlamalarını otomatik olarak sınıflandırmak için desen tanıma ve veri madenciliği tekniklerini kullanmayı amaçlamıştır. Çalışmada, 28 tam zamanlı ve 10 erken doğan bebeğin ağlama kayıtlarından 3000'den fazla ağlama biriminden 22 akustik parametre hesaplanmıştır. Özellik çıkarımı, Firenze Üniversitesi'nde geliştirilen BioVoice adlı özel bir yazılım aracı ile yapılmıştır. Sınıflandırma ve desen tanıma, en iyi öznelikleri seçmek için genetik algoritmalar temel alınmıştır. Eğitim, dört sınıflandırıcı ve üç farklı test seçeneği karşılaştırılarak gerçekleştirilmiştir. Sonuçlar, değerlendirme veri kümesinde %87'ye kadar doğruluk elde edildiğini göstermiştir. Bu 10 ağlama özelliği, klinik uzmanlara erken doğum nedeniyle nörolojik gelişimde olası gecikme veya bozuklukların tanısı ve takibinde önemli ek bilgiler sağlayacağı öne sürülmüştür [20].

3. MATERYAL

3.1. VERİ SETİ

Bu çalışmada github platformu üzerinde yayınlanan Donate-a-cry [21] açık kaynak veri seti kullanılmıştır. Donate-a-cry, temelde ebeveynlerin kendi bebeklerinin ağlama seslerini ve gerekli bilgileri (ağlama nedeni, kaç aylık olduğu, cinsiyeti gibi) bağışlaması ile oluşmuştur. Bunun için Donate-a-cry adlı iOS ve Android cihazlarda çalışan bir uygulama kullanılmıştır (Uygulama 2015 yılından itibaren aktif değildir). Bu uygulama sayesinde toplanan 2000'e yakın ses kaydı Gaber Veres tarafından 2015 yılında github platformuna yüklenmiştir. Bu veri seti üzerinde 2019 yılında bir güncelleştirme yapılmıştır. Bu güncellemede ses kayıtları gürültülerden arındırılmış, sessizlik içeren kısımları silinmiş kalan veriler 5 ayrı sınıfa (ağlama nedenlerine göre) ayrılmıştır. Ses dosyaları 1 ila 22 aylık bebek ağlamalarını içermektedir. Dosyaların frekans değeri 1800, bit hızı değeri ise 128 kbps'dir. Bu çalışmada veri setinin güncellenmiş versiyonu kullanılmıştır. Verilerin sınıflara dağılımı Çizelge 3.1'de yer almaktadır.

Çizelge 3.1. Donate-a-cry Veri Setinin Sınıflara Dağılımı

Ağlama Nedeni	Kısaltma	Kayıt Sayısı
Karın Ağrısı	KA	16
Geğirme	GE	8
Huzursuzluk	HU	27
Açlık	AÇ	382
Yorgunluk	YO	24
Toplam		457

3.2. SES İŞLEME VE ÖZNETELİK ÇIKARMA

Öznitelik çıkarma, ham verilerden makine öğrenmesi algoritmalarının işleyebileceği sayısal öznitelikler elde etme sürecidir. Bu süreçte orijinal veri kümesindeki bilgiler

korunur. Öznitelik çıkarma, başlangıçtaki özniteliklerden bir alt küme belirleme yöntemi olan öznitelik seçiminden ayrıdır. Öznitelik çıkarma, verinin türüne ve kullanım alanına göre elle veya otomatik olarak gerçekleştirilebilir. Öznitelik çıkarımı ayrıca önceki bir ağ tarafından öğrenilen temsilleri kullanarak yeni örneklerden ilgi çekici öznitelikler çıkarmayı içerir. Bu öznitelikler daha sonra sıfırdan eğitilen yeni bir sınıflandırıcıya aktarılır [22].

Öznitelik çıkarma yöntemlerinden bazıları şunlardır:

Principal Component Analysis (PCA): Veriyi en çok değişkenliğin olduğu yönleri bulup daha az boyutlu bir uzama yansıtan bir boyut azaltma yöntemidir.

Independent Component Analysis (ICA): Çok değişkenli bir sinyali istatistiksel olarak bağımsız olan bileşenlere ayıran bir yöntemdir.

Latent Semantic Analysis (LSA): Bir metin kümesinin gizli konularını ortaya çıkarmak için kelime sayısı matrisine PCA uygulayan bir yöntemdir.

Dict Vectorizer: Öznitelik dizilerini Python sözlükleri listesi şeklinde ifade eden bir sınıf, bunları scikit-learn tahminleyicilerinin kullandığı NumPy/SciPy biçimine dönüştürür.

Görüntü işleme algoritmaları: Dijitalleştirilmiş bir görüntü veya video akışındaki istenilen bölge veya şekilleri (öznitelikleri) belirleyip ayıklayan bir algoritma setidir. Örneğin kenarlar, köşeler, bloklar vb.

Dijital ses işleme, ses sinyallerini bilgisayar tabanlı yöntemlerle analiz etme, işleme ve anlama sürecidir. Ses sinyallerini sayısal verilere çevirme ve bu veriler üzerinde matematiksel manipülasyon işlemlerini de içerir.

Ses işleme için öznitelik çıkarma, ses veri setlerinden anlamlı bilgileri çıkararak verilerin daha işlenebilir bir formda temsil edilmesini sağlayan bir işlemdir. Bu öznitelikler sesin spektral, zamansal veya enerji gibi özelliklerini ifade edebilir ve çeşitli ses işleme uygulamalarında kullanılır. Türlerine göre bazı ses işleme öznitelik çıkarma işlemleri aşağıdaki gibi verilebilir [23].

Zaman Alanı (Time Domain) Öznitelikleri:

- Ortalama ve Varyans: Ses sinyalinin zaman içindeki ortalama ve varyans düzeyini temsil eder.
- Tepe Değerleri: Ses sinyalinin zaman alanındaki tepe noktalarını ifade eder.
- Sıfır Geçiş Sayısı (Zero-Crossing Rate): Ses sinyalinin zaman içinde kaç kez sıfır noktasını geçtiğini belirler.

Frekans Alanı Öznitelikleri:

- Güç Spektrumu: Ses sinyalinin frekans bileşenlerini temsil eden güç spektrumu.
- MFCC: İnsan işitme sistemine benzer şekilde ses sinyalini frekans skalasında temsil eden özniteliklerdir.
- Spektral Bant Enerjileri: Belirli frekans bantlarındaki enerji seviyelerini temsil eder.

İstatistiksel Öznitelikler:

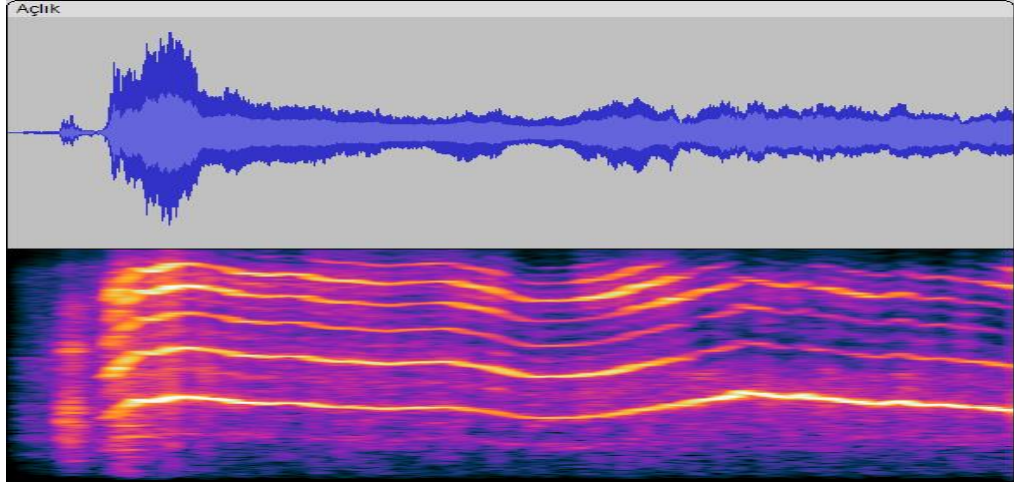
- Ortalama ve Standart Sapma: Ses sinyalinin istatistiksel merkezi eğilimini ve dağılımını temsil eder.
- Çarpıklık ve Basıklık: Ses sinyalinin dağılımının simetrisini ve düzlüğünü temsil eder.

Zaman-Frekans Öznitelikleri:

- Spektrogram: Ses sinyalinin frekans ve zaman bilgilerini temsil eden bir grafiksel gösterimdir.
- Dağılım Katsayıları: Ses sinyalinin zaman-frekans düzlemindeki enerji dağılımını temsil eder.

Bu tez çalışmasında öznitelik çıkartma işlemi için MFCC yöntemi kullanılmıştır.

MFCC, bir ses sinyalinin Mel ölçeğindeki algısal değişiklikleri temsil eden kısa vadeli güç spektrumunun bir gösterimidir. Örnek bir ses sinyali ve bu sinyalinden el edilmiş mel Spektrogram görüntüsü Şekil 3.1’de verilmiştir. MFCC yöntemi, ses sinyalini kısa zaman çerçevelerine böler ve her çerçeve için bir güç spektrumu analizi yapar. Bu analizde, sesin farklı frekans bantlarındaki enerjisi hesaplanır ve insan kulağının frekans hassasiyetine benzer bir şekilde ağırlıklı olarak değerlendirilir. Ardından, frekans ölçeğini dönüştürmek için logaritmik işleme ve Discrete Cosine Transform (DCT) uygulanır, bu da MFCC katsayılarının çıkarılmasını sağlar. Bu katsayılar ses sinyalinin farklı özelliklerini temsil eder ve konuşma tanıma, konuşmacı tanıma ve müzik sınıflandırma gibi çeşitli uygulamalarda kullanılabilir.



Şekil 3.1. Açlık sınıfına ait bir verinin sinyal ve mel-spectrogram görüntüsü

MFCC vektörlerinin oluşturulması şu adımlardan oluşur [24]:

- Önce ses sinyali ön vurgulama filtresinden geçirilir. Bu filtre yüksek frekanslardaki enerjiyi artırır ve ses tanıma performansını iyileştirir.

$$y[n] = x[n] - \alpha x[n - 1]$$

Burada $y[n]$ filtrelenmiş sinyal, $x[n]$ giriş sinyali ve α ön vurgulama katsayısıdır.

- Sonra ses sinyali kısa zaman aralıklarına bölünür. Bu aralıklara çerçeve denir. Her çerçeve için tipik olarak 25 ms uzunluk ve 10 ms kayma kullanılır. Çerçeveler arasında ani düşüşleri önlemek için Hamming penceresi uygulanır.

$$w[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N - 1}\right)$$

Burada $w[n]$ pencere fonksiyonu, n çerçeve indeksi ve N çerçeve uzunluğudur.

- Daha sonra her çerçeve için spektrum hesaplanır. Spektrum, ses sinyalinin frekans bileşenlerini gösterir. Spektrum hesaplamak için Fast Fourier Transform (FFT) kullanılır.

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

Burada $X[k]$ spektrum, $x[n]$ zaman sinyali, N çerçeve uzunluğu ve k frekans indeksidir.

- Sonra spektrum Mel-filtre bankası ile ağırlıklandırılır. Mel-filtre bankası, insan kulağının algıladığı frekans skalasını taklit eden bir dizi üçgen filtreden oluşur.

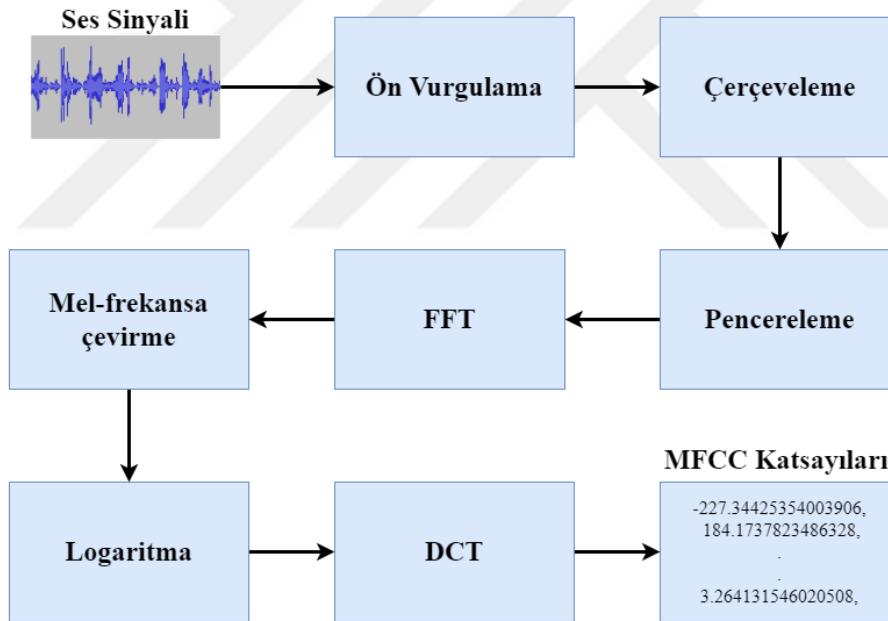
$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \frac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \leq k < f(m) \\ \frac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \leq k < f(m+1) \\ 0 & k \geq f(m+1) \end{cases}$$

Burada $H_m(k)$ m'inci filtrenin k'inci frekanstaki değeri, $f(m)$ m'inci filtrenin merkez frekansıdır.

-Son olarak, her filtreden geçen spektrumun logaritması alınır ve Discrete Cosine Transform (DCT) uygulanarak MFCC katsayıları elde edilir.

$$E = \sum_{n=0}^{N-1} x^2[n]$$

Burada E enerji ve $x[n]$ zaman sinylidir. MFCC'nin akış şeması Şekil 3.2'de verilmiştir.



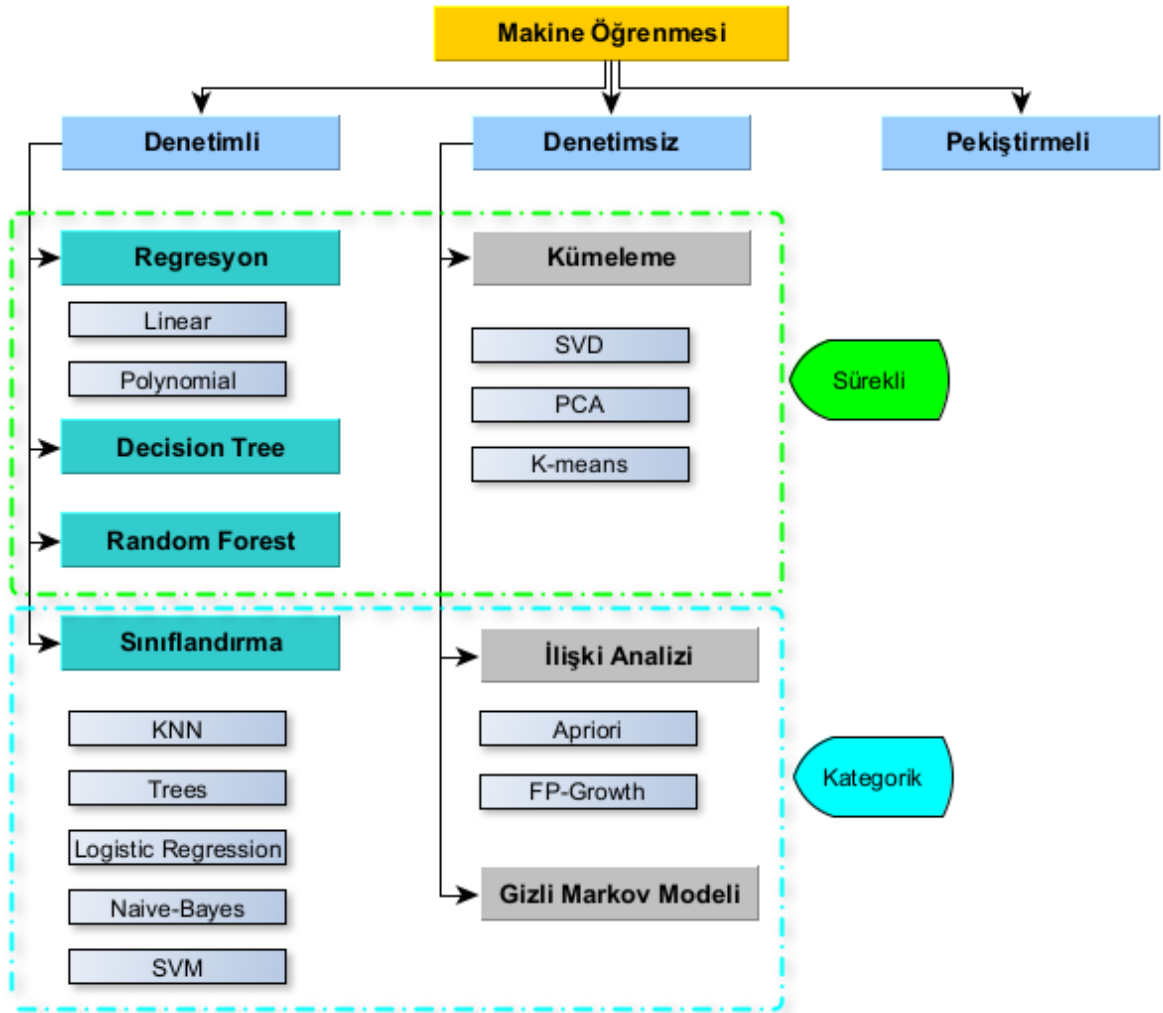
Şekil 3.2. MFCC akış şeması [25]

3.3. MAKİNE ÖĞRENMESİ ALGORİTMALARI

Makine öğrenimi, bilgisayar bilimindeki bir alt disiplindir ve bazı olayların bir dizi örneğine dayanan algoritmaların oluşturulmasıyla ilgilenir. Bu örnekler, doğadan kaynaklanabilir, insanlar tarafından hazırlanabilir veya başka bir algoritma tarafından üretilir. Makine öğrenimi ayrıca, bir veri kümesinin toplanması ve bu veri kümesine

dayalı istatistiksel bir modelin algoritmik olarak oluşturulmasıyla pratik bir sorunu çözme süreci olarak tanımlanabilir. Bu istatistiksel model, pratik sorunu çözmek için tasarlanır veya hazırlanır [26].

Yapay zekanın en çok başvurulan alt dallarından biri olan makine öğrenmesi, bünyesinde farklı amaçlar için kullanılan birçok algoritma barındırır. Makine öğrenmesi alanında kullanılan metotları içeren makine öğrenmesi metodolojisi Şekil 3.3'de yer almaktadır. Bu çalışmada makine öğrenmesinin önerilen yöntem ile altı ayrı sınıflandırma algoritması modellenmiş ve sınıflandırma performansları incelenmiştir.



Şekil 3.3. Makine öğrenmesi metodolojisi

3.3.1. Random Forest (RF)

Random forest algoritması, gözetimli öğrenme alanında kullanılan güçlü ve esnek bir makine öğrenmesi tekniğidir. Bu algoritma, birden çok karar ağacını bir araya getirerek

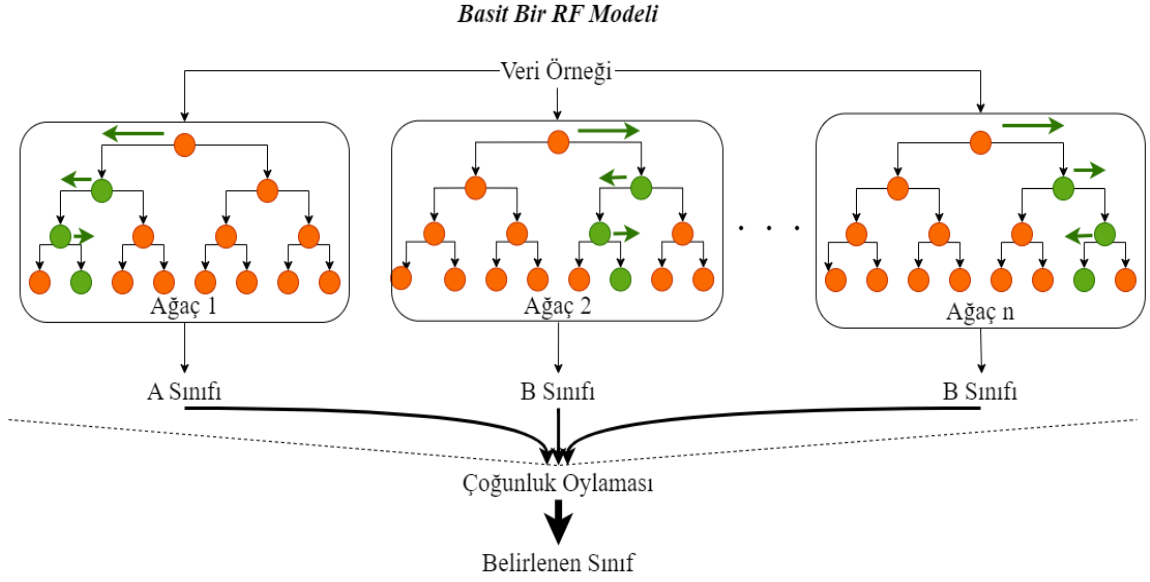
sınıflandırma veya regresyon problemlerini çözebilir. Ensemble learning kavramı üzerine kurulmuştur, bu da modelin performansını yükseltmek ve karmaşık bir problemi çözmek için çok sayıda sınıflandırıcının bir araya getirilmesi işlemidir.

Algoritma, eğitim zamanında çoklu karar ağaçları inşa ederek ve bireysel ağaçların sınıflarının en sık olanı (sınıflandırma) veya ortalama tahmini (regresyon) olan sınıfı belirleyerek işler. Karar ağaçları, özniteliklerin ve verilerin rastgele bir alt kümesi kullanılarak ve yerine koyma yöntemiyle oluşturulur (önyükleme). Bu, ağaçlar arasında rastgelelik ve çeşitlilik sağlar, bu da tahminlerin korelasyonunu ve varyansını düşürür [27].

RF, her bir ağacın eğitim kümesi S ve bir dağılımdan bağımsız ve ayrık olarak örneklenen ek bir rastgele vektör θ üzerinde algoritma A uygulanarak oluşturulan bir karar ağacı koleksiyonundan oluşan bir sınıflandırıcıdır. Rastgele ormanın tahmini, bireysel ağaçların tahminlerinin çoğunluk oyuyla elde edilir.

Belirli bir rastgele ormanı belirtmek için, algoritma A 'yı ve θ üzerindeki dağılımı tanımlamamız gerekmektedir. Bunu yapmanın birçok yolu vardır. θ 'yı oluşturulması ise aşağıdaki gibi olacaktır.

Öncelikle, S 'den yerine koyma yöntemiyle rastgele bir alt örneklem alınır; yani, S üzerindeki uniform dağılımı kullanarak m boyutunda yeni bir eğitim kümesi S' örneklenir. İkinci olarak, I_1, I_2, \dots şeklinde bir dizi oluşturulur, burada her I_t , $[d]$ kümesinin k boyutunda bir alt kümesidir ve $[d]$ 'den uniform dağılımla rastgele öğeler seçerek oluşturulur. Tüm bu rastgele değişkenler θ vektörünü oluşturur. Sonra, algoritma A , S' örneği üzerinde bir karar ağacı oluşturur (örneğin, ID3 algoritmasını kullanarak) ve algoritmanın her bölünme aşamasında, kazancı en çok arttıran özelliği I_t kümesinden seçmesine izin verir. Sezgisel olarak, eğer k küçükse, bu kısıtlama overfitting'i önleyebilir [28]. Bu şekilde oluşturulan basit bir RF modeli Şekil 3.4'de verilmiştir.

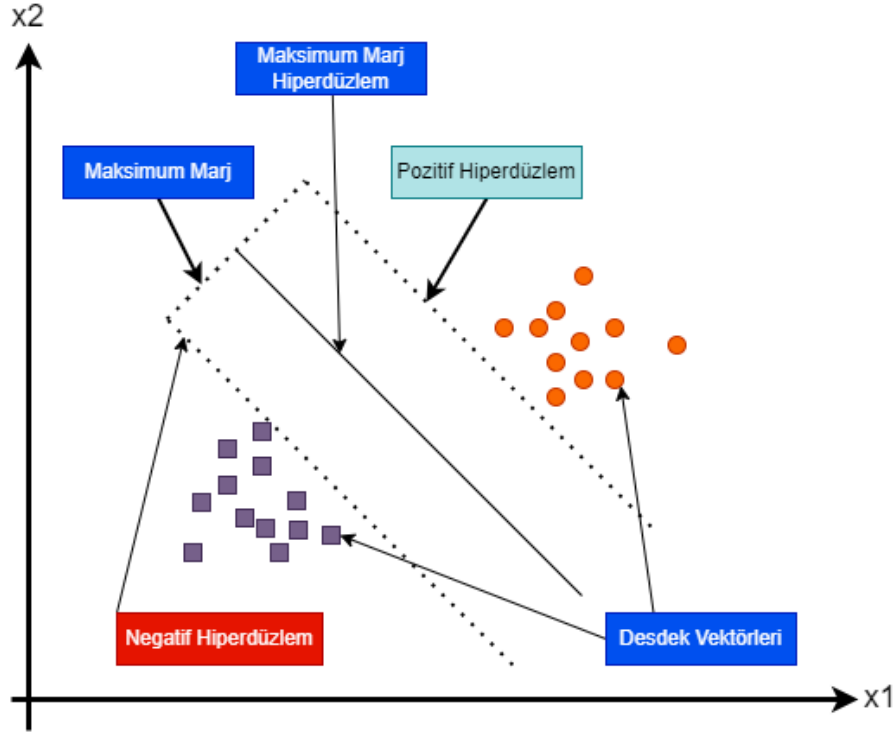


Şekil 3.4. Basit bir RF modeli [29]

3.3.2. Destek Vektör Makinesi (Support Vector Machine - SVM)

Makine öğrenmesinde, destek vektör makineleri sınıflandırma ve regresyon analizi için verileri analiz eden ilişkili öğrenme algoritmalarına sahip denetimli öğrenme modelleridir. Özellikle sınıflandırma problemleri için günümüzde kullanılan en popüler algoritmalarından biridir. SVM algoritmasının amacı, n-boyutlu uzayı sınıflara bölebilecek en iyi çizgi veya karar sınırını oluşturmaktır, böylece gelecekte yeni veri noktası doğru kategoriye yerleştirilebilecektir. Bu en iyi karar sınırına hiperdüzlem denir.

SVM, hiperdüzlemi oluşturmak için ekstrem nokta ya da vektörler seçer. Bunlara destek vektörleri adı verilir ve bu nedenle algoritma Destek Vektör Makinesi olarak adlandırılır. Şekil 3.5'deki diyagramda, bir karar sınırı veya hiperdüzlem kullanılarak sınıflandırılan iki farklı kategori bulunmaktadır [28].



Şekil 3.5. Hiper düzlem kullanarak sınıflandırma [30]

Hiperdüzlem, iki sınıf arasındaki margin adı verilen boşluğu maksimize eder. Margin, iki sınıfın en yakın noktaları arasındaki mesafedir.

Giriş vektörleri X ve karşılık gelen sınıf etiketleri Y olan bir eğitim veri kümesine alınsın. İki sınıflı bir sınıflandırma problemi için, veriler +1 ve -1 olarak etiketlenmiş olsun. Verilerin doğrusal bir hiperdüzlemle ayrılabilirliği varsayılır. Doğrusal hiperdüzlem denklemini şöyle yazılabilir:

$$w^T x + b = 0$$

W vektörü, hiperdüzleme dik olan normal vektörü temsil eder. Yani hiperdüzlem'e dik olan yön. Denklemdeki b parametresi, normal vektör w boyunca orijinden hiperdüzlemin uzaklığını veya ofsetini temsil eder. x_i veri noktası ile karar sınırı arasındaki mesafe şöyle hesaplanabilir:

$$d_i = \frac{w^T x_i + b}{\|w\|}$$

burada $\|w\|$, ağırlık vektörü w 'nin Öklid normunu temsil eder. Doğrusal SVM sınıflandırıcısı için normal vektör W 'nin Öklid normu şu şekildedir

$$\hat{y} = \begin{cases} 1: w^T x + b > 0 \\ 0: w^T x + b < 0 \end{cases}$$

Sınıflandırıcının optimizasyonu için hard margin ve soft margin svm kullanılır. Hard margin SVM, verilerin doğrusal olarak ayrılabilir olduğu ve hiçbir yanlış sınıflandırmaya izin verilmeyeceği durumlarda kullanılır. Bu durumda, iki sınıfı ayıran hiperdüzlemi bulmak için, hiperdüzleme en yakın veri noktaları arasındaki mesafe maksimize edilmelidir. Bu mesafe şöyle hesaplanabilir:

$$margin = \frac{2}{\|w\|}$$

Bu marjı maksimize etmek için, aşağıdaki optimizasyon problemi çözülür:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \text{ st } y_i(w^T x_i + b) \geq 1, \forall i = 1, 2, \dots, m$$

Bu problemdeki kısıtlar, veri noktalarının doğru sınıflandırılmasını sağlar.

Soft margin SVM ise, verilerin doğrusal olarak ayrılabilir olmadığı veya daha iyi genelleme elde etmek için bazı yanlış sınıflandırmalara izin verildiği durumlarda kullanılır. Bu durumda, marjı maksimize etmeye çalışırken, yanlış sınıflandırılan veya marj içinde kalan veri noktalarına bir ceza uygulanır. Bu ceza, C adında bir parametre ile kontrol edilir. C ne kadar büyükse, ceza o kadar büyük olur ve model o kadar katı olur. C ne kadar küçükse, ceza o kadar küçük olur ve model o kadar esnek olur.

Soft margin SVM için optimizasyon problemi şöyledir:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i$$

$$\text{st } y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i > 0, \forall i = 1, 2, \dots, m$$

Karar sınırının doğası temel alınarak, Destek Vektör Makineleri (SVM) iki ana bölüme ayrılabilir:

Doğrusal SVM: Doğrusal SVM'ler farklı sınıflardaki veri noktalarını ayırmak için doğrusal bir karar sınırı kullanır. Verilerin kesin bir şekilde doğrusal olarak ayrılabilir olduğu durumlarda, doğrusal SVM'ler çok uygundur. Bu, tek bir düz çizgi (iki boyut) veya bir hiperdüzlem (daha yüksek boyutlarda) ile veri noktalarını kendi sınıflarına tamamen ayırabileceğimiz anlamına gelir. Sınıflar arasındaki marjı maksimize eden bir hiperdüzlem karar sınırındır.

Doğrusal Olmayan SVM: Verilerin iki sınıfa bir doğru ile ayrılamadığı durumlarda (iki boyut durumunda olduğu gibi) verileri sınıflandırmak için kullanılabilir. Çekirdek fonksiyonlar kullanılarak, doğrusal olmayan SVM'ler doğrusal olarak ayrılabilir olmayan verilerle başa çıkabilir. Bu çekirdek fonksiyonlar, orijinal giriş verilerini daha yüksek boyutlu bir özellik uzayına dönüştürür ve veri noktalarının doğrusal olarak ayrılabilir olduğu bu değiştirilmiş uzayda doğrusal bir SVM kullanılarak doğrusal olmayan bir karar sınırı bulunur.

SVM çekirdeği, düşük boyutlu giriş uzayını alıp yüksek boyutlu uzaya dönüştüren bir fonksiyondur, yani ayrılabilir olmayan problemleri ayrılabilir problemlere dönüştürür. Doğrusal olmayan ayrılma problemlerinde çok faydalıdır. Basitçe söylemek gerekirse, çekirdek çok karmaşık veri dönüşümleri yapar ve sonra tanımlanan etiketlere veya çıktılara dayanarak verileri ayırma işlemini bulur. Literatürde bulunan çok sayıda çekirdek fonksiyonu arasında en popüler bazı çekirdek (Kernel) fonksiyonları aşağıda verilmiştir [26].

$$\text{Linear: } K(w, b) = w^T x + b$$

$$\text{Polynomial: } K(w, x) = (\gamma w^T x + b)^N$$

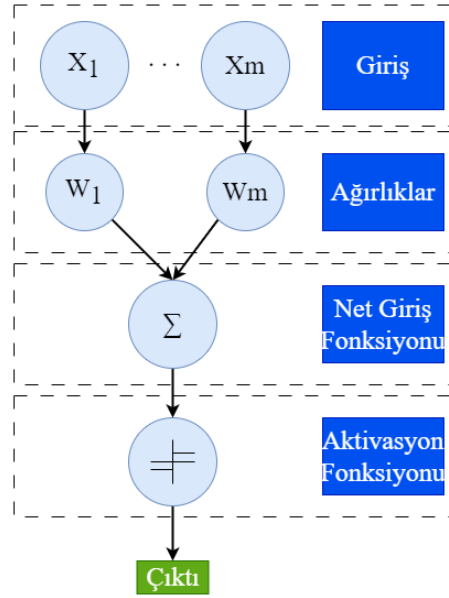
$$\text{Gaussian RBF: } K(w, b) = \exp(-\gamma \|x_i - x_j\|^n)$$

$$\text{Sigmoid: } K(x_i, x_j) = \tanh(\alpha x_i^T x_j + b)$$

Bu çalışmada da SVM algoritması için en yüksek performans skorlarını veren “rbf” çekirdek fonksiyonu kullanılmıştır.

3.3.3. Çok Katmanlı Algılayıcı (Multilayer Perceptron - MLP)

Algılayıcı (Perceptron), kaynağını insan beyninden alan bir yapay sinir ağı modelidir. Sinir ağları içindeki yaklaşım teorisi açısından önemli bir modeldir. Frank Rosenblatt tarafından 1957 yılında geliştirilen algılayıcı, basit bir yapay sinir hücresini temsil eder. Genel bir algılayıcı modeli Şekil 3.6'da gösterilmiştir.



Şekil 3.6. Algılayıcı modeli [31]

Bu hücre en temel işlem ünitesidir. Ortamdan veya diğer algılayıcılardan gelen giriş verilerine sahiptir. Her $x_i \in \mathbb{R}, i = 1, 2, \dots, m$ giriş verisiyle ilişkili bir $w_i \in \mathbb{R}$ bağlantı ağırlığı vardır ve en temel durumda y ağırlıkların girdilerinin ağırlıklı toplamıdır.

$$y = \sum_{j=1}^d w_j x_j + w_0$$

Burada w_0 modeli daha genel hale getiren ara kesme değeridir, genellikle ağırlığı +1 olan ek bir eğilim (bias) birimi x_0 dan gelen ağırlık olarak modellenir. MLP'nin çıktısı

$$y = w^T \cdot x$$

nokta çarpımı ile ifade edilebilir. Burada $w = [w_0, w_1, \dots, w_m]$ ve $x = [1, x_1, \dots, x_m]$ eğilim ağırlığı ve girişi içeren genişletilmiş vektörlerdir [32].

Sınıflandırma için algılayıcı, aktivasyon fonksiyonu olarak genellikle Heaviside adım fonksiyonunu kullanır:

$$f(x) = \begin{cases} 1 & , wx + b > 0 \\ 0 & , \text{diğer} \end{cases}$$

ama Sigmoid, Relu, Tanh gibi fonksiyonlar da kullanılabilir.

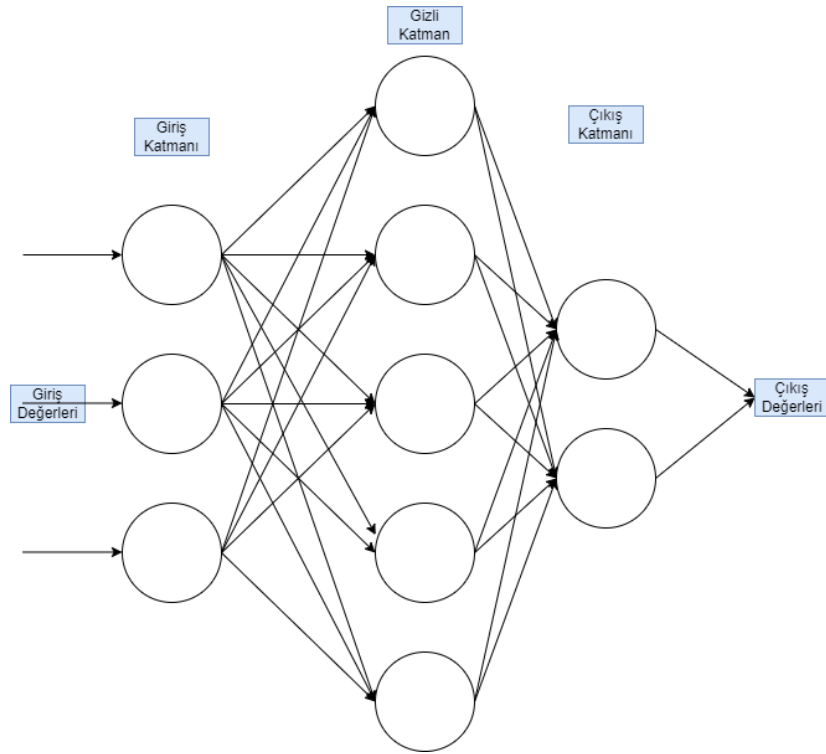
Gerçek çıktı ve beklenen çıktı arasındaki farka dayalı hata hesaplanarak bu hata ve öğrenme katsayısı α ile ağırlıklar güncellenir.

$$w(t + 1) = w(t) + \alpha \cdot \text{hata} \cdot x$$

Burada $w(t + 1)$ güncellenmiş ağırlıkları, $w(t)$ eski ağırlığı ve x giriş verilerini temsil eder. Hata 0'a ulaşana kadar ya da verilen iterasyon sayısı kadar ağırlık güncelleme işlemi devam eder. Eğitim tamamlandıktan sonra yeni algılayıcı modeli ile tahmin yapılır.

Tek katmanlı ağırlıklara sahip bir algılayıcı girdilerin doğrusal fonksiyonlarıyla yaklaşabilir ve lineer olmayan problemleri çözemez. Benzer şekilde algılayıcı lineer olmayan regresyon için kullanılmaz. Bu durumda Çok Katmanlı Algılayıcı (MLP) kullanılır [32].

MLP bir giriş katmanı, bir veya daha fazla gizli katman ve bir çıkış katmanından oluşur. Her katmandaki birimler (nöronlar), giriş katmanından başlayarak çıkış katmanına doğru bağlantılarla birbirine bağlanır. MLP'nin temel yapısı, giriş verilerini gizli katmanlardaki ağırlıklar ve aktivasyon fonksiyonları aracılığıyla işleyerek bir çıkış üretmektir. Her bir gizli katmanda, girişler, ağırlıklarla çarpılır, bir aktivasyon fonksiyonuna tabi tutulur ve ardışık katmanlara iletilir. Her katmanda farklı aktivasyon fonksiyonları kullanılabilir. Bu işlem, gizli katmanlardan geçerken tekrarlanır ve sonunda çıkış katmanında bir çıkış elde edilir. Örneğin bu çalışmada aktivasyon fonksiyonu olarak "relu" kullanılmıştır. Genel olarak bir MLP modeli Şekil 3.7'de gösterilmiştir.



Şekil 3.7. Basit bir MLP modeli [31]

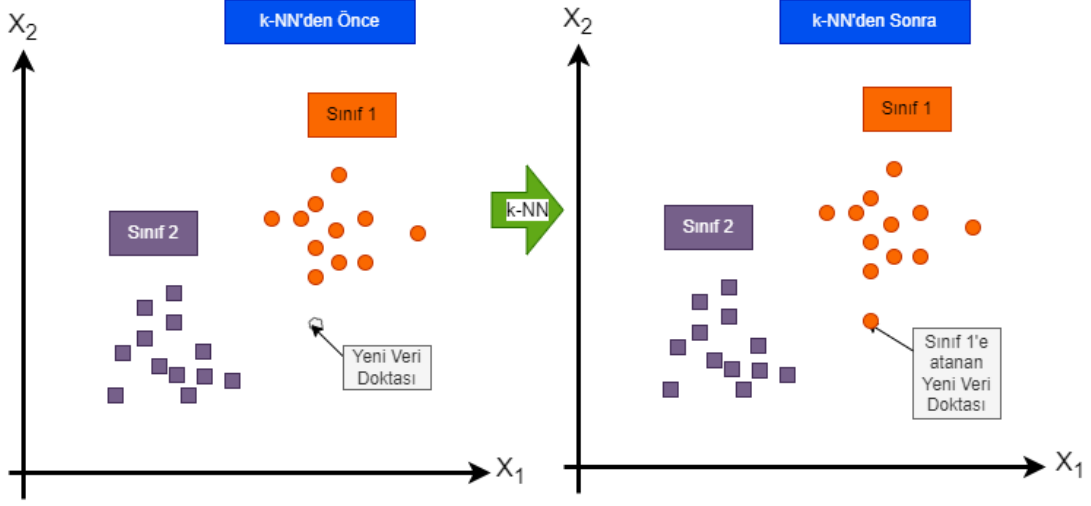
MLP modeli, giriş verilerinin karmaşık ve non-lineer ilişkilerini yakalamak için kullanılır. Bu, çeşitli aktivasyon fonksiyonları (örneğin sigmoid, ReLU, tanh) kullanarak yapılır. MLP, bu şekilde veriye uyan bir öğrenme süreciyle ağırlıkları ayarlar ve ardından yeni veriler üzerinde tahminler yapabilir. MLP'nin geniş bir uygulama alanı vardır ve örüntü tanıma, sınıflandırma, regresyon, zaman serisi analizi gibi birçok problemde başarılı sonuçlar elde edebilir. Ayrıca, MLP'nin çoklu gizli katmanlara sahip olması, daha karmaşık ilişkileri ve özneliklerin öğrenilmesini mümkün kılar [31].

3.3.4. k-Nearest Neighbours (k-NN)

k-NN, parametrik olmayan bir öğrenme algoritmasıdır. Model oluşturulduktan sonra eğitim verilerini atma izni veren diğer öğrenme algoritmalarının aksine, k-NN tüm eğitim örneklerini hafızada tutar. Daha önce görülmemiş bir örnek x geldiğinde, k-NN algoritması x 'e en yakın olan k eğitim örneğini bulur ve sınıflandırma durumunda çoğunluk etiketini, regresyon durumunda ise ortalama etiketini döndürür [26].

Bu algoritmanın adımları şunlardır:

- Önce k parametresi seçilir. Bu parametre, verilen bir örneğe en yakın olan komşu sayısını gösterir. Örneğin $k = 3$ ise, en yakın 3 komşu dikkate alınacaktır.
- Verilen örneğin diğer örneklerden ne kadar uzak olduğu hesaplanır. Bunun için çeşitli uzaklık ölçütleri kullanılabilir. Örneğin Öklid uzaklığı, Manhattan uzaklığı, Minkowski uzaklığı gibi.
- Daha sonra uzaklıklar küçükten büyüğe doğru sıralanır ve en küçük k uzaklıktaki örnekler belirlenir. Bu örnekler en yakın k komşu olarak isimlendirilir.
- Eğer problem bir sınıflandırma problemiyse, en yakın k komşunun ait olduğu sınıflar incelenir ve çoğunlukla görülen sınıf tahmin olarak verilir.
- Eğer problem bir regresyon problemiyse, en yakın k komşunun hedef değişken değerlerinin ortalaması alınır ve bu değer tahmin olarak sunulur.



Şekil 3.8. k-NN Modeli [33]

k-NN algoritması, basit ve kolay uygulanabilir bir yöntemdir. Fakat bazı dezavantajları da bulunmaktadır. Örneğin, k parametresinin doğru seçilmesi önemlidir, çok büyük veya çok küçük olması başarıyı azaltabilir. Ayrıca her yeni örnek için tüm veri seti ile uzaklık hesaplaması yapmak zaman ve bellek açısından maliyetli olabilir [34]. Örnek bir k-NN modeli Şekil 3.8'de verilmiştir.

3.3.5. Extreme Gradient Boosting (XGBoost)

Extreme gradient boosting algoritması, gradient boosted trees algoritmasının optimize edilmiş ve ölçeklenebilir bir uygulamasıdır. XGBoost, yüksek tahmin gücü, aşırı uyum önleme, boş veri yönetimi ve hızlı hesaplama gibi özellikleriyle sıkça kullanılan ve başarılı sonuçlar veren bir algoritmadır. İlk olarak Chen ve arkadaşları tarafından verilmiştir [35].

XGBoost algoritmasının çalışma mantığı, gradient boosting algoritmasına çok benzerdir. Öncelikle, tüm veriler için bir ilk tahmin yapılır. Bu tahmin, ortalama değer gibi basit bir istatistik olabilir. Sonra, her veri noktası için gerçek değer ile ilk tahmin arasındaki fark hesaplanır. Bu farka kalıntı (residual) denir. Kalıntıları hedef değişken olarak kullanarak yeni bir karar ağacı oluşturulur. Bu ağaç, kalıntıları en iyi şekilde açıklayan dallara sahiptir. Daha sonra, ilk tahmin ile yeni ağacın tahminleri arasında bir ağırlıklı ortalama alınır. Bu ağırlık, öğrenme oranı (learning rate) olarak adlandırılan bir hiperparametre ile belirlenir. Öğrenme oranı, modelin ne kadar hızlı öğrendiğini kontrol eder. Öğrenme oranı küçükse, model daha yavaş öğrenir ama daha doğru olabilir. Öğrenme oranı büyükse, model daha hızlı öğrenir ama daha fazla hata yapabilir. Bu işlem, belirlenen bir

durdurma kriterine kadar tekrarlanır. Durdurma kriteri, maksimum ağaç sayısı, minimum kalıntı değeri veya doğrulama verisi üzerindeki performans şeklinde olabilir.

Matematiksel olarak, XGBoost algoritması şöyle ifade edilebilir:

- Verilen bir veri kümesi $(x_1, y_1), \dots, (x_n, y_n)$ ve bir kayıp fonksiyonu $L(y, f(x))$ olsun.

Burada x girdi değişkeni, y çıktı değişkeni ve $f(x)$ tahmin edici fonksiyondur.

- İlk tahmin yaprağı $f_0(x)$ şöyle hesaplanır:

$$f_0(x) = \operatorname{argmin}_y \sum_{i=1}^n L(y_i, y)$$

- Her iterasyonda $m = 1, \dots, M$ için, kalıntılar r_{im} şöyle hesaplanır:

$$r_{im} = - \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]$$

- Kalıntılara göre yeni bir karar ağacı $h_m(x)$ oluşturulur.

- Ağacın dallarına göre her bölge için optimal değerler γ_m şöyle hesaplanır:

$$\gamma_m = \operatorname{argmin}_y \sum_{x_i \in R_j} L(y_i, f_{m-1}(x_i) + \gamma) \quad j = 1, \dots, J_m$$

Burada R_j j 'inci bölgeyi ve J_m m 'inci ağacın bölge sayısını gösterir.

- Son olarak, yeni tahmin edici fonksiyon $f_m(x)$ şöyle güncellenir:

$$f_m(x) = f_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_m I(x \in R_j)$$

Burada ν öğrenme oranını ve $I(x \in R_j)$ x 'in j 'inci bölgede olup olmadığını gösteren bir gösterge fonksiyonudur.

- Bu işlem M iterasyonu tamamlanana kadar devam eder. Son tahmin edici fonksiyon $f_M(x)$ şöyle olur:

$$f_M(x) = f_0(x) + \nu \sum_{m=1}^M \sum_{j=1}^{J_m} \gamma_m I(x \in R_j)$$

XGBoost algoritması, gradient boosting algoritmasından farklı olarak, ağaçların karmaşıklığını kontrol etmek için bir düzenleme (regularization) terimi ekler.

Düzenleştirme, modelin aşırı uyum yapmasını önlemek için kullanılan bir tekniktir. XGBoost algoritmasında, düzenleştirme terimi şöyle tanımlanır:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

Burada γ yaprak sayısının cezasını, λ yaprak ağırlıklarının cezasını, T yaprak sayısını ve w_j j 'inci yaprağın ağırlığını gösterir. Bu terim, kayıp fonksiyonuna eklenerek, modelin daha basit ve genelleştirilebilir olması sağlanır.

XGBoost algoritması, ayrıca, verilerin alt kümesini rastgele seçerek ve ağaç dallarındaki bölünme noktalarını rastgele seçerek ekstra bir rastgelelik parametresi sunar. Bu parametre, modelin çeşitliliğini artırarak ve aşırı uyum riskini azaltarak performansını iyileştirir.

XGBoost algoritması, hesaplama maliyetini düşürmek ve paralelleştirmeyi kolaylaştırmak için de bazı optimizasyon teknikleri kullanır. Örneğin, verileri sütun bazlı olarak depolayarak, bölünme noktalarının hızlı bir şekilde bulunmasını sağlar. Ayrıca, verileri bloklara bölerek, birden fazla işlemci veya makine üzerinde paralel olarak çalıştırılmasını mümkün kılar [35].

3.3.6. Lojistik Regresyon (LR)

LR, ikili sınıflandırma problemlerinde kullanılan bir istatistiksel yöntemdir. İkili sınıflandırma, bir veri noktasının iki olası kategoriye ait olup olmadığını belirlemek için kullanılır. Örneğin, bir e-postanın spam olup olmadığını, bir hastanın hastalıklı olup olmadığını veya bir müşterinin bir ürünü satın alıp almayacağını gibi iki seçenek üzerinden işlem yapar. Çok sınıflı veri setlerinde ise LR, birine karşı hepsi ("one-vs-all") stratejisini kullanır. Her bir sınıfın olasılığını tahmin etmek için ayrı ayrı bir LR modeli oluşturulur. Her model, bir sınıfın olasılığını tahmin etmek için veri kümesinin geri kalanını temsil eden diğer sınıflarla karşılaştırılır. Bu çalışmada çok sınıflı veri seti kullanılmış ve bu nedenle LR modeli çok sınıflı yapıya uygun oluşturulmuştur.

Lojistik regresyonun temel fikri, bağımsız değişkenlerin (X) bir lineer kombinasyonunu alarak bağımlı değişkenin (Y) olasılığını tahmin etmektir. Ancak, lineer regresyonda olduğu gibi doğrudan Y 'yi tahmin etmek yerine, Y 'nin logit dönüşümünü tahmin eder. Logit dönüşümü, Y 'nin olasılığının logaritmik oranıdır. Yani,

$$\text{logit}(Y) = \log \frac{P(Y = 1)}{P(Y = 0)}$$

Logit dönüşümü, Y 'nin değerini $-\infty$ ile $+\infty$ arasında değişen bir sayıya dönüştürür. Bu sayede, lineer regresyon modeli uygulanabilir. Lineer regresyon modeli şöyledir:

$$\text{logit}(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$$

Burada, β_0 sabit terim, $\beta_1, \beta_2, \dots, \beta_n$ katsayılar ve X_1, X_2, \dots, X_n bağımsız değişkenlerdir. Bu modelde, katsayılar, bağımsız değişkenlerin Y 'nin logit dönüşümü üzerindeki etkisini gösterir. Örneğin, β_1 bir birim arttığında, $\text{logit}(Y)$, β_1 birim artar.

Modeli tahmin etmek için, veri setindeki gözlemleri kullanarak katsayıları bulmak gerekir. Bunun için en çok kullanılan yöntem maksimum olabilirlik yöntemidir. Maksimum olabilirlik yöntemi, verilen katsayılara göre veri setinin ortaya çıkma olasılığını maksimize eden katsayıları bulmaya çalışır. Bu yöntemle elde edilen katsayılar, lojistik regresyon modelinin parametreleri olarak adlandırılır.

Lojistik regresyon modeli kurulduktan sonra, yeni veri noktalarının sınıflarını tahmin etmek için kullanılabilir. Bunun için, yeni veri noktasının bağımsız değişkenlerini modele yerleştirerek $\text{logit}(Y)$ değerini hesaplamak gerekir. Daha sonra, $\text{logit}(Y)$ değerini tersine çevirerek Y 'nin olasılığını bulmak gerekir. Tersine çevirme işlemi şöyledir:

$$P(Y = 1) = \frac{e^{\text{logit}(Y)}}{1 + e^{\text{logit}(Y)}}$$

Bu formülde, e matematiksel sabit olan Euler sayısıdır. Bu formül, lojistik fonksiyonu olarak da bilinir ve S şeklinde bir eğri çizer. Lojistik fonksiyonu, Y 'nin değerini 0 ile 1 arasında sınırlar ve olasılık yorumuna uygun hale getirir.

Y 'nin olasılığını hesapladıktan sonra, belirlenen bir eşik değerine göre Y 'nin sınıfını belirlemek gerekir. Genellikle eşik değeri 0.5 olarak alınır. Yani, eğer $P(Y = 1) \geq 0.5$ ise, Y 'nin sınıfı 1 olarak tahmin edilir. Eğer $P(Y = 1) < 0.5$ ise, Y 'nin sınıfı 0 olarak tahmin edilir.

Lojistik regresyon modelinin performansını değerlendirmek için, gerçek sınıflar ile tahmin edilen sınıflar arasındaki uyumu ölçmek gerekir [12].

3.3.7. K-Katlı Çapraz Doğrulama (K-Fold Cross Validation)

Çapraz Doğrulama, makine öğrenmesi modellerinin performansını değerlendirmek için

kullanılan bir yöntemdir. Çapraz Doğrulama, veri setini eğitim ve test alt kümelerine bölerek, modelin hem eğitim hem de test verilerinde ne kadar iyi çalıştığını ölçer. Çapraz Doğrulama, modelin overfitting (aşırı uyum) veya underfitting (yetersiz uyum) olup olmadığını belirlemeye yardımcı olur.

K-Katlı Çapraz Doğrulama, Çapraz Doğrulama'nın en yaygın uygulanan biçimidir. K-Katlı Çapraz Doğrulama, veri setini K eşit parçaya bölerek, her seferinde bir parçayı test, geri kalan K-1 parçayı ise eğitim olarak kullanır. Bu işlem K kez tekrarlanır ve her parça bir kez test olarak kullanılır. Sonuç olarak, modelin K farklı eğitim-test bölünmesindeki performansı elde edilir. Bu performans sonuçları ortalama alınarak, modelin genelleştirme yeteneği hakkında bir tahmin yapılır.

K-Katlı Çapraz Doğrulama, model seçimi ve parametre ayarlaması gibi amaçlar için de kullanılabilir. K değeri arttıkça, modelin varyansı azalır ancak hesaplama maliyeti artar. Genellikle K=5 veya K=10 olarak seçilir. Bu çalışmada K değeri 10 alınmıştır.

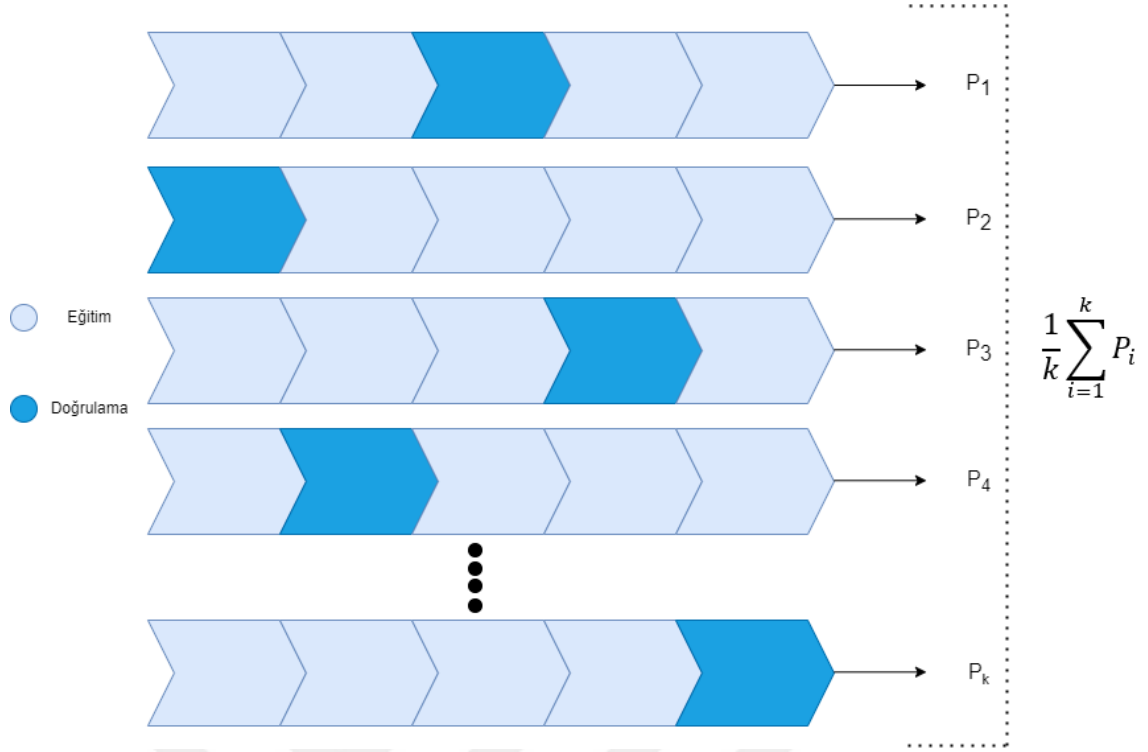
K-Katlı Çapraz Doğrulama'nın avantajları şunlardır:

- Veri setinin tamamını hem eğitim hem de test için kullanır.
- Modelin genelleştirme yeteneği hakkında güvenilir bir tahmin sağlar.
- Rastgelelikten etkilenmez.

K-Katlı Çapraz Doğrulama'nın dezavantajları şunlardır:

- Hesaplama maliyeti yüksektir.
- Veri setinin dağılımına bağlı olarak yanlış sonuçlar verebilir.
- Zaman serisi veya gruplu veriler için uygun değildir [36].

K-Katlı Çapraz Doğrulama akış şeması Şekil 3.9'da verilmiştir.



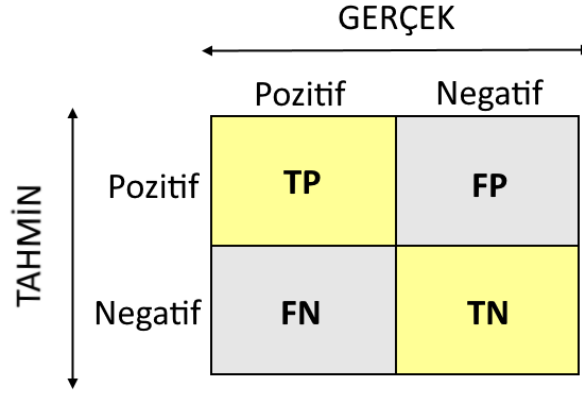
Şekil 3.9. K-Katlı Çapraz Doğrulama akış şeması [37]

3.4. PERFORMANS SKORLARI

Bu çalışmada makine öğrenmesi modellerinin sınıflandırma performansını ölçmek için farklı metrikler kullanılmıştır. Kullanılan bu metrikler aşağıda tanımları ile birlikte verilmiştir.

3.4.1. Karışıklık Matrisi (Confusion Matrix)

Sınıflandırma modelinin performansını değerlendirmek için kullanılan bir matristir ve sınıflandırma modelinin doğru ve yanlış tahminlerinin sayısını gösterir. Karışıklık Matrisi, dört ana kategori içerir: True Positive (TP), False Positive (FP), True Negative (TN) ve False Negative (FN). Bu kategoriler, sınıflandırma modelinin hangi durumlarda doğru veya yanlış tahmin yaptığını gösterir. Örnek bir Karışıklık Matrisi yapısı Şekil 3.10'da verilmiştir [38].



Şekil 3.10. Genel olarak bir Karışıklık Matrisi

3.4.2. Doğruluk (Accuracy)

Doğru sınıflandırılmış örneklerin toplam örnek sayısına oranı ifade eder. Yani, doğru sınıflandırılmış tüm örneklerin toplam örnekler içindeki oranını gösterir. Sınıflandırmanın doğruluğunu temsil eder. Doğruluk 0 ile 1 arasında değer alır. Yüksek çıkması sonucun doğruluğunun yüksek olduğunu ifade eder [38].

$$\text{Doğruluk} = \frac{TP + TN}{TP + TN + FP + FN}$$

3.4.3. Duyarlılık (Recall)

Duyarlılık, bir sınıflandırma modelinin gerçek pozitif örneklerin ne kadarını doğru bir şekilde tanımladığını gösteren bir metriktir. Duyarlılık, aynı zamanda "duyarlılık" veya "hassasiyet" olarak da adlandırılır. Duyarlılık değeri 0 ile 1 arasında bir değer alır. Yüksek bir duyarlılık değeri, modelin pozitif sınıfı kaçırma olasılığının düşük olduğunu gösterir ve modelin gerçek pozitifleri yüksek bir doğrulukla tespit ettiğini ifade eder [38].

$$\text{Duyarlılık} = \frac{TP}{TP + FN}$$

3.4.4. Kesinlik (Precision)

Kesinlik, bir sınıflandırma modelinin pozitif olarak tahmin ettiği örneklerin ne kadarının gerçekten pozitif olduğunu gösteren bir değerdir. Daha basit bir ifadeyle, kesinlik, modelin doğru pozitif tahminlerinin toplam pozitif tahminler içindeki oranını ifade eder. Kesinlik, sınıflandırmanın hassasiyetini temsil eder. Kesinlik değeri de 0 ile 1 arasında bir değer alır. Yüksek bir kesinlik değeri, modelin pozitif olarak tahmin ettiği örneklerin

gerçek pozitif olma olasılığının yüksek olduğunu gösterir ve modelin yanlış pozitifleri düşük bir oranda tahmin ettiğini ifade eder [38].

$$Kesinlik = \frac{TP}{TP + FP}$$

3.4.5. F1

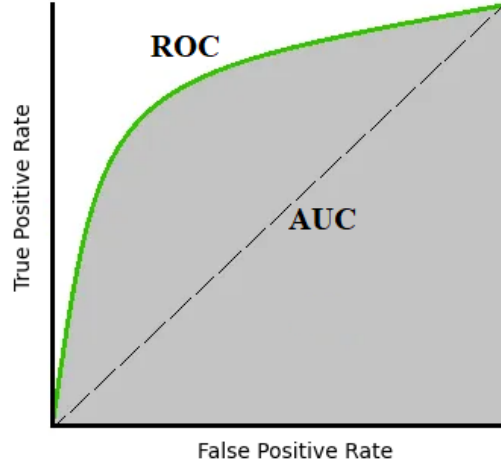
Kesinlik ve duyarlılık metriklerinin harmonik ortalamasıdır ve modelin performansını daha geniş bir perspektiften değerlendirmeye yardımcı olur. F1 skor değeri 0 ile 1 arasında değişir ve yüksek bir değer, modelin iyi bir denge sağladığını ve sınıflandırmanın doğruluğunun yüksek olduğunu gösterir [38].

$$F1 = \frac{2 \times Kesinlik \times Duyarlılık}{Kesinlik + Duyarlılık}$$

3.4.6. ROC Eğrisi ve AUC Skoru

Alıcı İşlem Karakteristiği Eğrisi (Receiver Operating Characteristic Curve) yani ROC eğrisi, sınıflandırma problemleri için performans metriği olarak kullanılan bir grafikdir. ROC eğrisi, farklı eşik değerlerinde duyarlılık (gerçek pozitif oranı) ile yanlış pozitif oranı (gerçek negatiflerin yanlış pozitif olarak sınıflandırılma oranı) arasındaki ilişkiyi gösterir. ROC eğrisi, modelin sınıfları ne kadar iyi ayırt edebildiğini ve farklı eşik değerlerinin modelin performansını nasıl etkilediğini görmemizi sağlar. ROC eğrisinin sol üst köşesine yakın olması, modelin yüksek duyarlılık ve düşük yanlış pozitif oranına sahip olduğunu, yani iyi bir performans gösterdiğini ifade eder. ROC eğrisinin sağ alt köşesine yakın olması, modelin düşük duyarlılık ve yüksek yanlış pozitif oranına sahip olduğunu, yani kötü bir performans gösterdiğini ifade eder. ROC eğrisinin diyagonal yani 45 derecelik doğruya yakın olması, modelin rastgele tahmin yaptığını gösterir.

Eğrinin altındaki Alan (Area Under the Curve) yani AUC skoru, ROC eğrisinin altında kalan alanı ifade eder. AUC skor, 0 ile 1 arasında değer alır ve 1'e ne kadar yakın olursa model de o kadar başarılıdır. AUC skor, modelin ayrılabirlik derecesini veya ölçüsünü temsil eder. Yani, modelin gerçek pozitifleri gerçek negatiflerden ne kadar iyi ayırt edebildiğini gösterir. AUC skorunun yüksek olması, modelin sınıflandırma problemini iyi çözdüğü anlamına gelir. Örnek bir ROC Eğrisi ve AUC skoru Şekil 3.11'de verilmiştir [38].

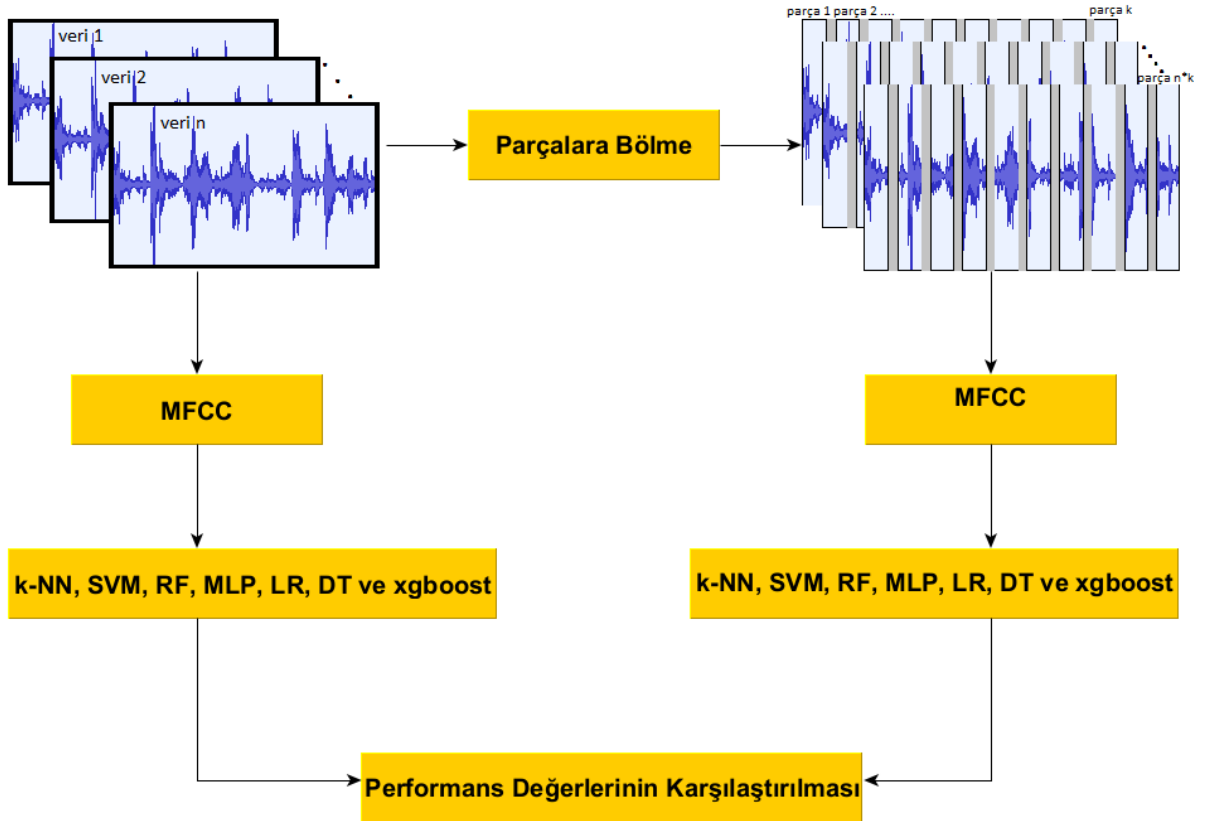


Şekil 3.11. ROC Eğrisi ve AUC skoru [38]



4. YÖNTEM

Bu çalışmada öncelikli amacımız bebek ağlamalarının sınıflandırılması konusundaki çalışmalara katkıda bulunmaktır. Bu amaç doğrultusunda öncelikle kullanılacak veri seti belirlenmiştir. Veri seti belirlenirken açık kaynak veri setleri incelenmiş ve son zamanlarda birçok çalışmada yer bulan, ses işleme ön adımları (sessizliği silme, gürültü azaltma vb.) uygulanmış verilerden oluşan “Karın Ağrısı”, “Geğirme”, “Huzursuzluk”, “Açlık” ve “Yorgunluk” şeklinde beş sınıfa ayrılmış olarak paylaşılan Donate-a-cry veri seti kullanılmıştır. Kullanılan veri setinin veri sayısı yetersiz görülerek veri artırma yöntemleri incelenmiş, müzik parçalarının kısa bir kesitinden tanınması için uygulanan verileri eşit parçalara bölme yöntemi kullanılmıştır [39], [40]. Bu yöntemle her veri 2’den 15’e kadar parçalara ayrı ayrı bölünmüş bu bölme işlemi sonucu, çalışmada ana veri seti olarak adlandırılan Donate-a-cry da dahil 15 veri seti oluşturulmuştur (Çizelge 4.1).



Şekil 4.1. Yöntemin Akış Şeması

Çizelge 4.1. Parçalara bölme yöntemi ile oluşan veri setleri ve veri sayıları

Donate-a-cry	KA	GE	HU	AÇ	YR	Toplam
Ana Veri Seti	16	8	27	382	24	457
vs02	32	16	54	764	48	914
vs03	48	24	81	1146	72	1371
vs04	64	32	108	1528	96	1828
vs05	80	40	135	1910	120	2285
vs06	96	48	162	2292	144	2742
vs07	112	56	189	2674	168	3199
vs08	128	64	216	3056	192	3656
vs09	144	72	243	3438	216	4113
vs10	160	80	270	3820	240	4570
vs11	176	88	297	4202	264	5027
vs12	192	96	324	4584	288	5484
vs13	208	104	351	4966	312	5941
vs14	224	112	378	5348	336	6398
vs15	240	120	405	5730	360	6855

Veri setlerinin oluşturulmasından sonra her veri için MFCC kullanılarak öznitelik çıkarma işlemi uygulanmıştır. MFCC uygulamak için her parçanın ses sinyalleri 22050Hz örnekleme oranı ve %50 örtüşme ile pencerelere ayrılmıştır. Pencereleme işlemi, sinyalin zaman ve frekans bileşenlerini daha iyi görmemizi sağlar ve Fourier dönüşümünde yan etkileri azaltır. Daha sonra her pencere için Mel-frekans filtre bankası ve ayrık kosinüs dönüşümü uygulanmıştır. Sonuç olarak bir veri seti için Çizelge 4.2’de görüldüğü gibi bir

tensör oluşmuştur.

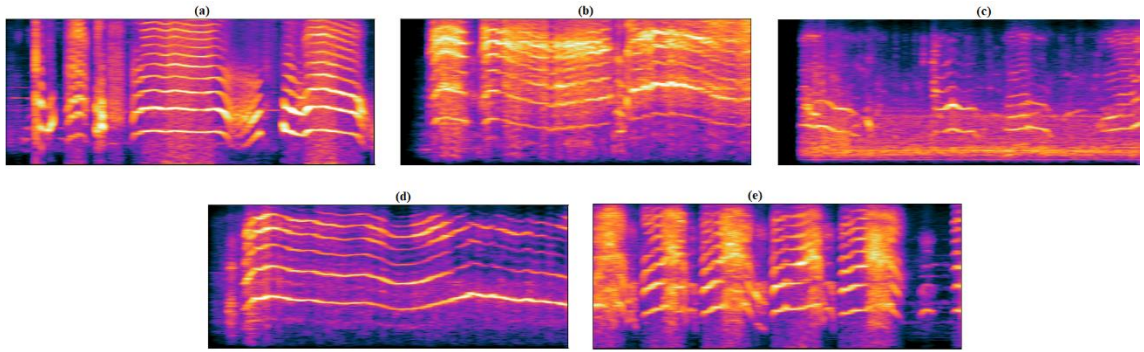
Çizelge 4.2. Ana veri seti ve vs02 veri seti için oluşan tensör

	Veri Sayısı	Pencere Sayısı	MFCC Sayısı	Tensör Boyutu
Ana veri seti	457	130	13	(457,130,13)
vs02 veri seti	914	65	13	(914,65,13)

Karın Ağrısı sınıfına ait bir ses dosyasının bir penceresine ait MFCC katsayıları örneği aşağıdaki gibidir.

(-557.8212, 118.20502,-39.979004, 2.4987278, 22.138535, -25.614408, -15.321253, -0.86993605, -11.822699, -7.62405, -12.903666, -16.226084, -7.451709)

Sonuç olarak her pencere için 13 adet MFCC katsayısı hesaplanmıştır. MFCC Sınıflara ait örnek ses dosyalarının birer sinyal ve spektrogram görüntüleri Şekil 4.2’de verilmiştir.



Şekil 4.2. (a) karın ağrısı, (b) geğirme, (c) huzursuzluk, (d) açlık ve (e) yorgunluk sınıflarına ait birer spektrogram görüntüsü

Öznitelik vektörleri ve etiketler oluşturulduktan sonra sınıflandırma için XGBoost, k-NN, MLP, RF, LR ve SVM için performans skorları hesaplanmıştır. Çalışmada kullanılan sınıflandırma algoritmalarına ait bazı önemli parametreler Çizelge 4.3’de verilmiştir.

Çizelge 4.3. Sınıflandırma algoritmalarının bazı önemli parametreleri

Algoritma		p1	p2	p3	p4
XGBoost	Parametre	learning_rate	n_estimators	max_depth	subsample
	Değer	0,3	100	3	1
k-NN	Parametre	n_neighbors	weights	algorithm	
	Değer	5	'uniform'	'auto'	
MLP	Parametre	hidden_layer_sizes	activation	solver	
	Değer	100	'relu'	'adam'	
RF	Parametre	n_estimators	max_features		
	Değer	100	“sqrt”		
LR	Parametre	penalty	multi_class	C	
	Değer	'l2'	multinomial	1	
	Değer	'gini'			
SVM	Parametre	C	kernel	gamma	
	Değer	1	'rbf'	'scale'	

Çalışmada performans skorları 10 katlı Çapraz Doğrulama ile elde edilmiştir. 10 katlı çapraz doğrulama ile veri setindeki verilerin hepsi hem test hem de eğitim için kullanılması sağlanarak veri israfı önlenmiştir. Ayrıca doğruluk, duyarlılık, kesinlik ve f1 skorları 30 kez hesaplanmıştır. Hesaplanan değerlerin en yüksek, en düşük değerleri ve ortalamaları bulgular kısmına eklenmiştir. Yapılan bu işlemler sonucu performans değerlerinin güvenilirliği ve objektifliği artırılmıştır. Çalışmada uygulanan yöntemin akış şeması Şekil 4.1’de verilmiştir.

5. BULGULAR

Bu bölümde elde edilen sonuçlar yedi başlık altında verilmiştir. İlk altı başlıkta her algoritma için ayrı ayrı elde edilen bulgular verilmiş, son başlıkta ise algoritmaların sonuçları birbirleri ile karşılaştırılmıştır.

5.1. XGBOOST ALGORİTMASI İLE ELDE EDİLEN BULGULAR

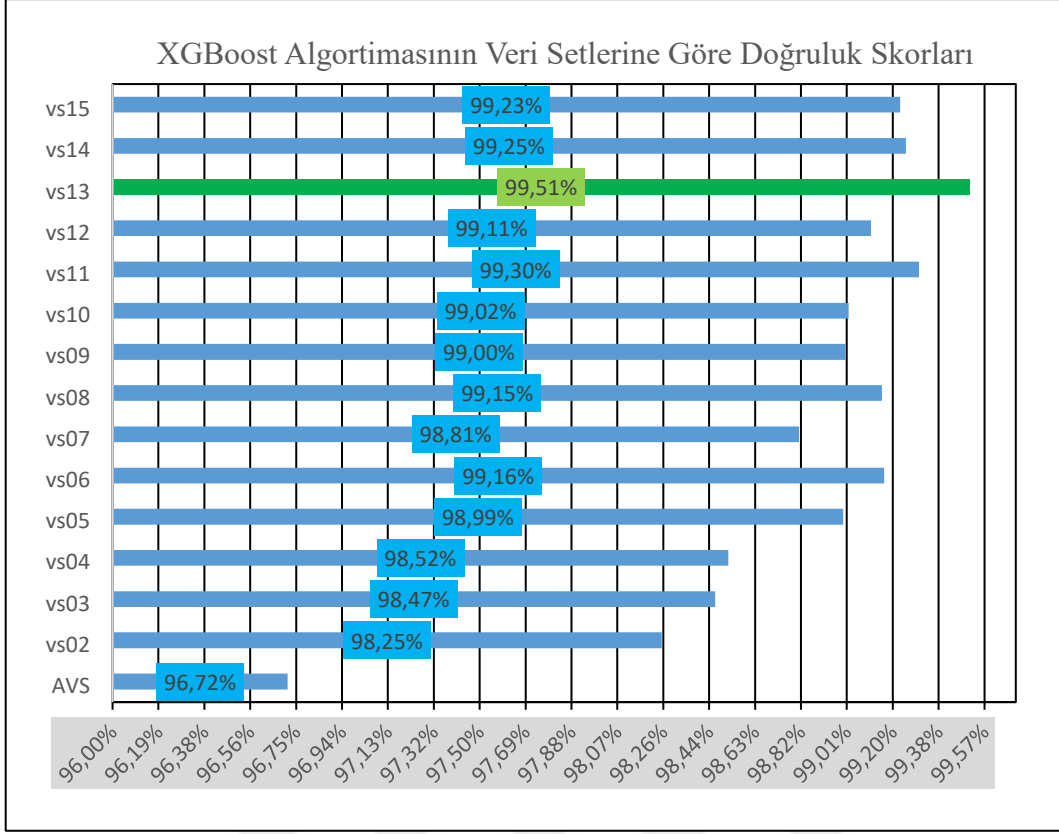
Bu kısımda, XGBoost algoritması ile elde edilen bulgular yer almaktadır.

Şekil 5.1’de oluşturulan veri setleri üzerinde XGBoost algoritması ile elde edilen doğruluk skorlarının dağılımı verilmiştir.

Çizelge 5.1’de, XGBoost algoritması ile ana veri seti ve en yüksek performans skoru alınan veri setinin karşılaştırılması adına veri setleri üzerinde elde edilen doğruluk, duyarlılık, kesinlik ve f1 skorlarının 30 kez hesaplanan değerleri arasından en yüksek, en düşük ve ortalama değerlerine yer verilmiştir.

Şekil 5.2’de XGBoost algoritması için ana veri seti ve en yüksek performans skoru elde edilen veri seti kullanılarak oluşturulan Karışıklık Matrisleri yer almaktadır. Bu Karışıklık Matrisleri, sınıflar arasındaki farklılaşma veya benzerliği ve bu farkın karşılaştırılan veri setlerine bağlı olarak nasıl değiştiğini gözlemlemek için kullanılmıştır.

Son olarak Şekil 5.3’de XGBoost algoritması için ana veri seti ve en yüksek performans skoru alınan veri seti ile oluşturulan grafikler verilmiştir. Bu grafikler sınıfların ROC eğrilerini ve AUC skorlarını göstermektedir. Sınıfların farklılaşma veya benzerliğini görmek ve sınıflandırma performansının veri setine göre değişimini gözlemlemek adına kullanılmıştır.



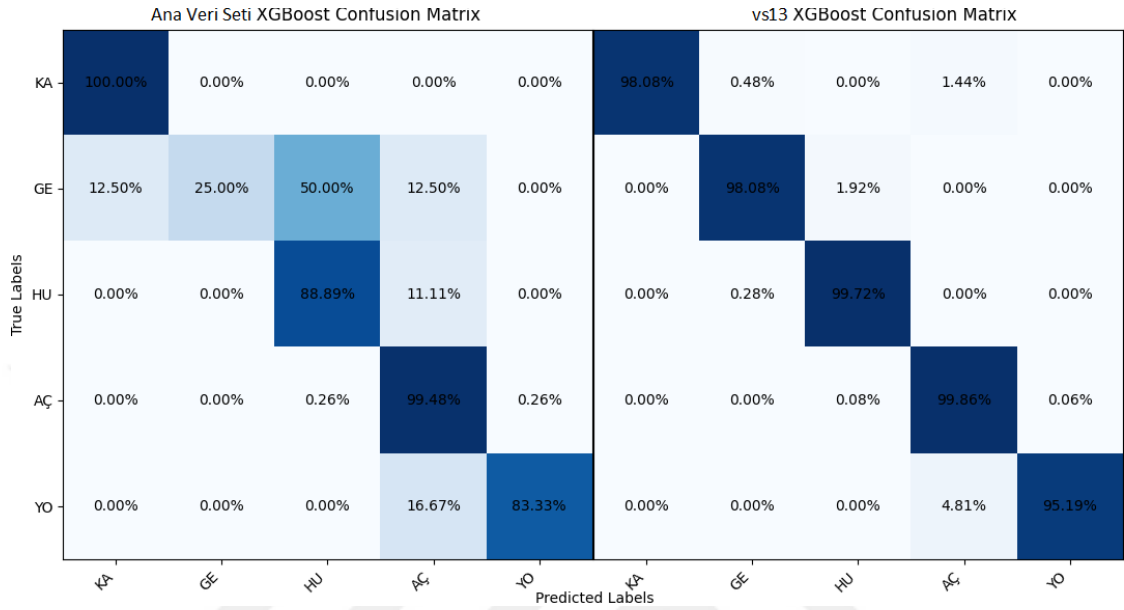
Şekil 5.1. Tüm veri setleri üzerinde XGBoost ile hesaplanan doğruluk skorları

Şekil 5.1’de görülmektedir ki XGBoost algoritmasıyla elde edilen en düşük doğruluk skoru ana veri seti kullanılarak bulunmuştur. Diğer taraftan, en yüksek doğruluk skoru ise verilerin 13 parçaya bölüdüğü vs13 veri seti kullanılarak elde edilmiştir.

Çizelge 5.1. İki veri seti üzerinde XGBoost ile performans skorlarının karşılaştırılması

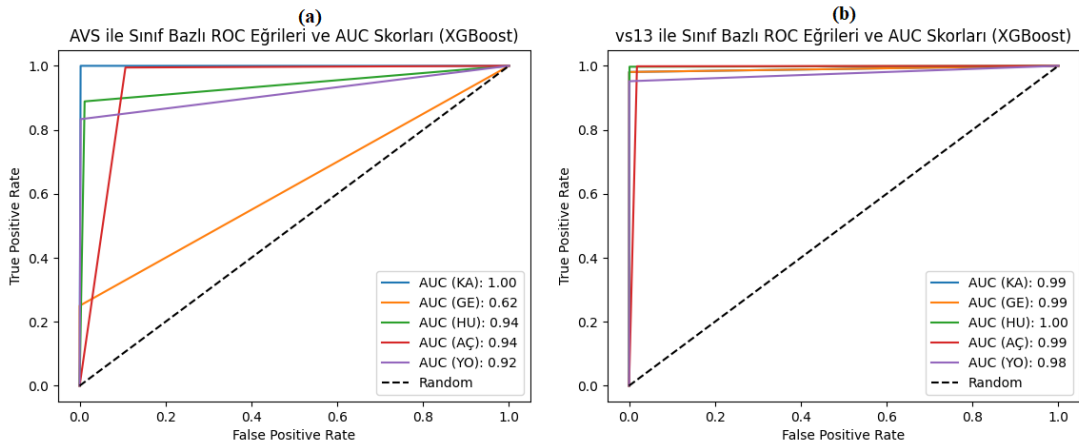
XGBoost	Ana Veri Seti			vs13		
	En Yüksek	En Düşük	Ortalama	En Yüksek	En Düşük	Ortalama
Doğruluk ↑	96,72%	96,72%	96,72%	99,51%	99,51%	99,51%
Duyarlılık ↑	96,72%	96,72%	96,72%	99,51%	99,51%	99,51%
Kesinlik ↑	96,80%	96,80%	96,80%	99,51%	99,51%	99,51%
F1 ↑	96,33%	96,33%	96,33%	99,51%	99,51%	99,51%

Çizelge 5.1’de vs13 veri seti ve ana veri seti kullanılarak hesaplanan XGBoost algoritması performans skorları incelendiğinde vs13 veri seti ile her skorda ortalama %2,87 yükselme olduğu hesaplanmıştır.



Şekil 5.2. İki veri seti üzerinde XGBoost ile oluşturulan Karışıklık Matrislerinin karşılaştırması

Şekil 5.2’de görüldüğü üzere, XGBoost algoritmasıyla elde edilen sonuçlara göre, "Karnın ağrısı" sınıfı dışında diğer tüm sınıflarda daha yüksek performans elde edilmiştir ve sınıflar arasındaki farklılaşma azalmıştır.



Şekil 5.3. XGBoost ile (a) Ana veri Seti (b) vs13 veri seti kullanılarak elde edilen ROC Eğrileri ve AUC skorları

Şekil 5.3’de görüldüğü üzere, XGBoost algoritmasıyla elde edilen sonuçlara göre, "Karın ağrısı" sınıfı dışında diğer tüm sınıflarda daha yüksek AUC skoru elde edilmiştir ve sınıflar arasındaki farklılaşma ana veri setine kıyasla daha düşüktür. İki veri setinde de tüm sınıfların eğrileri diyagonal çizgiden pozitif yönde uzaklaşmıştır.

5.2. k-NN ALGORİTMASI İLE ELDE EDİLEN BULGULAR

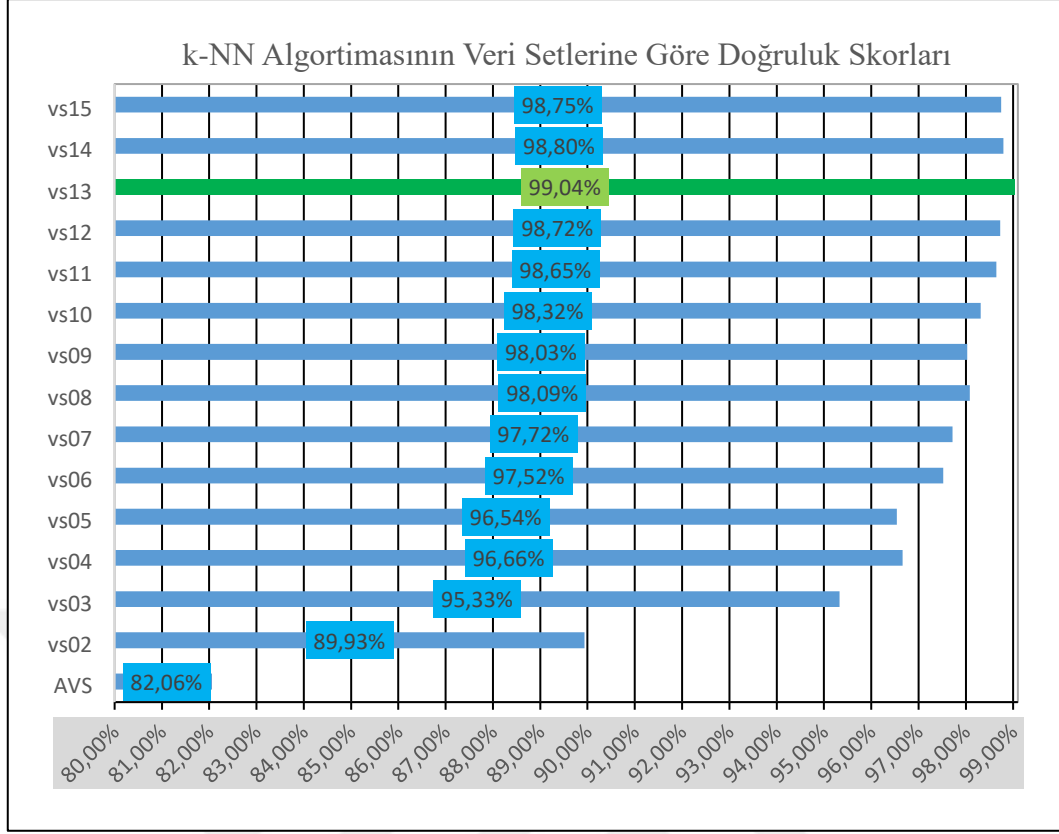
Bu kısımda, k-NN algoritması ile elde edilen bulgular yer almaktadır.

Şekil 5.4’de, oluşturulan veri setleri üzerinde k-NN algoritması ile elde edilen doğruluk skorlarının dağılımı verilmiştir.

Çizelge 5.2’de, k-NN algoritması ile ana veri seti ve en yüksek performans skoru alınan veri setinin karşılaştırılması adına veri setleri üzerinde elde edilen doğruluk, duyarlılık, kesinlik ve f1 skorlarının 30 kez hesaplanan değerleri arasından en yüksek, en düşük ve ortalama değerlerine yer verilmiştir.

Şekil 5.5’te k-NN algoritması için ana veri seti ve en yüksek performans skoru elde edilen veri seti kullanılarak oluşturulan Karışıklık Matrisleri yer almaktadır. Bu Karışıklık Matrisleri, sınıflar arasındaki farklılaşma veya benzerliği ve bu farkın karşılaştırılan veri setlerine bağlı olarak nasıl değiştiğini gözlemlemek için kullanılmıştır.

Son olarak Şekil 5.6’de K-NN algoritması için ana veri seti ve en yüksek performans skoru alınan veri seti ile oluşturulan grafikler verilmiştir. Bu grafikler sınıfların ROC eğrilerini ve AUC skorlarını göstermektedir. Sınıfların farklılaşma veya benzerliğini görmek ve sınıflandırma performansının veri setine göre değişimini gözlemlemek adına kullanılmıştır.



Şekil 5.4. Tüm veri setleri üzerinde k-NN ile hesaplanan doğruluk skorları

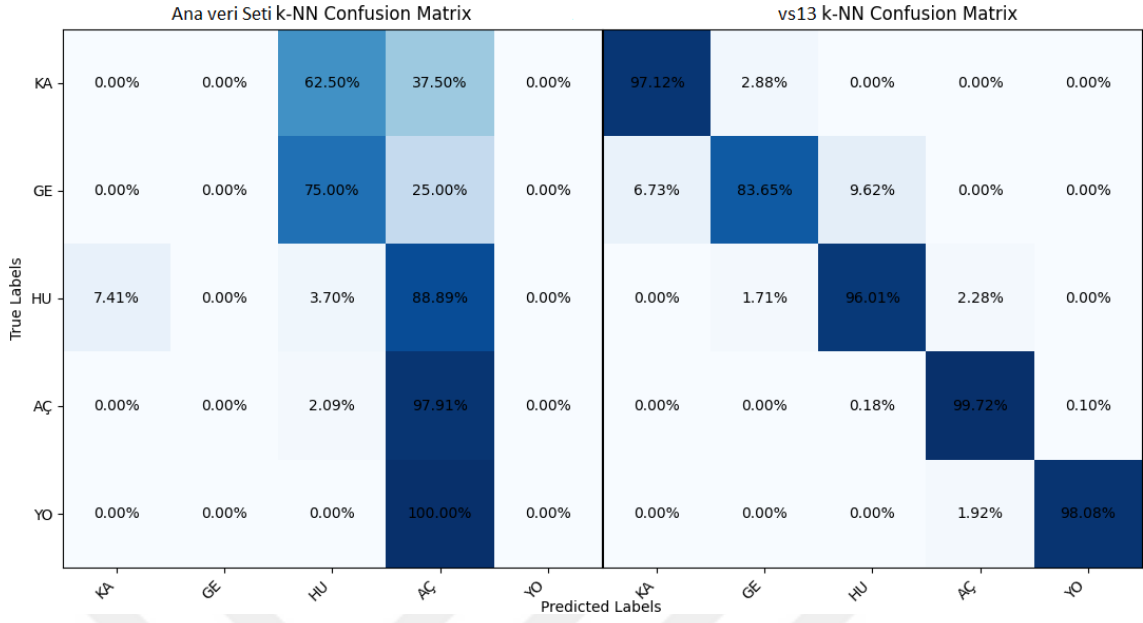
Şekil 5.4’de görülmektedir ki k-NN algoritmasıyla elde edilen en düşük doğruluk skoru ana veri seti kullanılarak bulunmuştur. En yüksek doğruluk skoru ise verilerin 13 parçaya bölüldüğü vs13 veri seti kullanılarak elde edilmiştir.

Çizelge 5.2. İki veri seti üzerinde k-NN ile performans skorlarının karşılaştırılması

k-NN	Ana Veri Seti			vs13		
	En Yüksek	En Düşük	Ortalama	En Yüksek	En Düşük	Ortalama
Doğruluk ↑	82,06%	82,06%	82,06%	99,04%	99,04%	99,04%
Duyarlılık ↑	82,06%	82,06%	82,06%	99,04%	99,04%	99,04%
Kesinlik ↑	72,94%	72,94%	72,94%	99,04%	99,04%	99,04%
F1 ↑	77,23%	77,23%	77,23%	99,04%	99,04%	99,04%

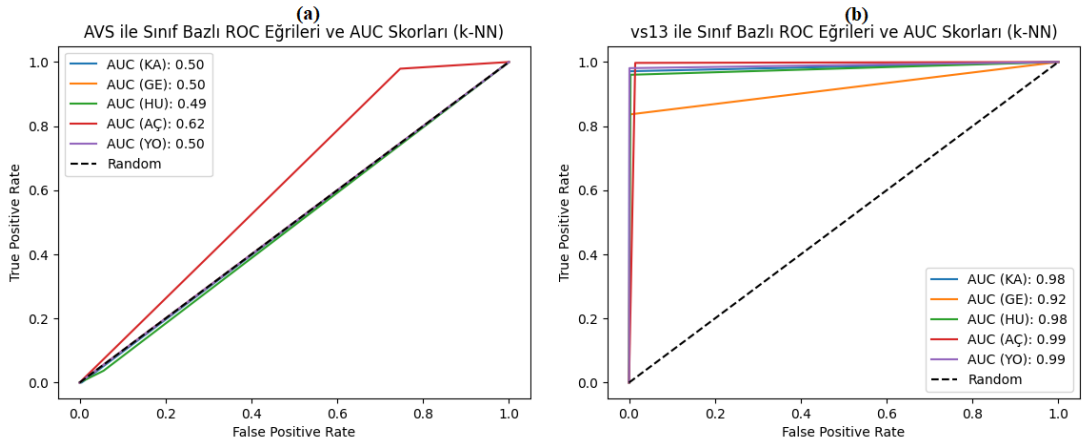
Çizelge 5.2’de vs13 veri seti ve ana veri seti kullanılarak hesaplanan k-NN algoritması performans skorları incelendiğinde vs13 veri seti ile her skorda ortalama %20,46

yükselme olduğu hesaplanmıştır.



Şekil 5.5. İki veri seti üzerinde k-NN ile oluşturulan Karışıklık Matrislerinin karşılaştırması

Şekil 5.5’te görülmektedir ki k-NN algoritması ile tüm sınıflarda daha yüksek performans elde edilmiştir ve sınıflar arasındaki farklılaşma azalmıştır.



Şekil 5.6. k-NN ile (a) Ana veri Seti (b) vs13 veri seti kullanılarak elde edilen ROC Eğrileri ve AUC skorları

Şekil 5.6’da görülmektedir ki “Huzursuzluk” sınıfı k-NN algoritması ile ana veri seti üzerindeki ROC grafiğinde diyagonal çizginin altında kalmıştır. vs13 veri seti ile elde edilen eğriler ise diyagonal çizgiden pozitif yönde uzak ve sınıflar arasındaki farklılaşma ana veri setine kıyasla daha düşüktür.

5.3. MLP ALGORİTASI İLE ELDE EDİLEN BULGULAR

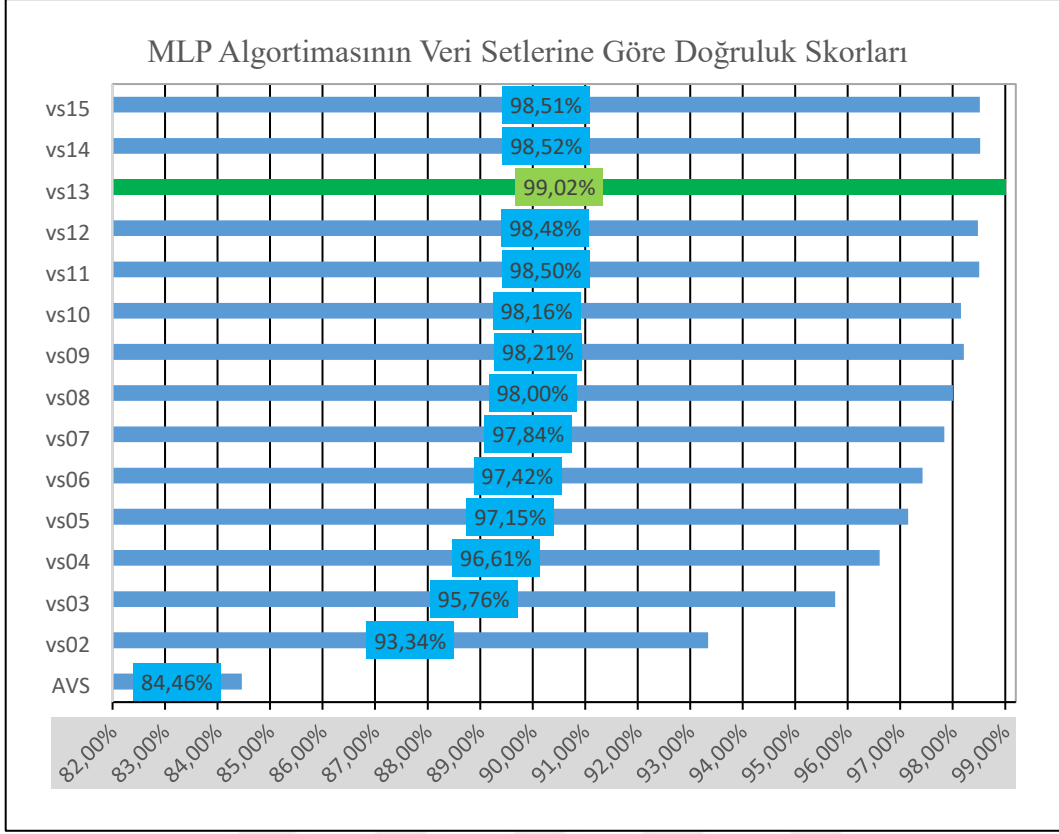
Bu kısımda, MLP algoritması ile elde edilen bulgular yer almaktadır.

Şekil 5.7’de oluşturulan veri setleri üzerinde MLP algoritması ile elde edilen doğruluk skorlarının dağılımı verilmiştir.

Çizelge 5.3’de, MLP algoritması ile ana veri seti ve en yüksek performans skoru alınan veri setinin karşılaştırılması adına veri setleri üzerinde elde edilen doğruluk, duyarlılık, kesinlik ve f1 skorlarının 30 kez hesaplanan değerleri arasından en yüksek, en düşük ve ortalama değerlerine yer verilmiştir.

Şekil 5.8’de MLP algoritması için ana veri seti ve en yüksek performans skoru elde edilen veri seti kullanılarak oluşturulan Karışıklık Matrisleri yer almaktadır. Bu Karışıklık Matrisleri, sınıflar arasındaki farklılaşma veya benzerliği ve bu farkın karşılaştırılan veri setlerine bağlı olarak nasıl değiştiğini gözlemlemek için kullanılmıştır.

Son olarak Şekil 5.9’da MLP algoritması için ana veri seti ve en yüksek performans skoru alınan veri seti ile oluşturulan grafikler verilmiştir. Bu grafikler sınıfların ROC eğrilerini ve AUC skorlarını göstermektedir. Sınıfların farklılaşma veya benzerliğini görmek ve sınıflandırma performansının veri setine göre değişimini gözlemlemek adına kullanılmıştır.



Şekil 5.7. Tüm veri setleri üzerinde MLP ile hesaplanan doğruluk skorları

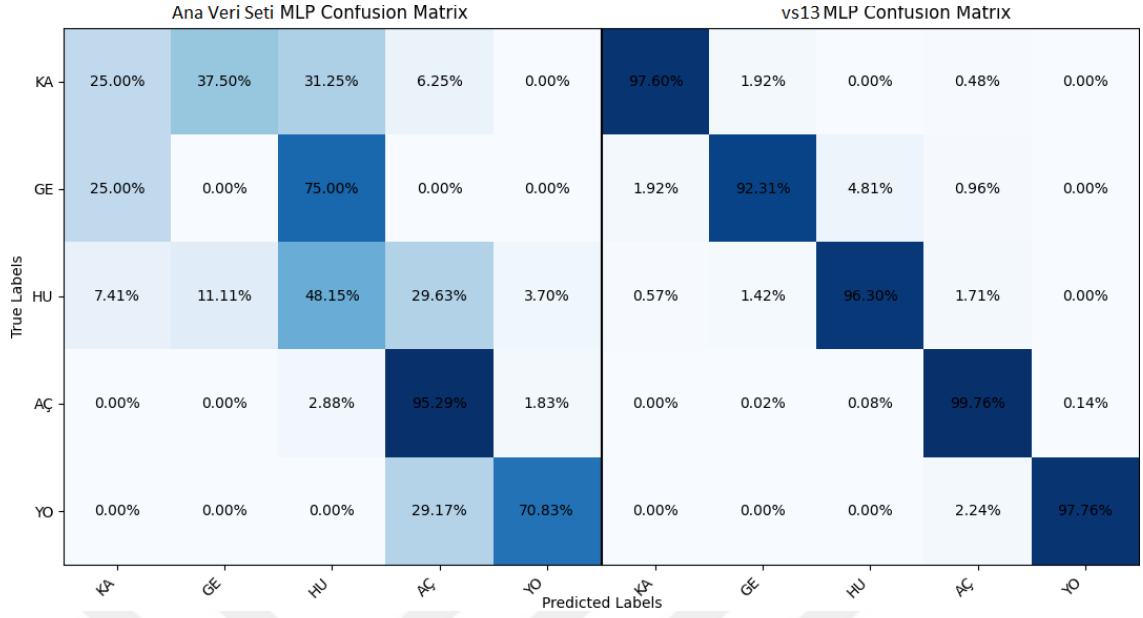
Şekil 5.7’de görülmektedir ki MLP algoritmasıyla elde edilen en düşük doğruluk skoru ana veri seti kullanılarak bulunmuştur. Diğer taraftan, en yüksek doğruluk skoru ise verilerin 13 parçaya bölüdüğü vs13 veri seti kullanılarak elde edilmiştir.

Çizelge 5.3. İki veri seti üzerinde MLP ile performans skorlarının karşılaştırılması

MLP	Veriseti1			vs13		
	En Yüksek	En Düşük	Ortalama	En Yüksek	En Düşük	Ortalama
Doğruluk ↑	88,62%	84,68%	86,71%	99,39%	98,62%	99,02%
Duyarlılık ↑	88,62%	84,68%	86,71%	99,39%	98,62%	99,02%
Kesinlik ↑	89,03%	84,96%	86,84%	99,40%	98,63%	99,02%
F1 ↑	88,48%	84,71%	86,66%	99,39%	98,62%	99,02%

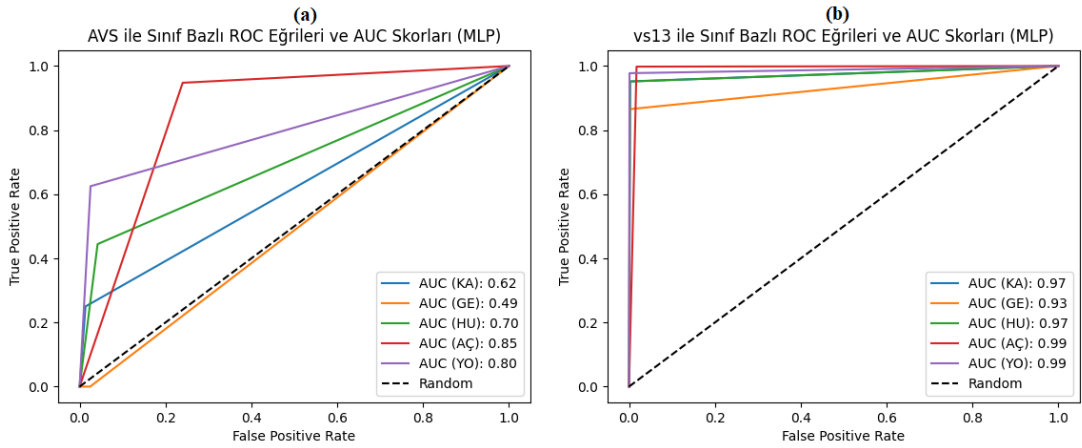
Çizelge 5.3’de vs13 veri seti ve ana veri seti kullanılarak hesaplanan MLP algoritması performans skorları incelendiğinde vs13 veri seti ile her skorda ortalama %12,29

yükselme olduğu hesaplanmıştır.



Şekil 5.8. İki veri seti üzerinde MLP ile oluşturulan Karışıklık Matrislerinin karşılaştırması

Şekil 5.8’de görülmektedir ki MLP algoritması ile tüm sınıflarda daha yüksek performans elde edilmiştir ve sınıflar arasındaki farklılaşma azalmıştır.



Şekil 5.9. MLP ile (a) Ana veri Seti (b) vs13 veri seti kullanılarak elde edilen ROC Eğrileri ve AUC skorları

Şekil 5.9’da görülmektedir ki en az veri sayısına sahip “Geçirme” sınıfı MLP algoritması ile ana veri seti üzerindeki ROC grafiğinde diyagonal çizginin altında kalmıştır. vs13 veri seti ile elde edilen eğriler ise diyagonal çizgiden pozitif yönde uzak ve sınıflar arasındaki farklılaşma ana veri setine kıyasla daha düşüktür.

5.4. RF ALGORİTMASI İLE ELDE EDİLEN BULGULAR

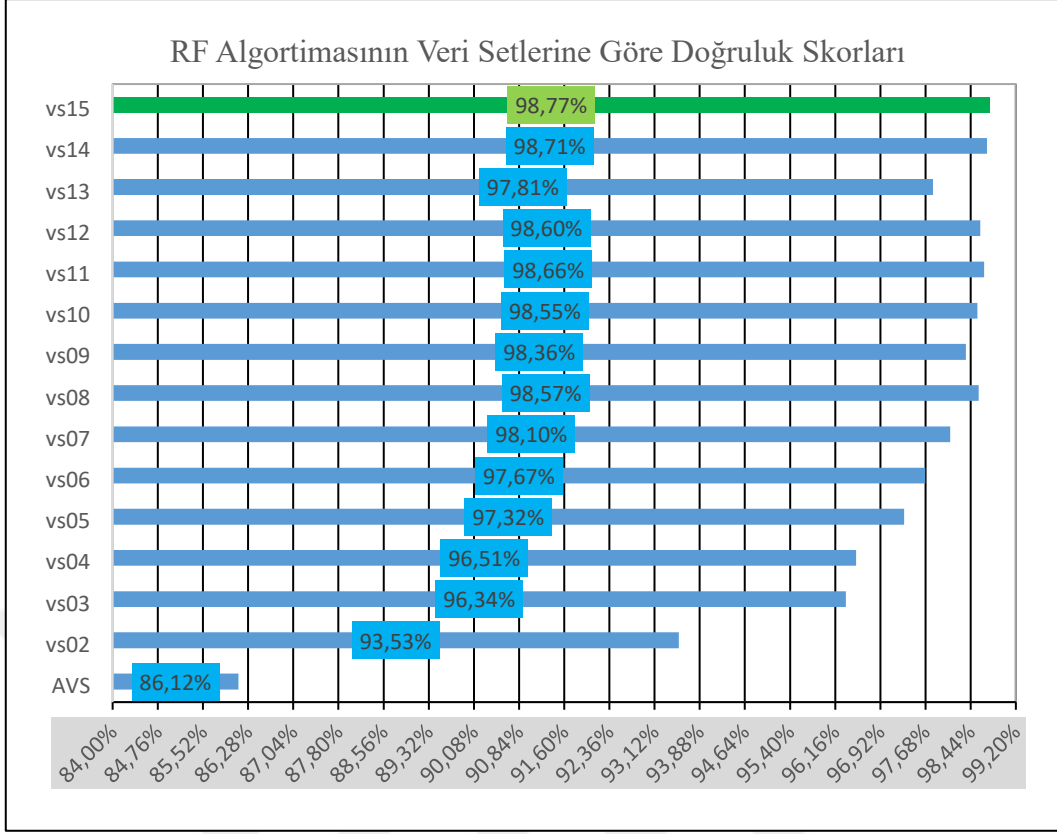
Bu kısımda, RF algoritması ile elde edilen bulgular yer almaktadır.

Şekil 5.10'da oluşturulan veri setleri üzerinde RF algoritması ile elde edilen doğruluk skorlarının dağılımı verilmiştir.

Çizelge 5.4'de, RF algoritması ile ana veri seti ve en yüksek performans skoru alınan veri setinin karşılaştırılması adına veri setleri üzerinde elde edilen doğruluk, duyarlılık, kesinlik ve f1 skorlarının 30 kez hesaplanan değerleri arasından en yüksek, en düşük ve ortalama değerlerine yer verilmiştir.

Şekil 5.11'de RF algoritması için ana veri seti ve en yüksek performans skoru elde edilen veri seti kullanılarak oluşturulan Karışıklık Matrisleri yer almaktadır. Bu Karışıklık Matrisleri, sınıflar arasındaki farklılaşma veya benzerliği ve bu farkın karşılaştırılan veri setlerine bağlı olarak nasıl değiştiğini gözlemlemek için kullanılmıştır.

Son olarak Şekil 5.12'de RF algoritması için ana veri seti ve en yüksek performans skoru alınan veri seti ile oluşturulan grafikler verilmiştir. Bu grafikler sınıfların ROC eğrilerini ve AUC skorlarını göstermektedir. Sınıfların farklılaşma veya benzerliğini görmek ve sınıflandırma performansının veri setine göre değişimini gözlemlemek adına kullanılmıştır.



Şekil 5.10. Tüm veri setleri üzerinde RF ile hesaplanan doğruluk skorları

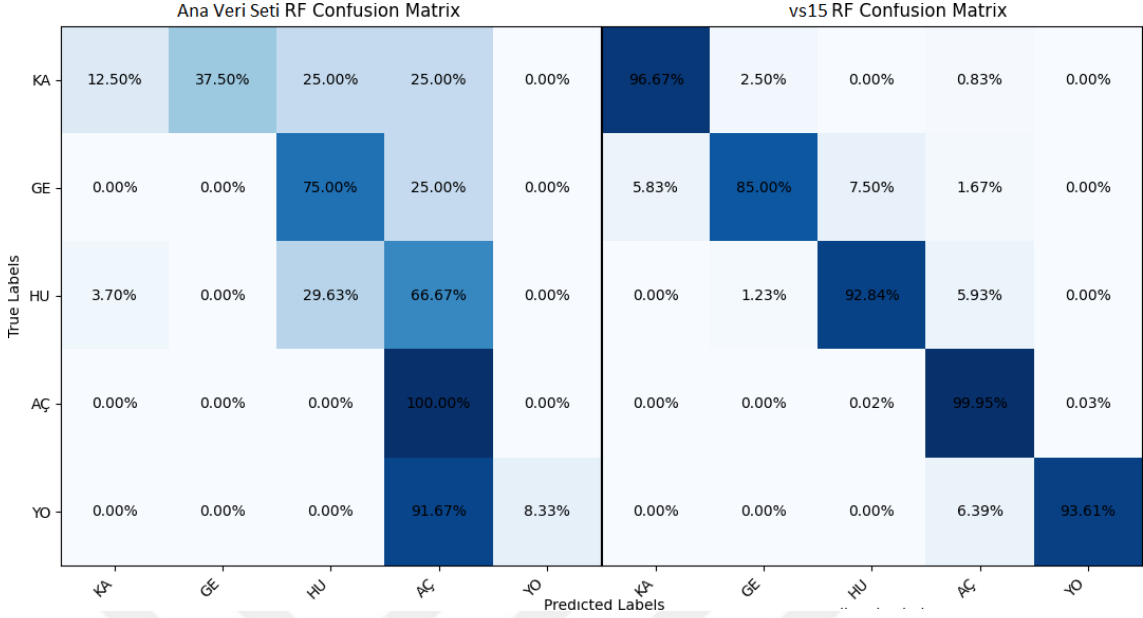
Şekil 5.10’da görülmektedir ki RF algoritmasıyla elde edilen en düşük doğruluk skoru ana veri seti kullanılarak bulunmuştur. Diğer taraftan, en yüksek doğruluk skoru ise verilerin 15 parçaya bölüdüğü vs15 veri seti kullanılarak elde edilmiştir.

Çizelge 5.4. İki veri seti üzerinde RF ile performans skorlarının karşılaştırılması

RF	Ana Veri Seti			vs15		
	En Yüksek	En Düşük	Ortalama	En Yüksek	En Düşük	Ortalama
Doğruluk ↑	86,87%	85,56%	86,16%	98,89%	98,66%	98,77%
Duyarlılık ↑	86,87%	85,56%	86,16%	98,89%	98,66%	98,77%
Kesinlik ↑	85,79%	78,44%	83,01%	98,88%	98,41%	98,74%
F1 ↑	83,74%	81,24%	82,47%	98,87%	98,64%	98,75%

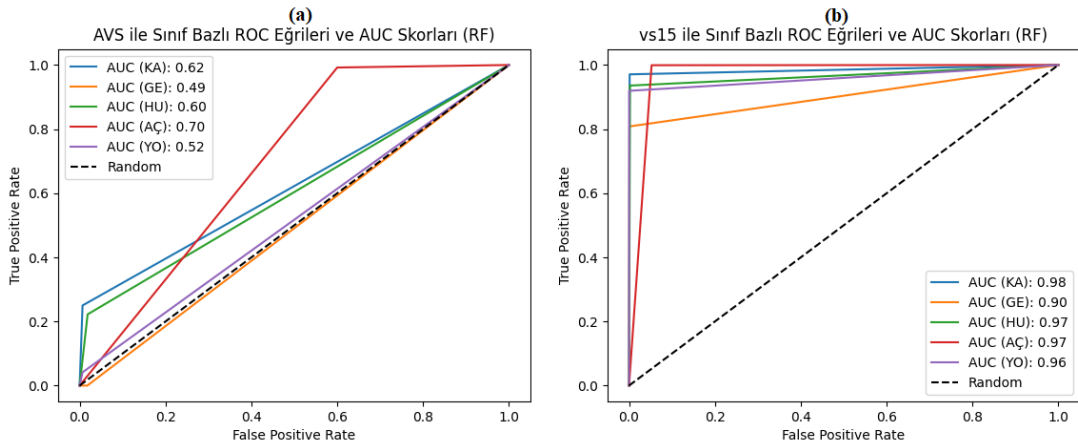
Çizelge 5.4’de vs15 veri seti ve ana veri seti kullanılarak hesaplanan RF algoritması performans skorları incelendiğinde vs15 veri seti ile her skorda ortalama %14,30

yükselme olduğu hesaplanmıştır.



Şekil 5.11. İki veri seti üzerinde RF ile oluşturulan Karışıklık Matrislerinin karşılaştırması

Şekil 5.11’de görülmektedir ki RF algoritması “Açlık” sınıfı dışında diğer tüm sınıflarda daha yüksek performans elde edilmiştir ve sınıflar arasındaki farklılaşma azalmıştır.



Şekil 5.12. RF ile (a) Ana veri Seti (b) vs15 veri seti kullanılarak elde edilen ROC Eğrileri ve AUC skorları

Şekil 5.12’de görülmektedir ki en az veri sayısına sahip “Geçirme” sınıfı RF algoritması ile ana veri seti üzerindeki ROC grafiğinde diyagonal çizginin altında kalmıştır. vs15 veri seti ile elde edilen eğriler ise diyagonal çizgiden pozitif yönde uzak ve sınıflar arasındaki farklılaşma ana veri setine kıyasla daha düşüktür.

5.5. LR ALGORİTMASI İLE ELDE EDİLEN BULGULAR

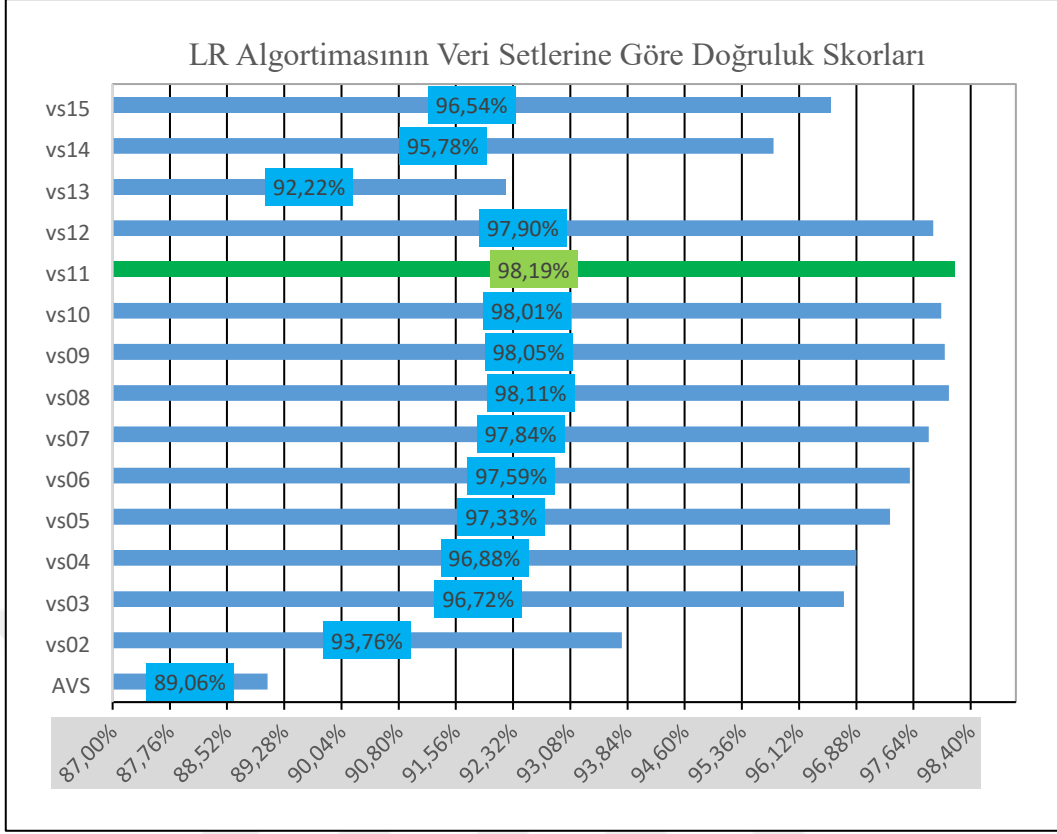
Bu kısımda, LR algoritması ile elde edilen bulgular yer almaktadır.

Şekil 5.13’de oluşturulan veri setleri üzerinde LR algoritması ile elde edilen doğruluk skorlarının dağılımı verilmiştir.

Çizelge 5.5’de, LR algoritması ile ana veri seti ve en yüksek performans skoru alınan veri setinin karşılaştırılması adına veri setleri üzerinde elde edilen doğruluk, duyarlılık, kesinlik ve f1 skorlarının 30 kez hesaplanan değerleri arasından en yüksek, en düşük ve ortalama değerlerine yer verilmiştir.

Şekil 5.14’te LR algoritması için ana veri seti ve en yüksek performans skoru elde edilen veri seti kullanılarak oluşturulan Karışıklık Matrisleri yer almaktadır. Bu Karışıklık Matrisleri, sınıflar arasındaki farklılaşma veya benzerliği ve bu farkın karşılaştırılan veri setlerine bağlı olarak nasıl değiştiğini gözlemlemek için kullanılmıştır.

Son olarak Şekil 5.15’te LR algoritması için ana veri seti ve en yüksek performans skoru alınan veri seti ile oluşturulan grafikler verilmiştir. Bu grafikler sınıfların ROC eğrilerini ve AUC skorlarını göstermektedir. Sınıfların farklılaşma veya benzerliğini görmek ve sınıflandırma performansının veri setine göre değişimini gözlemlemek adına kullanılmıştır.



Şekil 5.13. Tüm veri setleri üzerinde LR ile hesaplanan doğruluk skorları

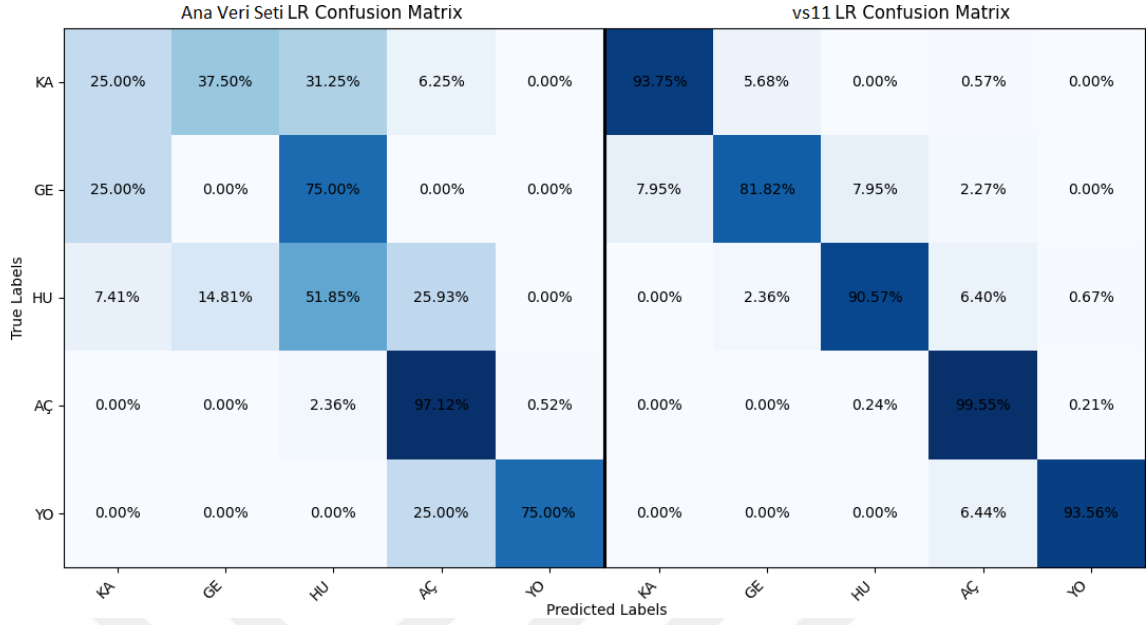
Şekil 5.13’de görülmektedir ki LR algoritması, en düşük doğruluk skorunu ana veri seti ile en yüksek doğruluk skorunu ise verilerin 11 parçaya bölüdüğü vs11 veri seti ile elde etmiştir.

Çizelge 5.5. İki veri seti üzerinde LR ile performans skorlarının karşılaştırılması

LR	Ana Veri Seti			vs11		
	En Yüksek	En Düşük	Ortalama	En Yüksek	En Düşük	Ortalama
Doğruluk ↑	89,06%	89,06%	89,06%	98,19%	98,19%	98,19%
Duyarlılık ↑	89,46%	89,46%	89,46%	98,18%	98,18%	98,18%
Kesinlik ↑	89,06%	89,06%	89,06%	98,19%	98,19%	98,19%
F1 ↑	89,04%	89,04%	89,04%	98,18%	98,18%	98,18%

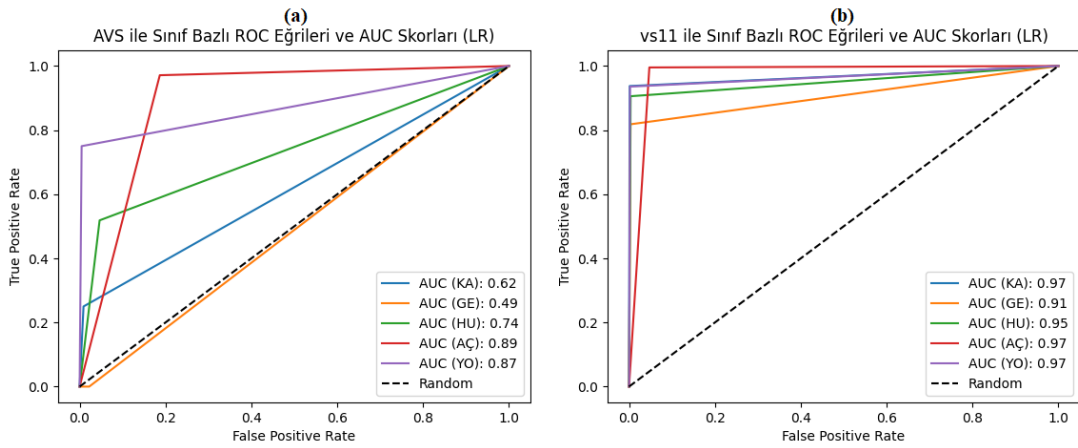
Çizelge 5.5’de vs11 veri seti ve ana veri seti kullanılarak hesaplanan LR algoritması performans skorları incelendiğinde vs11 veri seti ile her skorda ortalama %9,03 yükselme

olduğu hesaplanmıştır.



Şekil 5.14. İki veri seti üzerinde LR ile oluşturulan Karışıklık Matrislerinin karşılaştırması

Şekil 5.14’de görülmektedir ki LR algoritması tüm sınıflarda daha yüksek performans elde edilmiştir ve sınıflar arasındaki farklılaşma azalmıştır.



Şekil 5.15. LR ile (a) Ana veri Seti (b) vs11 veri seti kullanılarak elde edilen ROC Eğrileri ve AUC skorları

Şekil 5.15’de görülmektedir ki en az veri sayısına sahip “Geçirme” sınıfı LR algoritması ile ana veri seti üzerindeki ROC grafiğinde diyagonal çizginin altında kalmıştır. vs11 veri seti ile elde edilen eğriler ise diyagonal çizgiden pozitif yönde uzak ve sınıflar arasındaki farklılaşma ana veri setine kıyasla daha düşüktür.

5.6. SVM ALGORİTMASI İLE ELDE EDİLEN BULGULAR

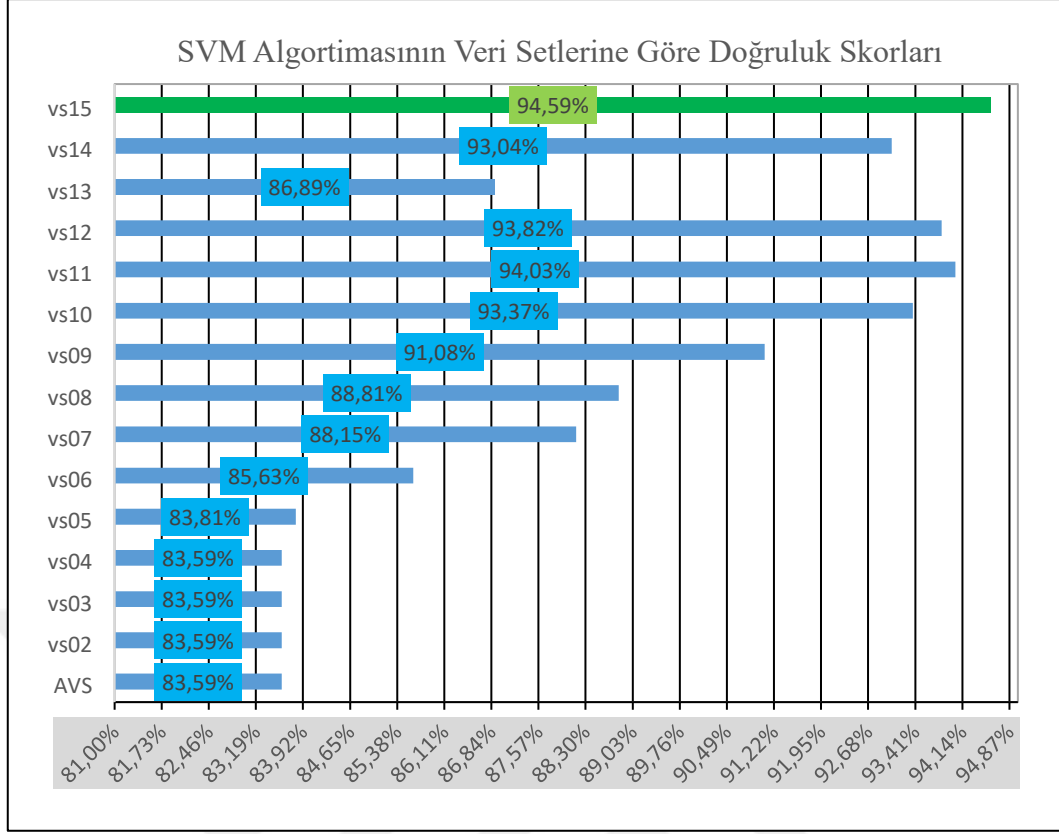
Bu kısımda, SVM algoritması ile elde edilen bulgular yer almaktadır.

Şekil 5.16'da oluşturulan veri setleri üzerinde SVM algoritması ile elde edilen doğruluk skorlarının dağılımı verilmiştir.

Çizelge 5.6'da, SVM algoritması ile ana veri seti ve en yüksek performans skoru alınan veri setinin karşılaştırılması adına veri setleri üzerinde elde edilen doğruluk, duyarlılık, kesinlik ve f1 skorlarının 30 kez hesaplanan değerleri arasından en yüksek, en düşük ve ortalama değerlerine yer verilmiştir.

Şekil 5.17'de SVM algoritması için ana veri seti ve en yüksek performans skoru elde edilen veri seti kullanılarak oluşturulan Karışıklık Matrisleri yer almaktadır. Bu Karışıklık Matrisleri, sınıflar arasındaki farklılaşma veya benzerliği ve bu farkın karşılaştırılan veri setlerine bağlı olarak nasıl değiştiğini gözlemlemek için kullanılmıştır.

Son olarak Şekil 5.18'de SVM algoritması için ana veri seti ve en yüksek performans skoru alınan veri seti ile oluşturulan grafikler verilmiştir. Bu grafikler sınıfların ROC eğrilerini ve AUC skorlarını göstermektedir. Sınıfların farklılaşma veya benzerliğini görmek ve sınıflandırma performansının veri setine göre değişimini gözlemlemek adına kullanılmıştır.



Şekil 5.16. Tüm veri setleri üzerinde SVM ile hesaplanan doğruluk skorları

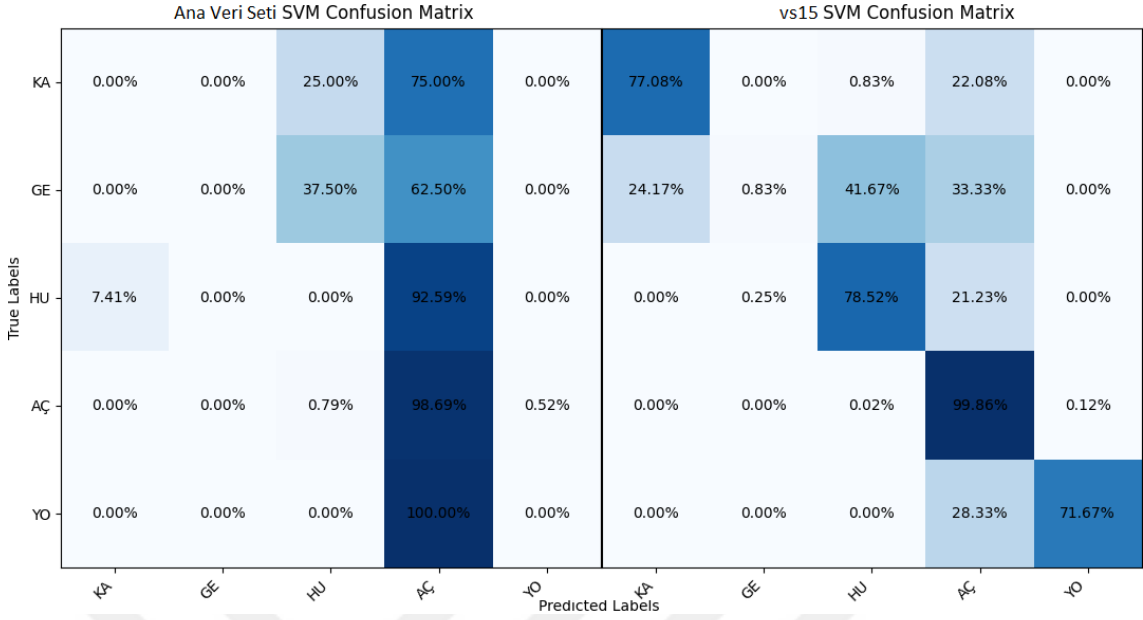
Şekil 5.16’de görülmektedir ki SVM algoritmasıyla elde edilen en düşük doğruluk skoru ana veri seti, vs02, vs03 ve vs04 kullanılarak bulunmuştur. Diğer taraftan, en yüksek doğruluk skoru ise vs15 veri seti kullanılarak elde edilmiştir.

Çizelge 5.6. İki veri seti üzerinde SVM performans skorlarının karşılaştırılması

SVM	Ana Veri Seti			vs15		
	En Yüksek	En Düşük	Ortalama	En Yüksek	En Düşük	Ortalama
Doğruluk ↑	83,59%	83,59%	83,59%	94,59%	94,59%	94,59%
Duyarlılık ↑	83,59%	83,59%	83,59%	94,59%	94,59%	94,59%
Kesinlik ↑	69,87%	69,87%	69,87%	93,75%	93,75%	93,75%
F1 ↑	76,12%	76,12%	76,12%	93,59%	93,59%	93,59%

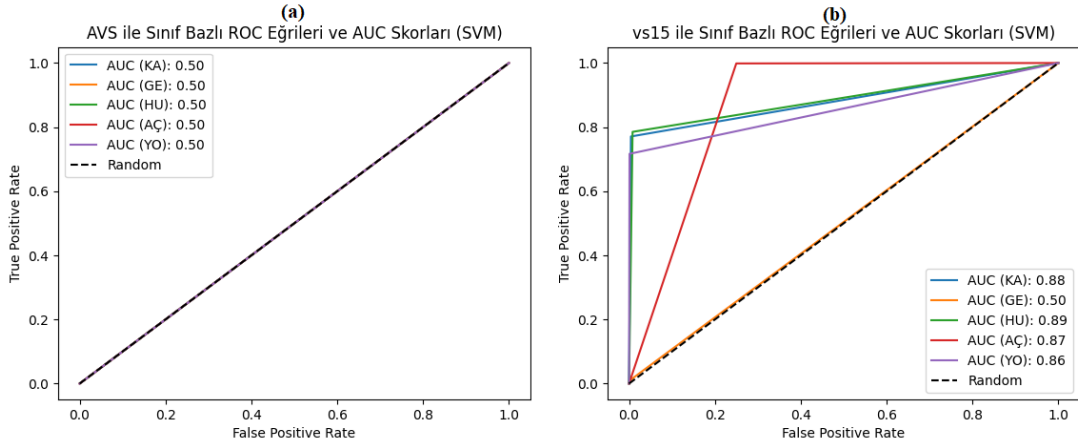
Çizelge 5.6 vs15 veri seti ve ana veri seti kullanılarak hesaplanan SVM algoritması performans skorları incelendiğinde vs15 veri seti ile her skorda ortalama %15,84

yükselme olduğu hesaplanmıştır.



Şekil 5.17. İki veri seti üzerinde SVM ile oluşturulan Karışıklık Matrislerinin karşılaştırması

Şekil 5.17’de görülmektedir ki SVM algoritması tüm sınıflarda daha yüksek performans elde edilmiştir ve sınıflar arasındaki farklılaşma nispeten azalmıştır. En az veri sayısına sahip olan “Geçirme” vs15 veri setinde de düşük kalmıştır.

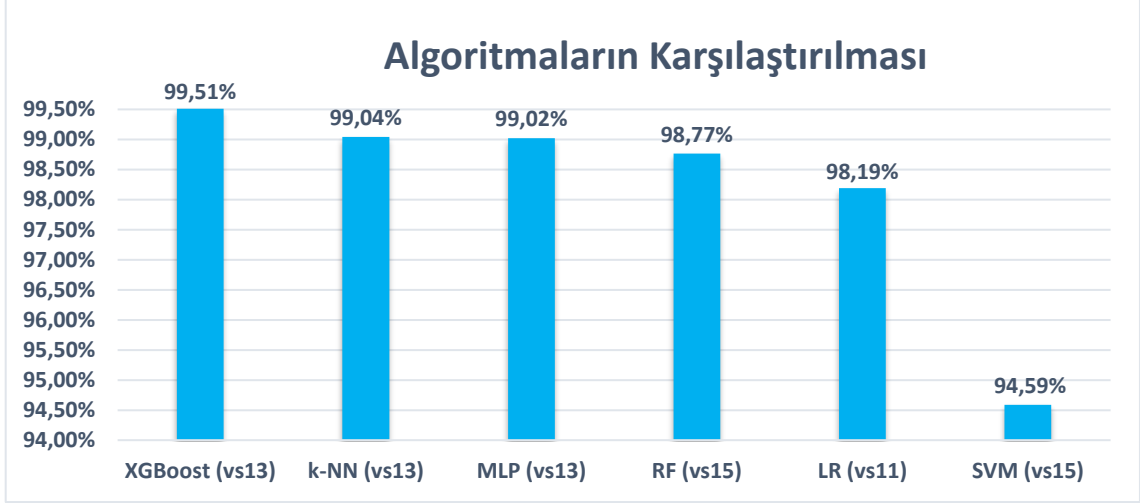


Şekil 5.18. SVM ile (a) Ana veri Seti (b) vs15 veri seti kullanılarak elde edilen ROC Eğrileri ve AUC skorları

Şekil 5.18’de görülmektedir ki SVM algoritması ile ana veri setinde ROC eğrileri diyalagonal çizgide kalmıştır. vs15 veri setinde “geçirme” sınıfı dışında tüm sınıflarda diyalagonal çizgiden uzak ROC eğrileri oluşmuş ve AUC skorları yükselmiştir.

5.7. ALGORİTMALARIN KARŞILAŞTIRILMASI

Bu kısımda XGBoost, k-NN, MLP, RF, LR ve SVM algoritmaları aldıkları en yüksek skorlar ve bu skoru elde ettikleri veri setleri verilmiş ve algoritmalar birbirleri ile karşılaştırılmıştır.



Şekil 5.19. Kullanılan algoritmaların karşılaştırılması

Şekil 5.19’de görüldüğü üzere XGBoost algoritması verilerin 13 eşit parçaya ayrıldığı vs13 veri seti ile en yüksek performans değerini elde etmiştir. Ayrıca k-NN ve MLP algoritmaları da yine vs13 veri seti ile performans değeri olarak %99’un üzerine çıkmıştır.

6. TARTIŞMA VE SONUÇ

Literatürde donate-a-cry veri seti ile yapılan çalışmalar göz önünde bulundurulduğunda, en yüksek performans değerini elde eden çalışma Rezaee ve arkadaşları tarafından DeepSVM algoritması kullanılarak elde edilmiştir ve %98,34 performans değeri rapor edilmiştir [2]. Bu tez çalışmasında ise XGBoost algoritması kullanılarak %99,51 performans değeri elde edilmiş ve donate-a-cry veri seti üzerinde yapılan önceki çalışmaların üzerine çıkılarak en yüksek başarı oranına ulaşılmıştır.

Bu çalışmada XGBoost, k-NN, MLP, RF, LR ve SVM gibi farklı algoritmalar kullanılmaktadır. Ancak, bu algoritmaların performansları farklılık gösterebilir. XGBoost algoritmasının daha iyi sonuçlar vermesinin nedenlerinden biri, çoklu öğrenme ve hataları düzeltme yetenekleridir. XGBoost, birden fazla karar ağacını birleştirerek çalışır. Bu ağaçlar, sırayla önceki ağaçların hatalarını düzeltmeye çalışır. Bu özellik, modelin veri setinin karmaşıklığını ve gürültüsünü daha iyi anlamasına yardımcı olur. Aynı zamanda XGBoost, aşırı uyuma karşı dirençlidir ve daha dengeli tahminler yapabilir. Bu sayede, özellikle büyük ve karmaşık veri setleri üzerinde daha istikrarlı sonuçlar elde edebilir. Diğer algoritmalar da etkili olabilir, ancak XGBoost'un çoklu öğrenme ve hataları düzeltme özellikleri, onu daha iyi sonuçlar elde etmek için güçlü bir aday yapar.

Veri analizi süreçlerinde, algoritmaların farklı veri yapılarına tepkisi önemli bir rol oynar. Özellikle karmaşık veri yapıları söz konusu olduğunda, algoritmaların performansı farklılık gösterebilir. İncelediğimiz k-nn ve MLP algoritmaları, veri setinizdeki düzensiz dağılımlara yüksek doğruluk skorları ile tepki verirken, Random Forest (RF) algoritmasının daha düşük sonuç vermesi dikkat çekici olabilir. Bu farklılığın arkasında yatan sebepler arasında, veri setinizin düzensiz dağılımı yer almaktadır.

k-NN ve MLP algoritmaları, özellikle düşük boyutlu veri setlerinde ve düzensiz veri dağılımlarında etkili olabilir. Bu algoritmalar, veri noktalarının benzerliklerine dayalı olarak tahminlerde bulunur. Bu nedenle, düzensiz dağılımlara sahip veri setlerinde daha iyi performans gösterebilirler. Öte yandan, Random Forest algoritması da çoklu öğrenme (ensemble) prensibiyle çalışmasına rağmen, karmaşık veri yapılarındaki düzensizliklere karşı k-NN ve MLP kadar hassas tepkiler veremeyebilir. RF'nin bir çoklu öğrenme

algoritması olması ve birden fazla ağacın birleştirilmesi, bazı durumlarda düzensiz veri dağılımlarının model tarafından tam olarak yakalanmasını engelleyebilir.

Sonuç olarak, k-NN ve MLP algoritmalarının düzensiz veri dağılımlarına daha iyi adapte olması ve bu nedenle yüksek doğruluk skorları elde etmesi, veri analizinde algoritma seçiminin veri yapısıyla uyumlu olmasının önemini göstermektedir. Karmaşık veri yapılarına sahip veri setlerinde, algoritma seçerken veri dağılımının etkisi göz önünde bulundurulmalıdır.

Donate-a-cry veri setinin iki önemli dezavantajı bulunmaktadır. İlk dezavantaj, verilerin sınıflar arasında düzensiz bir şekilde dağılım göstermesidir. İkinci dezavantaj ise veri sayısının düşük olmasıdır. Bu çalışmada verileri eşit parçalara bölerek bu dezavantajların üstesinden gelinmesi amaçlanmıştır. Bu amaç doğrultusunda elde edilen başarının daha açık görülebilmesi için k-NN algoritması ile elde edilen bulgular örnek verilebilir. K-NN için Ana veri seti ile elde edilen sonuçlarda düşük performans skorları ve sınıflar arasındaki farklılaşmanın yüksek olduğu, vs13 veri seti ile elde edilen sonuçlarda ise her sınıfta yüksek performans skorları ve sınıflar arasındaki farklılaşmanın daha az olduğu gözlemlenmiştir. Bu durum, veri artırma yönteminin verilerin sınıflara düzensiz dağılımının olumsuz etkisini azalttığını göstermektedir. Yine k-NN ile vs13 veri seti ile elde edilen performans skorlarının, ana veri seti ile elde edilen performans skorlarından ortalama %20,47 daha yüksek olduğu da bulgularda görülmektedir. Bu durum, veri sayısının az olmasının olumsuz etkisinin veri artırma yöntemi sayesinde azaldığını göstermektedir.

Bulgular kısmında SVM algoritmasının performans skorlarının en iyi veri setinde bile düşük sonuçlar verdiği görülmektedir. Ayrıca kullanılan veri setinin düzensizliğinden kullanılan diğer algoritmalarla göre daha fazla etkilendiği de görülmektedir. Bu durum kullanılan veri artırma yönteminin SVM algoritmasında diğer algoritmalar kadar etkili olmadığını göstermektedir.

Bulgulara göre her bir algoritmanın ana veri setinde en düşük performans değerlerini gösterdiği görülmektedir. Bu sonuç, çalışmada kullanılan veri artırma yönteminin, tüm algoritmalar için performans değerlerini olumlu yönde etkilediğini ortaya koymaktadır.

Bulgular kısmında verilen ROC eğrileri incelendiğinde ana veri setinde tüm sınıflar genel olarak diyagonal çizgiye daha yakın olduğu görülmektedir. Bu durum algoritmanın sınıflandırma işlemini rasgele yaptığını ve sınıfları ayırt etme konusunda yetersiz

kaldığını gösterir. Öte yandan algoritmaların en yüksek performans gösterdiği veri setinden elde edilen ROC eğrilerinin neredeyse her sınıfta sol üst köşeye yaklaştığı ve AUC skorunun arttığı görülmektedir. Bu durum, yeni veri seti ile oluşturulan modelin sınıflandırma başarısının arttığını işaret etmektedir. Özetle ana veri setinde algoritmaların sınıflandırma başarısı veri artırma ile oluşturulan sınıflardaki başarısından daha düşüktür. Performans skorlarına dayanarak, bebeklerin belirli bir sebeple ağlama seslerinin küçük parçalarında bile karakteristik özellikler sergilediği güçlü bir varsayım olarak değerlendirilmiştir.

Bu tez çalışması, ses sınıflandırmasıyla ilgili yapılan araştırmalara katkı sağlamayı hedefleyen bir çalışmadır. Odaklanılan konu, veri artırma yöntemlerinin kullanılmasıdır. Veri artırma yöntemleri, mevcut veri setini çeşitlendirmek ve sınıf dağılımını dengelemek amacıyla kullanılır.

Gelecekteki çalışmalarda, aynı veri seti üzerinde benzer bir veri artırma yöntemi uygulanarak verilerin sınıflara daha düzenli bir şekilde dağılımının nasıl iyileştirilebileceği incelenebilir. Bu, sınıflandırma performansı, doğruluk oranını artırma ve sınıflar arası farklılaşmayı minimize etme potansiyeline sahip olabilir.

Ayrıca, veri seti dışında daha düzenli bir dağılıma sahip başka veri setleri de kullanılarak sınıflandırma çalışmaları yapılabilir. Bu, genellemeyi artırabilir ve modelin farklı veri kümelerinde nasıl performans gösterdiğini anlamamıza yardımcı olabilir.

Sonuç olarak, bu çalışma veri artırma yöntemlerinin ses sınıflandırması alanındaki uygulamalarına odaklanmaktadır. Gelecekteki çalışmalarda, aynı yöntemle veri setinin düzenlenmesi ve farklı veri setlerinin kullanılmasıyla ilgili çalışmalara ışık tutacaktır.

7. KAYNAKLAR

- [1] T. Ozseven, “Infant cry classification by using different deep neural network models and hand-crafted features”, *Biomedical Signal Processing and Control*, c. 83, s. 104648, May. 2023, doi: 10.1016/j.bspc.2023.104648.
- [2] K. Rezaee, H. Ghayoumi Zadeh, L. Qi, H. Rabiee, ve M. R. Khosravi, “Can you Understand why I am Crying? A Decision-making System for Classifying Infants’ Cry Languages Based on deepSVM Model”, *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, Oca. 2023, doi: 10.1145/3579032.
- [3] S. Khan ve D. S. Neware, “AUTOMATED SYSTEM FOR CRY ANALYSIS AND PAIN ASSESSMENT IN NEONATES AND INFANTS”, 2023.
- [4] H.-N. Ting, Y.-M. Choo, ve A. Ahmad Kamar, “Classification of asphyxia infant cry using hybrid speech features and deep learning models”, *Expert Systems with Applications*, c. 208, s. 118064, Ara. 2022, doi: 10.1016/j.eswa.2022.118064.
- [5] K. Ashwini ve P. M. D. R. Vincent, “A Deep Convolutional Neural Network Based Approach for Effective Neonatal Cry Classification”, *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, c. 15, sy 2, ss. 229-239, Şub. 2022, doi: 10.2174/2666255813999200710135408.
- [6] C.-Y. Chang, S. Bhattacharya, P. M. D. Raj Vincent, K. Lakshmana, ve K. Srinivasan, “An Efficient Classification of Neonates Cry Using Extreme Gradient Boosting-Assisted Grouped-Support-Vector Network”, *Journal of Healthcare Engineering*, c. 2021, s. e7517313, Kas. 2021, doi: 10.1155/2021/7517313.
- [7] A. Mohite ve Prof. D. Jadhav, “Infant Monitoring System for Deaf Parents”, *IJSRCSEIT*, ss. 527-534, Haz. 2021, doi: 10.32628/CSEIT2173163.
- [8] A. K, P. M. D. R. Vincent, K. Srinivasan, ve C.-Y. Chang, “Deep Learning Assisted Neonatal Cry Classification via Support Vector Machine Models”, *Front Public Health*, c. 9, s. 670352, Haz. 2021, doi: 10.3389/fpubh.2021.670352.
- [9] P. Kulkarni, S. Umarani, V. Diwan, V. Korde, ve P. P. Rege, “Child Cry Classification - An Analysis of Features and Models”, içinde *2021 6th International*

- Conference for Convergence in Technology (I2CT)*, Nis. 2021, ss. 1-7. doi: 10.1109/I2CT51068.2021.9418129.
- [10] E. Sutanto, F. Fahmi, W. Shalannanda, ve A. Aridarma, “Cry Recognition for Infant Incubator Monitoring System Based on Internet of Things using Machine Learning”, *International Journal of Intelligent Engineering and Systems*, c. 14, ss. 444-452, Şub. 2021, doi: 10.22266/ijies2021.0228.41.
- [11] C. Ji, T. B. Mudiyansele, Y. Gao, ve Y. Pan, “A review of infant cry analysis and classification”, *EURASIP Journal on Audio, Speech, and Music Processing*, c. 2021, sy 1, s. 8, Şub. 2021, doi: 10.1186/s13636-021-00197-5.
- [12] D. Widhyanti ve D. Juniati, “Classification of Baby Cry Sound Using Higuchi’s Fractal Dimension with K-Nearest Neighbor and Support Vector Machine”, *J. Phys.: Conf. Ser.*, c. 1747, sy 1, s. 012014, Şub. 2021, doi: 10.1088/1742-6596/1747/1/012014.
- [13] L. Jiang, Y. Yi, D. Chen, P. Tan, ve X. Liu, “A novel infant cry recognition system using auditory model-based robust feature and GMM-UBM”, *Concurrency and Computation: Practice and Experience*, c. 33, sy 11, s. e5405, 2021, doi: 10.1002/cpe.5405.
- [14] A. Bashiri ve R. Hosseinkhani, “Infant Crying Classification by Using Genetic Algorithm and Artificial Neural Network”, *Acta Medica Iranica*, ss. 531-539, Ara. 2020, doi: 10.18502/acta.v58i10.4916.
- [15] T. N. Maghfira, T. Basaruddin, ve A. Krisnadi, “Infant cry classification using CNN – RNN”, *J. Phys.: Conf. Ser.*, c. 1528, sy 1, s. 012019, Nis. 2020, doi: 10.1088/1742-6596/1528/1/012019.
- [16] L. Le, A. N. M. H. Kabir, C. Ji, S. Basodi, ve Y. Pan, “Using Transfer Learning, SVM, and Ensemble Classification to Classify Baby Cries Based on Their Spectrogram Images”, içinde *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW)*, Kas. 2019, ss. 106-110. doi: 10.1109/MASSW.2019.00028.
- [17] S. P. Dewi, A. L. Prasasti, ve B. Irawan, “Analysis of LFCC Feature Extraction in Baby Crying Classification using KNN”, içinde *2019 IEEE International Conference on Internet of Things and Intelligence System (IoT&IS)*, Kas. 2019, ss.

86-91. doi: 10.1109/IoTaIS47347.2019.8980389.

- [18] R. I. Tuduce, H. Cucu, ve C. Burileanu, “Why is My Baby Crying? An In-Depth Analysis of Paralinguistic Features and Classical Machine Learning Algorithms for Baby Cry Classification”, içinde *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, Tem. 2018, ss. 1-4. doi: 10.1109/TSP.2018.8441363.
- [19] E. Franti, I. Ispas, ve M. Dascalu, “Testing the Universal Baby Language Hypothesis - Automatic Infant Speech Recognition with CNNs”, içinde *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, Tem. 2018, ss. 1-4. doi: 10.1109/TSP.2018.8441412.
- [20] S. Orlandi, C. A. Reyes Garcia, A. Bandini, G. Donzelli, ve C. Manfredi, “Application of Pattern Recognition Techniques to the Classification of Full-Term and Preterm Infant Cry”, *Journal of Voice*, c. 30, sy 6, ss. 656-663, Kas. 2016, doi: 10.1016/j.jvoice.2015.08.007.
- [21] G. Veres, “donateacry-corpus”. 17 Mayıs 2023. Erişim: 19 Mayıs 2023. [Çevrimiçi]. Erişim adresi: <https://github.com/gveres/donateacry-corpus>
- [22] F. Chollet, *Deep Learning with Python*, First Edition. Shelter Island, New York: Manning, 2017.
- [23] B. Gold, N. Morgan, ve D. Ellis, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*, 2nd Edition | Wiley. Wiley, 2011.
- [24] V. T. Tiwari, “MFCC and its applications in speaker recognition”, *Int. J. Emerg. Technol.*, c. 1, Oca. 2010.
- [25] Taegyun Lim, Keunsung Bae, Chansik Hwang, ve Hyeonguk Lee, “Classification of underwater transient signals using MFCC feature vector”, içinde *2007 9th International Symposium on Signal Processing and Its Applications*, Sharjah, United Arab Emirates: IEEE, Şub. 2007, ss. 1-4. doi: 10.1109/ISSPA.2007.4555521.
- [26] A. Burkov, *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019.
- [27] P. Probst, M. N. Wright, ve A.-L. Boulesteix, “Hyperparameters and tuning strategies for random forest”, *WIREs Data Mining and Knowledge Discovery*, c. 9, sy 3, s. e1301, 2019, doi: 10.1002/widm.1301.

- [28] S. Shalev-Shwartz ve S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge: Cambridge University Press, 2014. doi: 10.1017/CBO9781107298019.
- [29] A. B. Shaik ve S. Srinivasan, “A Brief Survey on Random Forest Ensembles in Classification Model”, içinde *International Conference on Innovative Computing and Communications*, S. Bhattacharyya, A. E. Hassanien, D. Gupta, A. Khanna, ve I. Pan, Ed., içinde *Lecture Notes in Networks and Systems*, vol. 56. Singapore: Springer Singapore, 2019, ss. 253-260. doi: 10.1007/978-981-13-2354-6_27.
- [30] N. Soares, E. Pestana de Aguiar, ve L. Goliatt, *Failure Classification in Electric Switch Machines Using Symbolic Concepts and Computational Intelligence*. 2017. doi: 10.20906/CPS/CILAMCE2017-1021.
- [31] H. Taud ve J. F. Mas, “Multilayer Perceptron (MLP)”, içinde *Geomatic Approaches for Modeling Land Change Scenarios*, Springer, Cham, 2018, ss. 451-455. doi: 10.1007/978-3-319-60801-3_27.
- [32] E. Alpaydin, *Introduction to Machine Learning*, Second edition. Cambridge, Mass: The MIT Press, 2009.
- [33] O. Uludağ ve A. Gürsoy, “Financial Risk Estimation with KNN Classification Algorithm on Determined Financial Ratios”, *European Journal of Science and Technology*, Ara. 2021, doi: 10.31590/ejosat.1001663.
- [34] O. Kramer, “K-Nearest Neighbors”, içinde *Dimensionality Reduction with Unsupervised Nearest Neighbors*, O. Kramer, Ed., içinde *Intelligent Systems Reference Library*. Berlin, Heidelberg: Springer, 2013, ss. 13-23. doi: 10.1007/978-3-642-38652-7_2.
- [35] T. Chen ve C. Guestrin, “XGBoost: A Scalable Tree Boosting System”, içinde *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, içinde *KDD '16*. New York, NY, USA: Association for Computing Machinery, Ağu. 2016, ss. 785-794. doi: 10.1145/2939672.2939785.
- [36] D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto, ve S. Ridella, “The ‘K’ in K-fold Cross Validation”, *Computational Intelligence*, 2012.
- [37] D. Berrar, “Cross-Validation”, 2018. doi: 10.1016/B978-0-12-809633-8.20349-X.
- [38] H. M ve S. M.N, “A Review on Evaluation Metrics for Data Classification

- Evaluations”, *IJDKP*, c. 5, sy 2, ss. 01-11, Mar. 2015, doi: 10.5121/ijdkp.2015.5201.
- [39] V. Velardo, “Deep Learning for Audio with Python”, 2020. https://github.com/musikalkemist/DeepLearningForAudioWithPython/blob/master/12-%20Music%20genre%20classification:%20Preparing%20the%20dataset/code/extract_data.py (erişim 23 Temmuz 2023).
- [40] V. Valerio, R. Miranda Pereira, Y. Costa, D. Bertolini, ve C. Silla, “A Resampling Approach for Imbalanceness on Music Genre Classification Using Spectrograms”, May. 2018.



ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı :Adem EKİNCİ

Doğum Tarihi ve Yeri :

Yabancı Dili :İngilizce

E-posta :

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Bilgisayar Mühendisliği	Düzce Üniversitesi	2023
Lisans	Bilgisayar Mühendisliği	Düzce Üniversitesi	2017
Lisans	Bilgisayar Sistemleri Öğretmenliği	Sakarya Üniversitesi	2014
Lise	Bilgisayar	Fatih Mesleki ve Teknik Anadolu Lisesi	2008

TEZDEN ÇIKAN YAYIN

A. Ekinci ve E. Küçükkülahlı , "Classification of Baby Cries Using Machine Learning Algorithms", *Eastern Anatolian Journal of Science*, c. 9, sayı. 1, ss. 16-26, Haz. 2023