T.C.
TOKAT GAZİOSMANPAŞA ÜNİVERSİTESİ SOSYAL BİLİMLER ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ İLE KALÇA ÇIKIĞI TEŞHİSİNDE DOĞRU GÖRÜNTÜNÜN TESPİT EDİLMESİ

Hazırlayan
Onur ALTINTAŞ

Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman
Doç. Dr. Bülent TURAN

TOKAT – 2024

T.C.
TOKAT GAZİOSMANPAŞA ÜNİVERSİTESİ SOSYAL BİLİMLER ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ İLE KALÇA ÇIKIĞI TEŞHİSİNDE DOĞRU GÖRÜNTÜNÜN TESPİT EDİLMESİ

Hazırlayan
Onur ALTINTAŞ

Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman
Doç. Dr. Bülent TURAN

TOKAT – 2024

BİLİMSEL ETİK SAYFASI

Tokat Gaziosmanpaşa Üniversitesi Lisansüstü tez yazım kılavuzuna göre, Doç. Dr. Bülent Turan danışmanlığında hazırlamış olduğum “Makine Öğrenmesi ile Kalça Çıkığı Teşhisinde Doğru Görüntünün Tespit Edilmesi” adlı Yüksek Lisans tezimin bilimsel etik değerlere ve kurallara uygun, özgün bir çalışma olduğunu, aksinin tespit edilmesi halinde her türlü yaptırımı kabul edeceğimi beyan ederim.

12.06.2024

Onur ALTINTAŞ

İmza

İTHAF

Bu tez kendilerinden eksilttiğim enerjimi çalışmalarıma vermemi sağlayan eşim ve kızıma ayrıca her türlü yokluk ve yoksulluğa rağmen dünyayı güzelleştirme umudunu kaybetmeyen çocuklara ithaf edilmiştir.

ÖNSÖZ

İnsan sağlığı dünyada en önemli konuların başında gelmektedir. Özellikle bebeklerde oluşan sağlık sorunları, çözülmesi için hiçbir fedakarlıktan kaçınılmayacak bir konudur. Gelişimsel kalça displasizisi, bebeklik döneminde ortaya çıkan bir rahatsızlıktır. Erken teşhis edilmesi durumunda bebeklerin acısız ve kısa süreli yöntemlerle tedavisi mümkündür. Hayatımızın her alanına giren makine öğrenmesi ve yapay zekâ, sağlık alanında da etkili kullanılmaktadır. Bu doğrultuda yapılan çalışmada doktorların gelişimsel kalça displazisindeki erken teşhis başarısını artırmak için makine öğrenmesi algoritmaları kullanılarak doktorlara yardımcı olacak bir karar destek sistemi oluşturulmuştur.

Yüksek lisans eğitimim boyunca yönlendirmeleri ve verdiği bilgilerle her zaman yanımda olan, tecrübe ve deneyimlerini benden esirgemeyen Doç. Dr. Bülent TURAN'a sonsuz teşekkürü borç bilirim. Hem lisans hem de yüksek lisans eğitimim boyunca ne zaman ihtiyacım olsa ulaşabildiğim Doç. Dr. Mehmet BAKIR'a; motive edici cümleleri ile ilerlememi sağladığı için ve yardımları, destekleri, abiliğinden dolayı şükranlarımı sunarım.

ÖZET

Gelişimsel kalça displazisi, bebeklik ve erken gelişim döneminde ortaya çıkan ortopedik bir rahatsızlıktır. Erken teşhis edilmesi durumunda kolay ve acısız tedavi yöntemlerine sahiptir. Ancak teşhisin gecikmesi durumunda ameliyat gibi zor, acılı ve maliyeti yüksek tedaviler gerektirmektedir. Ayrıca teşhisin gecikmesi hastalarda kalıcı ortopedik rahatsızlıklara da neden olmaktadır. Bu çalışma, gelişimsel kalça displazisinin teşhisinde ilk aşama olan doğru görüntü tespitinin ultrason kayıtlarından makine öğrenmesi algoritmaları ile çıkarılmasını amaçlamaktadır. Çalışma ile erken teşhis oranının yükseltilmesi, böylece daha kolay ve düşük maliyetli yöntemler ile tedavinin yapılması, kalıcı ortopedik rahatsızlıkların ortadan kaldırılması hedeflenmektedir. Çalışmada ultrason kayıtlarından yakalanan görüntülerin özellikleri, LBP yöntemi ve önceden eğitilmiş VGG16, VGG19, InceptionV3, MobileNet, DenseNet121, ResNet50 derin öğrenme modelleri ile elde edilmiştir. Çıkarılan özellikler ile ayrı ayrı veri setleri hazırlanmıştır. Hazırlanan veri setlerine Random Forest, Support Vector Machine, Naive Bayes, Logistic Regresyon, KNN, Decision Tree ve Gradient Boosting makine öğrenme algoritmaları uygulanmıştır. Elde edilen sonuçlar karşılaştırılmıştır. Bu sonuçlar incelendiğinde VGG16 derin öğrenme modeli kullanılarak elde edilen veri seti üzerinde çalışan Gradient Boosting makine öğrenme algoritması 0,9123 Accuracy oranıyla en başarılı yöntem olmuştur. Kaynak kullanım performansına bakıldığında en düşük kaynak kullanan yöntem olmasa da düşük kaynak kullanımı açısından dikkat çekicidir. Sonuçlar erken teşhis oranını artıracak kadar tatmin edici olarak değerlendirilmiştir. Yapılan çalışmanın bu alanda çalışan doktorlara destek olacağı düşünülmektedir.

Anahtar Kelimeler: Makine Öğrenmesi, Gelişimsel Kalça Displazisi, Sınıflandırma, Öznitelik Çıkarma, Ultrason Görüntüsü

İÇİNDEKİLER

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

KISALTMALAR

GKD	Gelişimsel Kalça Displazisi
OA	Osteoar Trit
AA	Alfa Açısı
US	Ultrason
OF	Optik Akış
KNN	K-Nearest Neighboors
GBA	Gradyan Artırma Algoritması
CART	Classification and Regression Tree
RTİ	Radyal Temel İşlev
KRF	Kernel Rastgele Orman
RF	Random Forest
SVM	Support Vector Machine
DVM	Destek Vektör Makineleri
NB	Naive Bayes
LR	Logistic Regresyon
DT	Decision Tree
GB	Gradient Boosting
CNN	Convolutional Neural Network
R-CNN	Region Based Convolutional Neural Network
DNN	Derin Sinir Ağı
FDCNN	Özellik Rehberli Gürültü Giderme Konvolüsyon Sinir Ağı
AIUM	Amerikan Tıp Ultrasonografisi Enstitüsü
ACR	Amerikan Radyoloji Koleji
ESA	Evrişimli Sinir Ağı

GİRİŞ

Gelişimsel kalça displazisi (GKD), genellikle kalıtsal, kültürel ya da intrauterin faktörlerden kaynaklanan ve bebeklerde görülen ortopedik bir rahatsızlıktır (Siegel, 2011) (Terence W O'Neill, 2018). Bebeklerin gevşek bir kalça eklemi ile doğmasından dolayı ortaya çıkmaktadır ve dünya ülkelerinde ‰1-‰3 oranında görüldüğü tahmin edilmektedir (V. Bialik, 1999) (Furnes O., 2001). Asetabular displazi, tam çıkık, subluksasyon, instabilite ve femoral displazi olarak sınıflara ayrılabilir (Bahar Kural, 2019). Erken teşhis edildiğinde atel ya da korse gibi kolay ve acısız tedavi yolları vardır. Erken teşhis edilemediğinde cerrahi operasyona gerek duyulabilmektedir. Hatta bacakta kısalık gibi kalıcı ortopedik rahatsızlıklara da neden olabilmektedir (H. Atalar, 2007) (Tao Chen, 2022) Teşhisin gecikmesi durumunda ortaya bu zor ve maliyetli tedavilerden kaçınmak için birçok ülke 6-8 haftalık bebeklere DDH testi uygulamaktadır (Jingnan He, 2023).

GKD; kalça radyografisi, ultrasonu ya da fiziki muayene gibi yöntemlerle teşhis edilebilmektedir (Abdulselam BATU, 2018). Kalça ultrasonu, radyografiden daha öncelikli bir araçtır. Bunun sebebi ise radyasyon riskinin olmaması ve maliyetinin daha düşük olmasıdır. Ancak ultrasonda tecrübe ile doğru teşhisin başarısı yüksek olmaktadır. Doktorun tecrübesi teşhisin başarısına doğrudan etki etmektedir. Yanlış teşhis, erken ve kolay tedavi şansının kaybolmasına neden olabilmektedir (Seda Sahin, 2017). Doğru yapılan ultrason teşhis için en önemli kriterdir (Si Cheng Zhang, 2023). Graf yöntemi, GKD'nin tespit edilmesinde kullanılan önemli yöntemlerdendir. 1978 yılında Graf tarafından önerilmiştir (Graf R. , 1984) (H. Ömeroğlu, 2001) (Tao Chen, 2022). Bu yöntem statik ultrasonagrafiye dayanır. Statik metotta femur başının yerleşim yeri, asetabulumun morfolojik yapısı ve açısal değerleri göz önüne alınarak teşhis konulur (Si Wook Lee, 2021) (Graf, 1984). Alfa ve beta açıları rahatsızlığın derecelendirilmesinde kullanılır. Alfa açısı, femur başı ile asetabulum arasındaki uygunluk derecesini değerlendirmek için kullanılır. Beta açısı ile de asetabulum çatısının diklik derecesinin tespiti yapılır. Bu açıların yorumu, kullanılan farklı özel ölçüm yöntemlerine ve referans değerlerine göre değişiklik gösterebilir. Ancak ultrasondan alınan görüntü yeterli değilse ölçümlerin yanlış olma riski artar (Vaquero-Picado, 2023).

Sağlık alanında teşhis ve tedavileri desteklemek amacıyla makine öğrenmesi ve derin öğrenme yöntemleri etkili bir şekilde kullanılmaktadır (Umut Kaya, 2019) (Haydar

Hoşgör, 2022) (Bas H.M. van der Velden, 2022) (Mark Cicero, 2017) (Manoj Mannil, 2018) (Recep Sinan Arslan, 2023). GKD teşhisi için doğru görüntünün sınıflandırılması konusunun; veri toplama, ön işleme, öznitelik çıkarma, eğitim, sınıflandırma ve test aşamalarından oluştuğu öngörülmektedir. Çalışma ile doktorların tecrübe koşulu olmadan doğru ultrason görüntüsünü alabilmelerine yardımcı olacak bir karar destek sistemi ortaya konmaktadır.

Paserin vd. CNN ve R-CNN kullanarak GKD teşhisi için bir tıbbi görüntü değerlendirme sistemi kurmuşlardır (Paserin, 2018). Xindi Hu vd. R-CNN kullanarak üç aşamada çalışan otomatik bir sistem geliştirmişlerdir. Bu sistem ile alfa açısını %93, beta açısını %85 Accuracy ile tespit edebilmektedir (Xindi Hu, 2021). Si-Cheng vd. GKD teşhisi için x ışını görüntülerini kullanarak oluşturdukları derin öğrenme sistemi ile %98.9 başarı elde etmişlerdir. (Si-Cheng Zhang, 2020). Weize Hu vd. 1265 hastanın görüntüsünü kullanarak lokal özelliklerin sınıflandırılması için mask R-CNN tabanlı bir sistem oluşturmuşlardır ve %95 başarıya ulaşmışlardır (Weize Xu, 2022). Than-Tu Pham vd. GKD radyografilerinde otomatik göç yüzdesi ölçümü için oluşturdukları CNN tabanlı sistem ile %94.5 başarıya ulaşmışlardır (Thanh-Tu Pham, 2021). Guanfang Dong vd. taşınabilir ultrason cihazlarında gürültüyü gidermek için Özellik Rehberli Gürültü Giderme Konvolüsyon Sinir Ağı (FDCNN) sistemi geliştirerek %87 başarıya ulaşmışlardır (Guanfang Dong, 2021). Bingxuan Huang vd. GKD teşhisi için çoklu ölçüm yöntemlerini kullanarak otomatik ölçüm için bir derin sinir ağı (DNN) geliştirmişlerdir. Bu çoklu ölçüm yöntemlerinde Amerikan Tıp Ultrasonografisi Enstitüsü (AIUM) ve Amerikan Radyoloji Koleji (ACR) kılavuzlarını baz almışlardır (Bingxuan Huang, 2023). Lee vd. GKD görüntülerindeki kemik yapısını segmente etmek için iki aşamalı bir derin öğrenme modeli önermişler ve kritik açıyı tespit edebilmek için Mask R-CNN modeli ile %80,9 Accuracy oranına ulaşmışlardır (Si Wook Lee, 2021). Lui vd. kalça kemik segmentasyonu için açı ölçümü sırasında doktorlara destek için bir model önerdiler ve herkese açık bir veri setinden 400 görüntü kullandılar. Bu model doktorların açı ölçümü sırasındaki hata oranını %6-10'dan %2'ye düşürmüştür (Ruhan Liu, 2021). Chen vd. Graft tip I ve II kalça kemikleri için düzlem tespiti ve açı ölçümü yapabilen bir derin öğrenme modeli önermişlerdir. Bu model ile %94,71 Accuracy oranına ulaşmışlardır (Tao Chen, 2022). Kinusaga vd. Graf tip III ve IV kalça kemikleri için GKD teşhisi için transfer öğrenme kullanarak oluşturdukları model ile %99 Accuracy oranına ulaşmışlardır (Maki

Kinugasa, 2023). Zhang vd. kalça kemiğinin olgunluğunu tespit edebilmek için yaptıkları çalışmada, 268 hastanın görüntüsünü kullanarak geliştirdikleri model ile %94'lük bir başarı elde etmişlerdir (Si Cheng Zhang, 2023). Hareendranathan vd. GKD'nin erken teşhis edilebilmesi için ultrason görüntülerinin kalitesini artırmayı amaçlamışlardır. Bu amaç doğrultusunda eğittikleri ağ ile taşınabilir cihazlardan elde edilen görüntülerin kalitesini artırmışlardır (A. R. Hareendranathan, 2022).

Bahsi geçen çalışmalar, GKD için ultrason görüntülemesinin tecrübeli ve iyi eğitilmiş bir personel tarafından yapılmasına ihtiyaç duymaktadır. Görüntüler hatalı ya da eksik ise ve teşhis için yeterli değilse GKD fark edilemeyecek ve teşhis gecikecektir. Bu nedenle doğru görüntüleri almak hayati öneme sahiptir (Xindi Hu, 2021) (Si Cheng Zhang, 2023) (Maki Kinugasa, 2023) (Hasan Basri Sezer, 2019). Çalışmada 100 kişinin ultrason kayıtlarından çıkarılan 3600 görüntü kullanılmıştır. Görüntüler "Doğru" ve "Yanlış" olarak iki sınıfa ayrılmıştır. Veri çoğaltma yöntemleri ile her iki sınıftan da 2078 olmak üzere toplamda 4156 görüntü elde edilmiştir. Bu görüntülerin öznitelikleri LBP yöntemi ve VGG16, VGG19, InceptionV3, MobileNet, DenseNet121, ResNet50 derin öğrenme modelleri ile çıkarılmış ve her model ile ayrı bir veri seti oluşturulmuştur. Bu veri setlerine; Random Forest, Support Vector Machine, Naive Bayes, Logistic Regresyon, KNN, Decision Tree ve Gradient Boosting makine öğrenme algoritmaları uygulanmıştır. Her algoritma; her veri setine ayrı ayrı uygulanmış ve sonuçları Accuracy, Recall, Precision, F1 Score performans metrikleri üzerinden karşılaştırılmıştır. VGG16 modeli ile elde edilen veri setine uygulanan Gradient Boosting makine öğrenmesi algoritması 0.9123 Accuracy oranı ile en başarılı yöntem olmuştur.

BÖLÜM 1: GELİŞİMSEL DİSPLAZİSİ

Gelişimsel kalça displazisi (GKD), bebeklik ve erken gelişim döneminde meydana gelen ve çocukluk çağı engelliliklerinde önemli bir paya sahip olan geniş bir anormal kalça gelişimi spektrumunu tanımlamaktadır. Gelişimsel kalça displazisi (GKD), bebeklerde ve küçük çocuklarda en sık görülen iskelet bozukluklarından biridir (Carol Dezatuex, 2007). GKD, kalça çıkığı olmaksızın hafif asetabular displaziden kalça çıkığının tespit edilmesine kadar değişen geniş bir yelpazedeki ciddi durumları kapsamaktadır (Scott Yang, 2019). Gelişimsel kalça displazisi (GKD), küresel insidansı Afrika'da %0,006'dan Amerika'da %7,61'e kadar değişen yaygın bir pediatrik hastalıktır (Randall T. Loder, 2011). 60 yaşın altındaki kişilerde primer kalça replasmanlarının %29'unun GKD'den kaynaklandığı bildirilmektedir (Carol Dezatuex, 2007). Ayrıca 40 yaş altı kadınlarda GKD'nin kalça artritinin en yaygın nedeni olduğu ve Amerika Birleşik Devletleri'ndeki tüm total kalça replasmanlarının %5 ila %10'unun GKD nedeniyle gerçekleştiği bildirilmiştir (Shaw, 2016).

GKD tanısı koymak için kullanılan başlıca yöntemler fiziksel muayene, ultrasonografi (altı aylıktan küçük bebekler için) ve ön-arka pelvik radyografilerdir (Mariana S. Silva, 2019). Altı aydan büyük çocuklar için en sık kullanılan muayene prosedürü ön-arka pelvik radyografidir (Tim P.C. Kane, 2003). 6 aylıktan küçük hastalar için ultrason tanısı, kalça gelişiminin taranması ve değerlendirilmesi için daha uygundur. Gelişimsel Kalça Displazisi (GKD) ‰1-‰3 bebeği etkiler (Furnes O., 2001) ve doğumdan sonraki 6 hafta içinde teşhis edilirse kolayca tedavi edilebilir (Carol Dezatuex, 2007). Gözden kaçması durumunda cerrahi müdahale gerekebilir ve ilerleyen yıllarda Osteoar- trit'e (OA) neden olabilir.

1980 yılında Graf tarafından önerilen yöntem, tam spektrumlu ultrason görüntülerini kullanarak kalça displazisinin şiddetini belirlemiştir ve bu metot, ölçülen eklem açılarının diğer yaklaşımlara göre daha az değişkenlik gösterdiği için pediatrik ortopedik cerrahlar tarafından tercih edilmektedir. 21.yy'da çoğu hastanede, deneyimli klinisyenler tarafından kalça ultrasonu taraması gerçekleştirilir ve değerlendirilir. Ancak kalça displazisi teşhisinde genellikle düşük Accuracy ve tekrarlanabilirlik sorunları yaşanmaktadır.

Yapılan çalışmalar GKD ultrason taramasının düşük tanısal doğruluğunu (%55,0-89,4) ve düşük tutarlılığını (%17,7-80,8) bildirmiştir (Sie Heng Sharon Tan, 2019) (Niamul Quader, 2018) (Anand Pillai, 2011) (Nerys F. Woolacott, 2005). Klinisyenler tarafından ölçülen α ve β açılarının standart sapma aralığı yaklaşık olarak ± 3 ~ 7,1 ve ± 5,9 ~ 10,1'dir (H. Ömeroğlu, 2001) (Jacob L Jaremko, 2014) (Si Wook Lee, 2021). Klinisyenlerin eğitimi ve deneyimi ile klinik taramanın doğruluğu arasında güçlü bir ilişki vardır (Ahmet Sinan Sarı, 2020) (E A Simon, 2004). Ayrıca Quader ve arkadaşları, ultrason tanı göstergelerinin zamansal değişkenlik gösterdiğini tespit ederek "yüksek oranda tekrarlanabilir ve güvenilir bir GKD ultrason tarama tanı yöntemine acil ihtiyaç olduğunu" bildirmişlerdir.

Barlow ve Ortolami manevraları (hafif displaziye duyarlı olmayan) ile kalçanın fiziki muayenesi yapılmaktadır. Kalça gevşekliği, asimetrik cilt kırışıklıkları ve makat pozisyonu gibi risk faktörlerine dayalı olarak klinik GKD şüphesi olan bebeklere, genellikle hafif GKD için oldukça hassas olan ultrason muayenesi önerilir. Gelişimsel kalça displazisi geleneksel olarak aşağıda belirtilen şekilde Alfa Açısı (AA) isimli asetabular ve ilium çatı arasında bulunan Graf kriterleri kapsamında 2D ultrason kalça görüntüsünden konmaktadır. Graf kriterlerine göre AA<43 ciddi displastik ve AA >60◦ normal kabul edilmektedir (Graf R. , 1984).

Bu teknik büyük ölçüde sonografın Graf Standart Düzleminde ilium, asetabular çatı, labrum ve femur başı gibi açıkça görülebilen işaretlerle 2D görüntü elde etme konusundaki uzmanlığına dayanır. Yapılan önceki araştırmalar prob pozisyon ve oryantasyonunda bulunan farklılıkların yenidoğan bebeklerin 2/3'ünde ve değerlendirmeye tabi tutulan bebeklerin yarısında hatalı tanıya sebep olabileceğini ortaya koymuştur (Jacob L Jaremko, 2014). Bu durumda doğumda gelişimsel kalça displazisinde evrensel tarama programları önerilmez (Susan T. Mahan, 2009).

Şekil 1. Tablete bağlı bir el probu kullanılarak gerçekleştirilen muayene (Abhilash Rakkunedeth Hareendranathan, 2022)

Ultrason (US) görüntüleme, erken çocukluk döneminde kalça displazisi (GKD) teşhisi için şu anda altın standart olarak kabul edilir. Çünkü düşük maliyetlidir ayrıca taşınabilir olması ve potansiyel olarak zararlı iyonlaştırıcı radyasyon kullanmaması gibi avantajlara sahiptir (Lamya Ann Athew, 2013). 2D ultrasonun mevcut klinik standart olmasına rağmen yapılan son zamanlardaki birkaç araştırma, 3D ultrasonun kullanımının anatomik deformitenin daha geniş bir ölçüsünü sağladığını ve probun oryantasyonuyla ilişkili hatalara daha az eğilimli olduğunu göstermektedir (Jacob L Jaremko, 2014). Ayrıca üç dakikalık çalışma süreleri klinik uygunluğu sınırlamakta ve mevcut analiz aşamaları hesaplama açısından maliyetlidir. Ayrıca 3D US'nin kullanılmaya başlanması, volümetrik taramalar konusunda deneyimi olmayanlara daha fazla zorluk çıkarmaktadır. Bu yöntem tanısal ölçümler için yeterli olan yüksek kaliteli US hacimlerinin elde edilmesi, bebek kalça anatomisi hakkında bilgi edinilmesi ve taramaların yorumlanmasında deneyim gerektirdiğinden özellikle zordur. Bu zorluklar 2D US kullanıldığında bile mevcuttur. Örneğin 2011 yılında 8 Alman eyaletinde kalça sonogramlarının kalitesi araştırıldığında test edilen kalça sonograflarının %43'e kadarının lisansı, görüntüleme kılavuzlarının niteliklerini karşılayamadığından iptal edilmiştir (Alexander Kolb, 2017).

Yanlış tanıların en önemli nedenleri şunlardır (Graf R. M., 2013):

1. US probu oryantasyon hataları,
2. yanlış anatomik yorumlama,
3. yeterlilik kontrollerinin eksikliğidir.

US tarama yeterliliğine benzer bir konu olan US standart düzlem tespiti, sonograflara geri bildirim sağlamak amacıyla fetal anormallik taraması ve kardiyak görüntüleme gibi diğer alanlarda ele alınmıştır. Rahmatullah ve arkadaşları, US hacim verilerinde AdaBoost öğrenme algoritmasına yönelik bir yöntem önermektedirler (Bahbibi Rahmatullah, 2011). Noble ile arkadaşları, Hao Chen ile arkadaşları ve Baumgartner ile arkadaşları ise ultrason kayıtlarının 2D dilimlerinin kategorize edilmesinde sınıflandırıcıları önermektedirler (Hao Chen, 2017) (Baumgartner, 2016) (Noble, 2017).

Raphael ve arkadaşları, ardışık 2D US dilimlerinin göreceli konumlarını çıkarmak için makine öğrenimi kullanımını tanıtmış ve optik akış (OF) bilgilerinin ve bir atalet ölçüm biriminden gelen verilerin eklenmesinin performansı artırabileceğini göstermiştir (Raphael Prevost, 2018). Bu gelişmenin ardından araştırma grupları derin öğrenme yaklaşımında varyasyonlar sunmuştur. Mingyuan ve arkadaşları, hacim yeniden yapılandırmasında bağlam ipuçlarını ve şekil öncellerini kullanarak eğitim sürecine kendi kendine süper ve karşıt öğrenme eklemeyi önermiştir (Mingyuan Luo, 2021).

BÖLÜM 2: DERİN ÖĞRENME

21.yy'da klasik sinir ağlarının yerini alan derin öğrenme; makine öğrenme tekniklerinin geniş bir sınıfını, çok sayıdaki katman ile öğrenmenin tanımıdır. Derin öğrenme işlem birimlerinden olan neuron'larda nonlineer fonksiyonu kullanılmaktadır (Schmidhuber, 2015).

Derin öğrenme ve makine öğrenmesi önceki verilerden öğrenme ve tahmin yapabilme kabiliyeti kazandırmak için kullanılır. Makine öğrenimi genellikle istatistiksel yöntemleri kullanırken derin öğrenme ise çok katmanlı sinir ağlarıyla hesaplamalar yapmaktadır (Ezel, 2018).

Derin öğrenme, birden fazla katmandan oluşan ve veri setleriyle sonuç tahmin eden bir makine öğrenme yöntemidir. Makine öğrenimi, yapay zekânın alt dalı olarak kabul edilebilirken derin öğrenme makine öğreniminin bir parçasıdır. Yani derin öğrenme ve makine öğrenimi birbirinden farklı anlamları olan terimlerdir. Ancak makine öğrenimi daha geniş kapsamlı olan yapay zekâ kavramının bir alt kümesidir.

Şekil 2. Yapay Zekâ, Derin Öğrenme ve Derin Öğrenme Hiyerarşisi (Derin Öğrenme (Deep Learning) Nedir?, 2022)

Derin öğrenme; gözetimli, yarı gözetimli veya gözetimsiz olabilir ve bu türlerde çok sayıda veri girişiyle ayırt edici özellikleri kendi kendine öğrenir. Öğrenme başarısı genellikle kullanılan veri kalitesiyle ve miktarıyla doğru orantılıdır. Veriler, birbirini izleyen çok katmanlı yapılar aracılığıyla işlenir. Üst katmanlar, daha genel özellikleri öğrenirken alt katmanlar daha ayrıntılı ve özelleşmiş özellikler çıkarmaya odaklanır. Bu

çok katmanlı yapı, veri setlerindeki karmaşık ilişkileri keşfetmek ve öğrenmek için tasarlanmıştır.

Ses tanıma, yüz tanıma, araçlarda ya da sürücüsüz araçlarda otopilot özelliği sağlamak gibi kullanım alanları bulunmaktadır. Derin öğrenme sayesinde kamera kayıtlarının devamlı izlenmesi yerine sadece olağandışı hareketlerde alarm sistemleri devreye girerek kolaylık sağlanmaktadır. Sağlık alanında kanser gibi önemli hastalıkların teşhisinde zamandan tasarruf sağlar. Derin öğrenme algoritmalarına kanserli hücreler tanıtılarak yeni hücrelerin kanserli olup olmadığı tanısı hızlı ve kolay bir biçimde gerçekleşmektedir. Ayrıca bu konuda başarıyı da artırmaktadır.

2.1. VGG16 VE VGG19

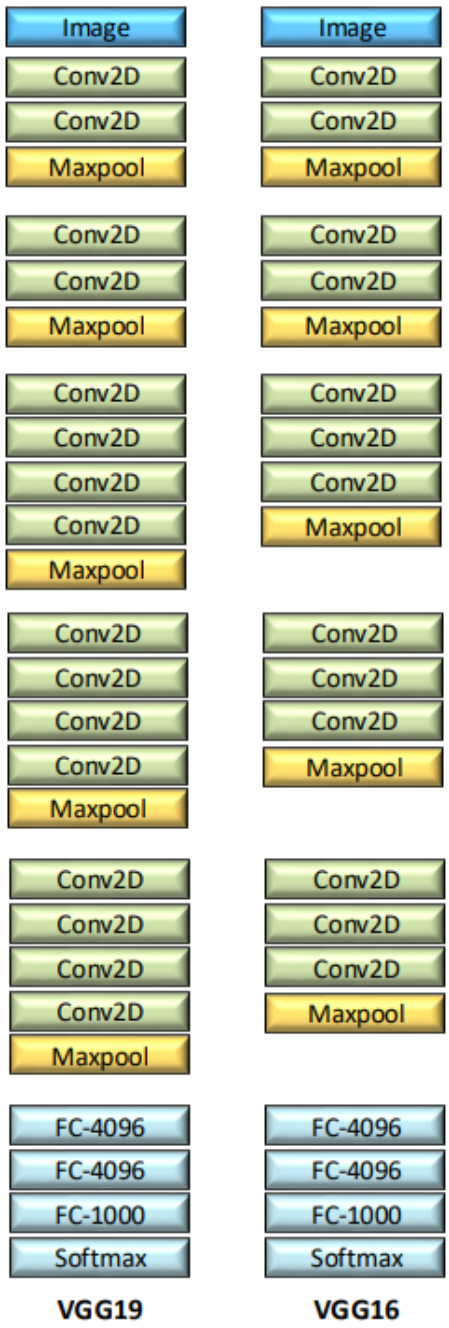
Evrişimsel sinir ağları sınıflandırma, sinyal/görüntü işleme, nesne takibi, nesne tanıma vb. alanlarda başarılı olduğundan dolayı 21.yy'da sıklıkla kullanılan yöntemler arasındadır (Suat Toraman, 2018).

VGG16, 2014 yılında Oxford Üniversitesi Görsel Geometri Grubu Laboratuvarından Karen Simonyan ve Andrew Zisserman tarafından önerilmiş daha sonra mimariye birkaç katman eklenerek VGG19 olarak bilinir hale gelmiştir (Manali Shaha, 2018) (Tianmei Gua, 2017).

Etkisi büyük olan bir model olan AlexNet, Evrişimli Sinir Ağları'nın (ESA) popüler hale gelmesinde önemli bir rol oynamıştır. Daha sonra, Oxford Üniversitesi'ndeki Visual Geometry Group (VGG) tarafından VGG16 modeli geliştirilmiştir. VGG16, konvolüsyon katmanlarında küçük filtreler (3x3) kullanan bir yapıya sahiptir. Bu model, 13 konvolüsyon katmanı ve 3 tam bağlantılı katmandan oluşur. Ayrıca 2x2 boyutunda 5 adet havuzlama (max pooling) katmanı bulunmaktadır. Son katmanda ise sınıflandırmayı gerçekleştiren bir softmax katmanı bulunmaktadır. Softmax katmanı, gelen giriş verisini sınıflandırmak için kullanılır. Bu yapı, görüntü sınıflandırma gibi görevlerde kullanılmak üzere tasarlanmıştır ve derin öğrenme alanında önemli bir adımı temsil etmektedir (Ihsan Ullah, 2018). VGG19, 16 konvolüsyon katmanı ve 3 tam bağlantılı katmandan meydana gelir. Bu modelde, VGG16 gibi 5 havuzlama (max pooling) katmanı ve sınıflandırmayı gerçekleştiren softmax son katmanı bulunmaktadır. VGG19 yaklaşık olarak 144 milyon parametre içerirken VGG16, 138 milyon parametreye sahiptir. Bu artış, daha fazla katman ve daha karmaşık bir yapıyla birlikte gelir. Genellikle daha yüksek bir model karmaşıklığına işaret eder. Bu durum, daha büyük ve daha geniş veri setleri üzerinde

performans artışı gösterebilme potansiyeline sahip olabilir (Kasthurirangan Gopalakrishnan, 2017) (Simonyan, 2014).

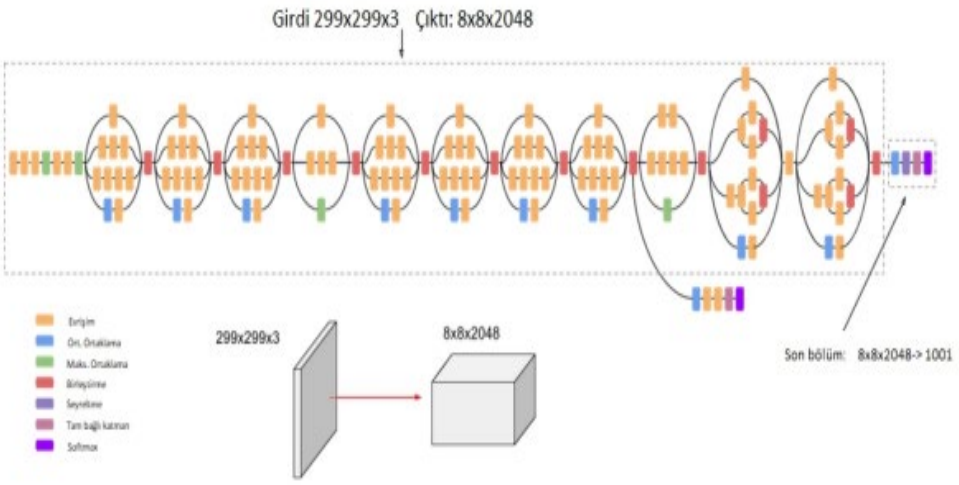
Aşağıdaki şekilde VGG16 ve VGG19'un mimarisi yapıları gösterilmektedir.



Şekil 3. VGG16 ve VGG19 Mimari Yapısı (Suat TOROMAN, 2020)

2.2. INCEPTİONV3

ILSVRC 2015 yarışmasında başarılı olan InceptionV3 mimarisi, Google araştırmacıları tarafından geliştirilmiştir ve ismini "Inception" filmindeki farklı gerçeklik katmanlarından esinlenerek almıştır. InceptionV3 mimarisi, farklı boyutlardaki evrişim ve havuzlama katmanlarının kombinasyonundan oluşan farklı modüllerden meydana gelir (Alex Krizhevsky, 2017). Her bir InceptionV3 modülü, farklı boyutlarda evrişim (convolution) ve havuzlama (pooling) katmanlarının paralel olarak çalıştığı bir yapıya sahiptir. Bu modüller, ağın farklı ölçeklerdeki öznitelikleri algılamasına olanak tanır ve aynı anda birden fazla özniteliği çıkarabilir. Böylece farklı boyutlardaki özniteliklerin aynı anda dikkate alınması ve öğrenilmesi sağlanır. InceptionV3 mimarisi, bu modüllerin farklı boyutlardaki evrişim ve havuzlama katmanlarını paralel olarak kullanarak derin ağın hem hesaplama açısından verimli olmasını sağlar hem de öznitelik çıkarımı ve öğrenme açısından daha etkili olmasını amaçlar. Bu modüler yapı, daha karmaşık ve farklı ölçeklerdeki bilgileri kapsayarak daha iyi bir performans ve genellemenin elde edilmesine katkıda bulunur (Christian Szegedy, 2016). Aşağıdaki şekilde InceptionV3 mimarisi gösterilmektedir (Christian Szegedy, 2016):

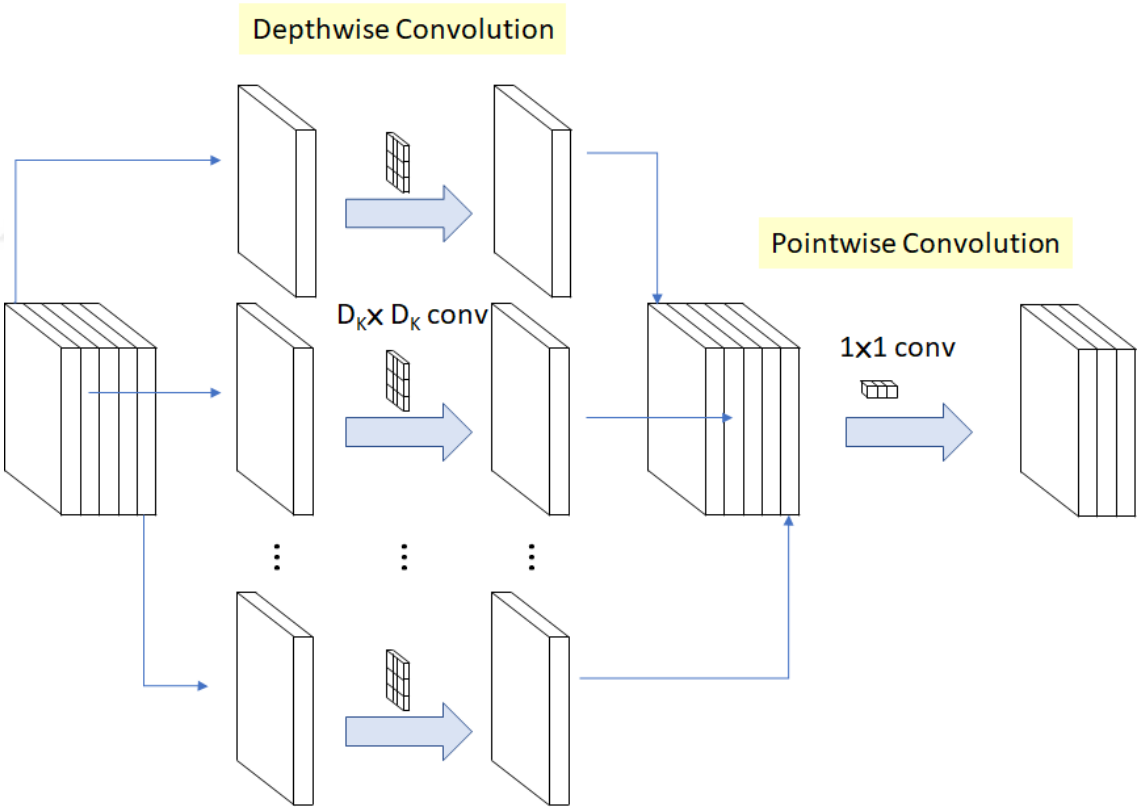


Şekil 4. InceptionV3 Mimarisi (Christian SZEGEDY, 2016)

2.3. MOBİLENET

MobileNet, Google araştırmacıları tarafından mobil uygulamalar ve kaynak sınırlı cihazlar için tasarlanmış bir Evrişimli Sinir Ağı (ESA) mimarisidir. MobileNet mimarisinde derinlemesine ayrılabilir sinir ağları (depthwise separable convolutions)

kullanılır. Bu, geleneksel evrişimli sinir ağı (CNN) yapılarına kıyasla daha az parametreye sahip olma özelliğine sahiptir. Derinlemesine ayrılabilir sinir ağları, evrişimli işlemleri iki aşamada gerçekleştirir: İlk olarak evrişim işlemi kanal bazında (depthwise convolution) gerçekleştirilir ve ikinci aşamada birleştirme işlemi yapılır (pointwise convolution). Bu yöntem, geleneksel evrişim işlemlerine kıyasla daha az parametre kullanır. Bu da daha hafif ve daha az kaynak tüketen modellerin oluşturulabilmesine imkân tanır. MobileNet, parametre sayısının azalmasıyla birlikte işlem maliyetini düşürerek mobil cihazlarda ve kaynakaları sınırlı ortamlarda daha etkili bir şekilde kullanılabilmesini sağlar. Bu, mobil uygulamalarda derin öğrenme modellerinin kullanımını kolaylaştırırken performansı da koruma veya artırma potansiyeline sahiptir (Andrew G. Howard, 2017). Aşağıdaki şekilde MobileNet mimarisi gösterilmektedir.

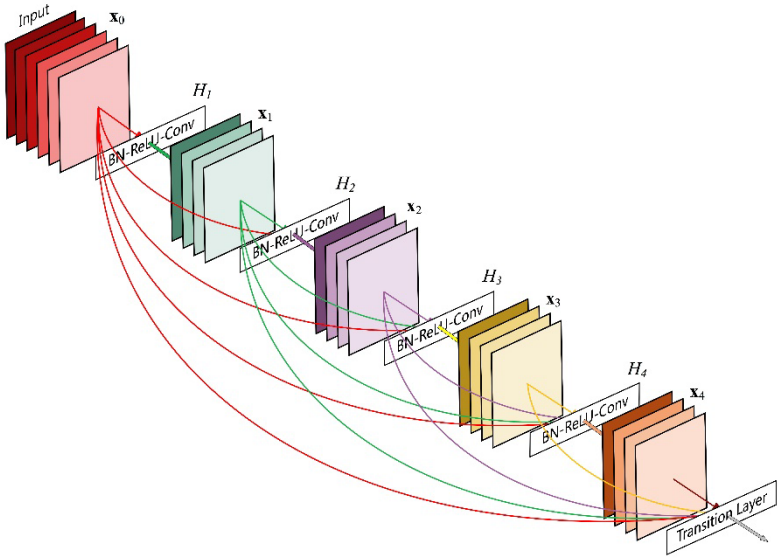


Şekil 5. MobileNet Mimarisi (MobileNet, 2023)

2.4. DENSENET121

DenseNet121, her katmanın ileri beslemeli olarak sonraki her katmana bağlı olduğu bir ağ mimarisidir. Bütün katmanlarda, önceki katmanların öznitelikleri farklı

girdiler olarak işlenirken kendi öznitelikleri sonraki her katmana girdi olarak gönderilir. ILSVRC 2012 veri setinde DenseNet121 ve ResNet benzer Accuracy oranına sahiptir. Fakat DenseNet121 FLOP miktarının yarısına yakınını, parametre sayısının ise yarısından daha azını kullanır (Gao Huang, 2016).



Şekil 6. DenseNet121 Mimarisi (Atik, 2022)

DenseNet121'de birleştirme kullanılır. Her katman kendinden önceki bütün katmanlardan kollektif bilgi yani öznitelik alır. Bundan dolayı kurulan ağ daha ince ve kompakt olabilmektedir. Yani kanal sayısı daha az olur. Sonuç olarak yüksek hesaplama gücü ve daha az bellek kullanımına sahiptir.

BÖLÜM 3: MAKİNE ÖĞRENMESİ ALGORİTMALARI

3.1. K-EN YAKIN KOMŞU (K-NEAREST NEİGHBOURS-KNN)

K-En Yakın Komşu, bir nesnenin ait olduğu sınıfın belirlenmesi amacı ile sınıflandırma ve regresyon problemlerinde kullanılan nesnelerin birbirlerine yakınlık derecelerine göre sınıflandırma yapan denetimli bir öğrenme algoritmasıdır. Sık kullanılan makine öğrenmesi algoritmalarından en yakın komşu algoritması, birbirine yakın durumda olan değerlerin benzer olacağı mantığına dayanır (Gamze Kaba, 2022). KNN (K-Nearest Neighbours) algoritması bir regresyon yöntemi olup etkili ve basit bir sınıflandırmadır (T. Cover, 1967). Bir veri noktasının değerini tahmin etmek ya da etiketlemek için çevresinde bulunan en yakın k komşuyu kullanmaktadır. Temel amaç bir veri noktasının büyük bir bölümünün ait olduğu değer ya da sınıfa göre yeni bir veri noktasının tahmin edilmesi ya da sınıflandırılmasıdır. K-En Yakın Komşu algoritmasından meteoroloji, film tavsiyesi, veri madenciliği, kontrol sistemleri ve görüntü işleme gibi farklı alanlardan yararlanılmaktadır. Bu algoritmada temel amaç hangi sınıfa ait olduğu bilinmeyen bir nesnenin bütün nesnelerle karşılaştırılması sonucunda kendisine en yakın k sınıf örneklerinin ortalaması ile gruplardan birine atanmasıdır. Atama işleminde nesnelerin birbirlerine olan uzaklarının hesaplanması için simple matching distance, manhattan distance, öklid uzaklığı, jaccard distance formülleri genellikle kullanılır (Çam, 2021).

K-En Yakın Komşu (KNN) algoritması 1950'lerin başında ortaya atılmıştır. Ancak ilk zamanlarında büyük veri setlerinin öğrenilmesi sürecindeki zaman maliyeti nedeniyle pek popüler olmamıştır. Ancak KNN algoritmasının gelişimi 1960'ların ortalarına doğru daha fazla ilgi çekmeye başlamıştır. 1965'te N.J. Nilsson tarafından oluşturulan minimum uzaklık sınıflayıcı gibi çalışmalar KNN algoritmasının gelişimine katkıda bulunmuştur. Bu çalışmalar, algoritmanın temel prensiplerini daha net bir şekilde açıklamış ve sınıflandırma alanında KNN'in potansiyelini göstermiştir. Daha sonra, 1967'de T. Cover ve P. Hart'ın sunduğu Yakın Komşular Örüntü Sınıflama çalışmalarıyla KNN algoritmasına netlik kazandırılmıştır. Bu çalışmalar, KNN'in sınıflandırma problemlerinde nasıl kullanılabileceği konusunda daha ayrıntılı bir anlayış sunmuş ve algoritmanın kullanım alanlarını genişletmiştir. Bu çalışmalar, KNN algoritmasının gelişmesine ve kullanımının artmasına katkıda bulunmuştur (Gizem Dilki, 2020).

K-En Yakın Komşu Algoritması; sınıflandırma problemleri, örüntü tanıma ve regresyon gibi alanlarda sıkça kullanılan basit bir makine öğrenme modelidir. Bu algoritma, veri noktaları arasındaki Öklid mesafesini kullanarak belirli bir noktanın en yakın komşularını bulur. Sınıflandırma problemlerinde sınıflandırma yaparken regresyon problemlerinde ise tahminlerde bulunmak için kullanılır. Bu basit ancak etkili algoritma, veri noktalarının birbirlerine olan benzerliği temel alarak çalışır ve genellikle birçok problemde iyi performans gösterir. Burada 'k' değeri, kullanıcı tarafından belirlenen bir sabittir ve yeni bir durumu sınıflandırmak veya tahmin etmek için kullanılır. Bu değer, yeni durumla birlikte mevcut tüm benzer özelliklere sahip durumları bulur. 'k' değeri, yeni durumun çevresindeki komşu sayısını temsil eder. Bu komşular, yeni durumun kategorisini belirlemek veya tahmin etmek için kullanılır. 'k' değeri arttıkça, daha fazla komşu dahil edilir ve bu da daha geniş bir bakış açısı sağlayabilir. Ancak aynı zamanda hesaplama maliyetini ve karmaşıklığı artırabilir. Düşük 'k' değerleri ise daha yerel ve spesifik bir analiz yapılmasını sağlar. Bu nedenle, 'k' değerinin seçimi, algoritmanın performansını ve sonuçlarını etkileyebilir (Animesh Urgiriye, 2020).

'k' değeri yukarıda belirtilen nedenlerden dolayı önemlidir ve seçiminde dikkat edilmelidir. 'k' değerinin küçük olması durumunda sistemde sapma artabilir. Büyük veri kümeleriyle çalışırken K-En Yakın Komşu Algoritması bazı sınırlamalarla karşılaşabilir. Her sorgu örneği için tüm diğer örneklerin uzaklığını ölçmek, hesaplama maliyetini artırır. Özellikle büyük veri kümeleriyle bu işlem çok zaman alabilir ve işlemci kaynaklarını fazlasıyla tüketebilir. Bu durum, algoritmanın pratikte kullanımını zorlaştırabilir. Bununla birlikte, bu maliyeti azaltmak için bazı optimizasyon teknikleri vardır. Örneğin, veri kümesini daha küçük parçalara bölmek veya veriye özgü teknikler kullanarak boyut azaltma gibi yöntemlerle hesaplama maliyeti düşürülebilir. Ayrıca yakınsama odaklı teknikler veya örnek seçimi gibi stratejiler de kullanılabilir. Bu yöntemler, algoritmanın uygulanabilirliğini büyük veri kümeleri üzerinde artırabilir. (Ibrahim Mahmood, 2021).

K-En Yakın Komşu Algoritması öğrenme tabanlı algoritmalardan biridir. Bir X değeri ile karşılaşılması durumunda eğitim setinde bulunan örnekler arasından benzerliklere göre sınıflandırılmaktadır (Erdal Taşçı, 2017). Sınıflandırmada değerler arasındaki uzaklık hesaplanmaktadır. Değerlerin birbirlerine uzaklıkları arasındaki k

adedi komşusu dikkate alınmaktadır. En yakın komşular arasından en çok benzerlik gösterilen sınıfa atama yapılmaktadır (Şengül, 2022).

KNN algoritması kullanılmadan önce, belirli sınıf değerlerine sahip örneklerden oluşan bir eğitim veri seti oluşturulur ve ayrıca 'k' parametresi ile kullanılacak mesafe fonksiyonu belirlenir. Sonrasında, sınıflandırma yapılacak yeni bir örneğin eğitim setindeki diğer örneklerden olan uzaklığı belirlenen mesafe fonksiyonuyla hesaplanır. Hesaplama sonucunda, belirlenen 'k' adet örnek eğitim setinden seçilir ve bu seçilen veri kümesi sınıflandırma veri seti olarak adlandırılır. Ardından, sınıflandırma veri setinde en fazla bulunan sınıf, tahmin edilmek istenen örneğin sınıfı olarak atanarak sınıflandırma işlemi tamamlanır. Bu sayede, KNN algoritması herhangi bir örnek için tahmin yapabilir ve sınıflandırma işlemi gerçekleştirilebilir. (Harrington, 2012).

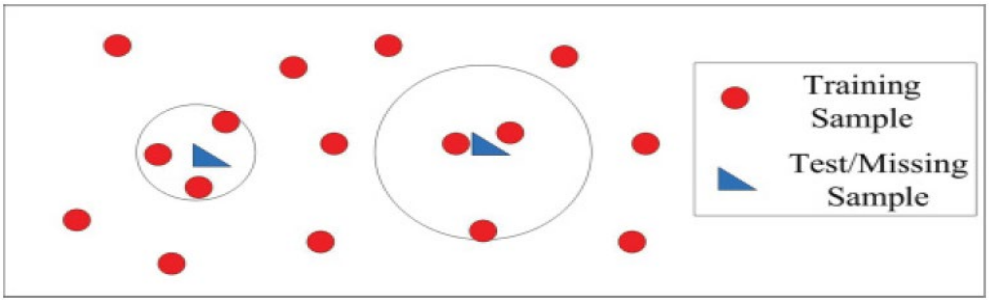
K-En Yakın Komşu algoritması, tembel öğrenme (lazy learning) yöntemiyle çalışır. Yani eğitim verileri model oluşturulana kadar saklanır ve yeni bir örnek sınıflandırılmak istendiğinde, o anda veriye dayalı olarak bir model oluşturulmaz. Algoritma uygulanırken tüm eğitim verileri hafızada tutulur ve yeni bir örneklemin sınıflandırılması gerektiğinde, bu örnek ile eğitim verileri arasındaki mesafeler hesaplanır. Sonrasında, yeni örneğin en yakın 'k' adet komşusunu bulur ve bu komşuların sınıflarını kullanarak yeni örneğin sınıfını belirler. Bu tembel yaklaşım, modelin esnekliğini ve öğrenme yeteneğini artırabilir. Çünkü herhangi bir model oluşturulmadan önce gerçek zamanlı veriye dayalı kararlar alınabilir. Ancak bu yöntemde sınıflandırma işlemi gerçekleştirilirken hesaplama maliyeti yüksek olabilir. Çünkü her sınıflandırma için tüm eğitim verileriyle mesafe hesaplaması yapılması gerekebilir. K-En Yakın Komşu (KNN) algoritması parametrik olmayan bir algoritmadır ve her defasında eğitim veri setini kullanarak sınıflandırma işlemini gerçekleştirir. Bu algoritma, belirli parametrelerin ön işlemlerle belirlenmesi yerine, sınıflandırma için her yeni örneğin eğitim veri setiyle işlenmesini gerektirir. Bu sayede, veri setinin dağılımına yönelik herhangi bir önceden belirlenmiş olasılık varsayımı yapılmaz.

Kullanışlı, uygulaması kolay, yüksek etkili bir yöntem olan K-En Yakın Komşu uygulaması makine öğrenmesinde sıklıkla kullanılan ilk 10 veri madenciliği algoritmalarından biridir (Na, 2021). Nispeten basit, doğrusal ve parametik olmayan sınıflandırıcılar arasındadır. Büyük eğitim veri setlerinde yoğun olarak çalışmaktadır.

Eğitim ve test arasında bulunan benzerliğe dayanmaktadır. Veri kümelerinde bulunan 'n' öznitelikler sınıflandırmalara ayrılmaktadır. Setlerin her biri 'n' boyutundaki uzayda bir nokta olup 'n' boyutlu model uzayını meydana getirmektedir. Testlerin veri seti yakında bulunan 'k' eğitim veri setlerine göre sınıflara atanmaktadır. Aşağıdaki denklem kullanılarak veri kümelerine yakınlık ölçülmektedir (Sharmila Ashok, 2016).

$$(ED) = \sqrt{\sum_{i=1}^{n} (Y_{1i} - Y_{2i})^2}$$

K-En Yakın Komşu (KNN) algoritması mesafe ölçütleri olarak farklı metrikleri kullanabilir ve bu metrikler veri setinin özniteliklerine bağlı olarak değişebilir. Öklid ve Manhattan mesafe ölçütleri, KNN algoritmasında sıkça kullanılan ölçütler arasındadır. Öklid ölçütü, veri noktaları arasındaki doğrudan uzaklığı hesaplar. Manhattan ölçütü ise dikey ve yatay eksende hareket ederek uzaklığı belirler. 'k' değerinin seçimi, algoritmanın performansı açısından oldukça kritiktir. Bu değer, Accuracy seviyesini etkileyebilir ve seçimi problemli olabilir. Literatürde, çapraz doğrulama gibi yöntemler kullanılarak bu 'k' değerinin seçilmesi önerilmektedir. Çapraz doğrulama, aynı veri seti üzerinde farklı 'k' değerlerini deneyerek modelin performansını değerlendirme ve en uygun 'k' değerini seçme yöntemidir. Bu şekilde, farklı 'k' değerlerinin modellerin doğruluğu üzerindeki etkisi daha iyi anlaşılabilir ve en uygun 'k' değeri belirlenebilir. Bu, KNN algoritmasının daha iyi bir performans sergilemesine yardımcı olabilir. 'k'’nın seçilmesindeki önem aşağıdaki şekilde gösterilmiştir. Bu şekilde eklenen bir şeklin kare ya da daire olarak sınıflandırılması hususunda 'nk'nın 1,5 ya da 10 olarak seçilmesinin sonucu farklılaştırdığı anlaşılmaktadır.



Şekil 7. KNN Algoritmasında k Değerinin Seçimi (Shichao ZHANG, 2017)

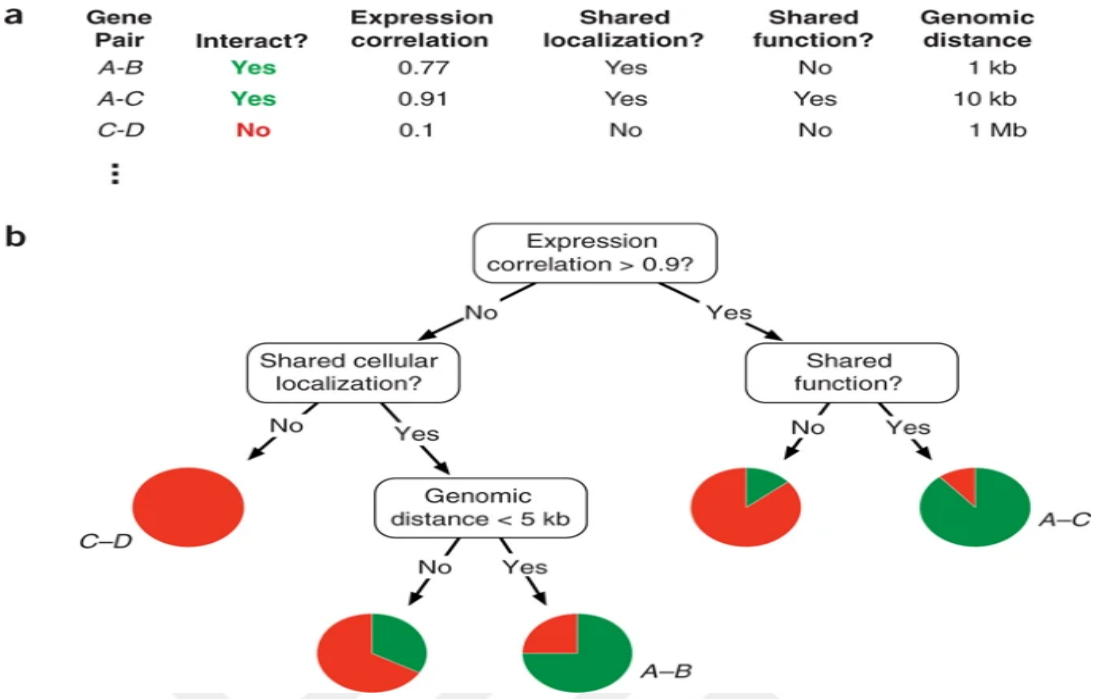
K-En Yakın Komşu (KNN) algoritması oldukça basit anlaşılabilir bir algoritma olsa da tahmin yapılabilmesi için hesaplamaların tamamı eğitim yerine test sırasında yapılır. Bu durum, özellikle büyük veri setlerinde ve yüksek boyutlu verilerde daha fazla hesaplama süresine neden olabilir. Her tahminde, tüm eğitim verisiyle mesafe hesaplamaları yapılması gerektiği için bu süre artabilir. Ayrıca eğitim setinde sınıflar arasında dengesizlik varsa (örneğin, bir sınıf diğerlerine göre daha fazla örnek içeriyorsa) bu durumda KNN algoritmasının dengesizlikten etkilenmemesi için normalleştirme veya sınıf dengesizliğini ele alma işlemleri gerekebilir. Bu tür durumlarda, örneğin ağırlıklı sınıf değerleri gibi teknikler kullanılarak sınıf dengesizliği ele alınabilir. Bu, algoritmanın dengeli bir şekilde tüm sınıfları göz önünde bulundurmasını sağlayarak daha adil bir tahmin yapılmasına yardımcı olabilir (Görmez, 2021).

3.2. KARAR AĞACI (DECİSİON TREE)

Karar ağacı algoritması, sınıflandırma veya regresyon görevlerini gerçekleştirmek için kullanılan bir algoritmadır. En basit tanımı ile karar ağacı analizi, sınıflandırma ve regresyona yönelik bir böl ve yönet yaklaşımıdır. Bu algoritma, bir ağaç yapısı oluşturarak sınıflandırma veya regresyon kararlarını temsil eder. Oluşturulan ağaç yapısında, her bir yaprak düğümü bir sınıf etiketini veya regresyon tahminini temsil eder. Kök düğümden başlayarak yaprak düğümlere giden yollar, veri özniteliklerine göre dallanmaları temsil eder. Her bir düğüm, bir özniteliği ve bu özniteliğin değerine göre bir karar kuralını ifade eder (Şenel, 2020).

Ağaçta karar ve bitiş düğümleri bulunmakta ve karar düğümlerinin her biri dalları etiketleyen ayrık bir fonksiyonu gerçekleştirir. Verilen değerde bu fonksiyonlar ile sınıflandırma işlemi yapılır (Kalaycı, 2018).

Karar ağacı algoritmasının öğrenmesi oldukça basittir ve son zamanlarda sıklıkla kullanılan popüler bir yöntem olmuştur. Bu yöntemin sıklıkla kullanılmasının nedeni sade bir yapıda olması, anlaşılır ve sürecinin basit olmasıdır. Karar ağaçları düğüm, dal ve yaprak olmak üzere 3 kısımdan oluşmaktadır. Özniteliklerin her biri bir düğümü belirtir. Karar ağacı yapısı dallar ile oluşturulur, üst kısım kök olarak ifade edilir, yapraklar ve kök arasında kalan kısımlar da dallardır. Bir karar ağacının yapısı aşağıdaki şekilde modellenmiştir (Kaya, 2023):



Şekil 8. Karar ağacı algoritması (Carl Kingsford, 2008)

Karar ağaçları iç düğümlerin her birinin (yaprak olmayan düğüm) bir öznitelik üzerinde testi ifade ettiği, her yaprak düğümün (uç düğüm) bir sınıf etiketine sahip olduğu, her dalın testin bir sonucunu temsil ettiği, akış şemasına benzer bir ağaçtır (Jiawei Han, 2022). Yukarıdaki şekilde görüldüğü gibi düğüm, dal ve yaprak olarak üç temel kısımdan oluşmaktadır.

Karar ağaçları üç tür düğüm içerir:

a. Kök Düğümü (Karar Düğümü): Ağacın en üst düğümüdür. Tüm kayıtları iki veya daha fazla birbirini dışlayan alt kümeler halinde bölen bir seçimi temsil eder.

b. İç Düğümler (Şans Düğümleri): Kök düğüme bağlı olarak farklı seçenekleri temsil eder. Üst kenarı üst düğüme, alt kenarı ise alt düğümlere veya yaprak düğümlere bağlıdır.

c. Yaprak Düğümleri (Uç Düğümler): Karar ağacının nihai sonucunu temsil eder. Yaprak düğümlerinde sonuçlar bulunur.

Karar ağacı modelinde kökten her bir yaprağa giden yollar, bir sınıflandırma karar kuralını temsil eder. Bu yollar, "eğer-o zaman" kuralları şeklinde ifade edilebilir. Örneğin, "Eğer Koşul 1 ve Koşul 2 ve ... ve Koşul k sağlanıyorsa, Sonuç j oluşur.". Bu şekilde, karar ağacı her bir özniteliğin ve onların değerlerinin kombinasyonlarını temsil eden kurallar ve yollarla veri setindeki desenleri açıklayabilir. Bu kurallar, veri setindeki

ilişkileri anlamak ve yeni verileri sınıflandırmak için kullanılabilir (Yan-Yan SONG, 2015).

Karar ağacı öğrenimi, bir örneklem setinden elde edilen verilerle bir karar ağacı yapısı oluşturarak kesikli değerli hedef fonksiyonları tahmin etmek için kullanılan bir yöntemdir. Tom Mitchell'in 1997'deki çalışmasında bu algoritma, öğrenilen fonksiyonların karar ağaçları ile temsil edildiği ve bu ağaçlar aracılığıyla kesikli hedef fonksiyonlarının tahmin edildiği bir yöntem olarak ifade edilmiştir. (Mitchell, 1997). Karar ağaçları büyük veri tabanlarında öznitelikleri keşfetmek ve örüntüleri çıkarmak için önemli bir araç olarak kullanılabilir. Bu algoritma, eğitim kümesinin öznitelik uzayını bölerek bir karar ağacı oluşturur. Amaç, bilgilendirici ve sağlam bir hiyerarşik sınıflandırma modeli oluşturmak için öznitelik uzayını bölen bir dizi karar kuralını bulmaktır. Bu süreçte, her bir karar kuralı seçildikten sonra öznitelik uzayı iki ayrık alt uzaya bölünür. Daha sonra, elde edilen alt uzaylar tek bir sınıfa ait örnekleri içerecek şekilde bölünene kadar bölümleme işlemi devam eder. Bu işlem, özyinelemeli olarak alt uzaylara uygulanır. Bu özyinelemeli bölünme işlemi sonucunda elde edilen alt uzaylar, sınıflandırma işleminin hiyerarşik yapısını temsil eden bir karar ağacı oluşturur. Bu ağaç yapısı, veri setindeki öznitelikleri ve ilişkileri anlamak için kullanılabilir. Her bir düğüm, veri öznitelikleri ve bu özniteliklerin değerlerine göre yapılan kararları temsil eder. Sonuç olarak karar ağacı veri setindeki örüntüleri ve ilişkileri açıklayabilir, sınıflandırma veya tahmin yapmak için kullanılabilir (Anthony J. Myles, 2004). Sınıflandırma için ağaç oluşturulan karar ağaçlarında girdi verilerinin çıktı değeri bilinmemekte, öznitelik değerleri karar ağaçlarında test edilmektedir. Kökten yaprak düğümlerine kadar yol izlenmekte ve çıktı için sınıf tahminleri yapılmaktadır (Meryem Pulat, 2023) (Mitchell, 1997).

İşlem adımları basit ve hızlı olan karar ağaçları yorumlaması kolay olan bir algoritmadır. Bu nedenle sıklıkla kullanılmaktadır (Jiawei Han, 2022). Karar ağacı algoritmalarında veri setlerine art arda sorular sorulur ve öğrenme gerçekleştirilir (Quinlan, 1986) (Dangeti, 2017). Veri setindeki nitelikler öğrenme işlemi esnasındaki soruları belirlemektedir.

Karar ağaçlarının kökeni Von Neumann ve Morgenstern'in oyun teorisi üzerine yaptıkları çalışmalarla başlamıştır. Ancak bu ağaçların istatistik ve yapay öğrenme alanındaki önemi daha sonraki dönemlerde anlaşılmıştır. Karar ağacı yöntemini

istatistikte ilk defa hayata geçirenlerden biri Breiman ve arkadaşları olmuş ve bu çalışma 1984 yılına dayanmaktadır. Yapay öğrenme alanında ise karar ağacı yöntemini ilk kez kullanan isimlerden biri Quinlan'dır. Bu isimler, karar ağaçlarını sınıflandırma ve örüntüleri çıkarma amacıyla yaygın şekilde kullanmaya başlayan öncülerdendir. Karar ağaçlarına ek olarak bir dizi ağaç tabanlı algoritma geliştirilmiştir. Bu algoritmalar, karar ağaçlarından farklı felsefi yaklaşımları veya özellikleri temsil eder. Son yıllarda, birden çok sınıflandırıcının bir araya getirilmesiyle oluşturulan yöntemler veya "komiteler" olarak adlandırılan algoritmalar önem kazanmıştır. Bu komite yöntemleri, farklı sınıflandırıcıların bir araya gelerek daha güçlü bir tahmin modeli oluşturmasına dayanır ve genellikle daha yüksek performans ve genelleme yeteneği sunar. Bu yöntemlerin popülaritesi, farklı sınıflandırıcıların güçlü yönlerini birleştirerek daha sağlam ve güvenilir sonuçlar elde etme potansiyeli taşımalarından kaynaklanmaktadır (Şeyda Demirel, 2019).

Karar ağaçlarının oluşturulmasında en kritik adımlardan biri, ağaç yapısının dallanmasının hangi öznitelikler veya kriterlere göre yapılacağının belirlenmesidir. Her bir farklı kriter bir karar ağacı algoritmasına karşılık gelebilir. Bu kriterlerden bazıları aşağıda belirtilmiştir (Taşkın Kavzoğlu, 2010):

Bilgi Kazancı (Information Gain): Özniteliklerin dallanma noktalarının seçilmesinde kullanılan yaygın bir kriterdir. Bilgi kazancı, bir özniteliğin belirli bir düğümdeki veri kümesindeki belirsizliği azaltma potansiyelini ölçer. Daha fazla bilgi kazancı sağlayan öznitelikler öncelikli olarak tercih edilir.

Kazanç Oranı (Gain Ratio): Bilgi kazancının veri kümesinin homojenliği ile düzeltilmiş halidir. Bu kriter, bilgi kazancının veri kümesinin entropisi ile orantılı olup daha dengeli dallanmış ağaçlar oluşturmayı hedefler.

Gini İndeksi: Her bir düğümdeki verinin homojenliğini ölçmek için kullanılır. Gini indeksi, bir düğümdeki veri kümesindeki farklı sınıflar arasındaki karışıklığı ölçer. Daha düşük Gini değeri, daha homojen bir düğümü temsil eder.

Ki-Kare İstatistiği (Chi-Square Statistic): Karar ağacı oluştururken bir özniteliğin ve hedef değişkenin birlikte bağımlılığını değerlendirmek için kullanılır. Bu istatistik, bağımsızlık hipotezini test eder.

Bu kriterler, karar ağaçlarının oluşturulmasında kullanılan farklı algoritmaların temelini oluşturur. Hangi kriterin kullanılacağı; veri setinin özelliklerine, algoritmanın tercihlerine ve problemin doğasına bağlı olarak değişebilir. Bu kriterlerin seçimi, ağacın yapısını ve sonuçlarını önemli ölçüde etkileyebilir.

Karar ağacının güçlü yönleri aşağıda sıralanmıştır (Jiawei Han, 2022) (Yan-Yan SONG, 2015):

- Sınıflandırıcılar genel olarak iyi bir doğruluğa sahiptir.
- Çok boyutlu veriler işlenebilir.
- Aykırı değerlere karşı sağlamdır.
- Hem nicel hem nitel değişkenlere karşı uyumludur.
- Karar ağacı parametre ayarı, sınıflandırıcıların oluşturulması ya da herhangi bir alan bilgisi gerektirmeyen, parametrik olmayan bir yaklaşımdır.
- Anlaşılabilir kurallar üretir.
- Diğer makine öğrenme algoritmalarına göre insanların karar verme süreçlerini daha fazla yansıtırlar.
- Bilgi ağaç biçiminde temsil edildiğinden konuya uzman olmayanlar bile kolayca yorumlayabilir.
- Aşırı hesaplama gerek duyulmadan sınıflandırma yapar.

Karar ağaçlarının dezavantajları arasında ise verilerde küçük bir değişiklik olması durumunda bile ağaçta büyük değişikliklere sebebiyet vermektedir (Gareth James, 2013).

3.3. GRADYAN ARTIRMA (GRADYAN BOOSTİNG)

Gradyan Boosting, zayıf tahmincileri güçlü tahminciler haline getirmeye odaklanan bir makine öğrenimi tekniğidir. Bu teknik, birden fazla zayıf modeli birleştirerek güçlü bir topluluk modeli oluşturmayı hedefler. Karar ağaçları, kök düğümden başlayarak yapraklara doğru dallanmalar oluşturan bir yapıya sahiptir. Gradyan Boosting algoritmasında ise, ilk olarak bir karar ağacı ile bir tahmin yapılır ve elde edilen tahmin ile hedef arasındaki fark hesaplanır. Bu fark, bir sonraki ağacın oluşturulmasında kullanılır. Gradyan boosting yöntemi, bu farkı en aza indirmeyi hedefler. Her bir yinelemede, bu farkı en aza indirmek için yeni bir karar ağacı eklenir.

Bu eklenen ağaçlar, öğrenme hızını belirleyen bir küçülme parametresi α kullanılarak küçültülebilir. Yani Gradyan Boosting algoritması, her yinelemede hedef ile tahmin arasındaki farkı azaltmaya odaklanarak yeni karar ağaçları ekler. Öğrenme hızı parametresi α, eklenen her ağacın etkisini kontrol eder ve bu parametre doğru ayarlandığında algoritma daha iyi sonuçlar elde edebilir. Bu şekilde Gradyan Boosting, ardışık adımlarda modelin performansını artırarak güçlü bir tahminci elde etmeyi amaçlar (Tuğçe Yüce, 2021).

Gradyan artırma algoritması, başlangıçta bir tahmin fonksiyonu oluşturur. Bu tahminlerle gerçek gözlemler arasındaki fark, bir kayıp fonksiyonu yardımıyla hesaplanır. Ardından, bu farkı en aza indirmek amacıyla yeni tahminler yapılır. İlk aşamada, tahminler yapıldıktan sonra gerçek gözlemlerle aralarındaki farklar hesaplanarak bir kayıp fonksiyonu elde edilir. İkinci aşamada, bu kayıp fonksiyonu ve yapılan tahminler birleştirilerek yeni tahminler yapılır ve bu süreç tekrarlanır. Art arda gerçekleştirilen bu işlemlerle tahmin fonksiyonunun başarı düzeyi artırılmaya çalışılır. Temel amaç, elde edilen tahminlerin gerçek gözlemlerle arasındaki farkın mümkün olduğunca sıfıra yakın olmasını sağlamaktır. Bu süreçte, her adımda yapılan tahminler öncekilerin hatalarını düzeltmeye çalışarak daha doğru sonuçlara ulaşmaya çabalar. Yani Gradyan artırma algoritması, ardışık adımlarla tahmin performansını artırarak istenen sonuca yaklaşmaya çalışır. Kayıp fonksiyonunun formülü aşağıda belirtilmiştir.

$$Loss\ Kayıp\ Fonksiyonu = MSE = \frac{1}{N}\sum(y_{\text{i}} - y_i^p)^2$$

Burada yer alan yi İ. gözlenen değeri, yi p ise i. tahmin değerini, L(yi, yi ^{p}) ise kayıp fonksiyonunu ve α değeri öğrenme oranını belirtir (Keleş vd., 2020).

$$y_i^p = y_i^p + \alpha * \delta \sum \frac{(y_{\text{i}} - y_i^p)^2}{\delta y_i^p}$$

$$y_i^p = y_i^p - \alpha * 2 * \sum (y_{\text{i}} - y_i^p)$$

Zayıf model, rastgele tahminden daha iyi performans gösteren ancak güçlü bir model kadar doğru olmayan bir model olarak tanımlanır. Gradyan Boosting algoritmasında, her yeni model bir öncekinin yaptığı hataları düzelterek zayıf modeller

yinelemeli olarak topluluğa eklenir. Bu, onu hem regresyon hem de sınıflandırma problemlerinde tahminlerin doğruluğunu geliştirmek için güçlü bir teknik yapar.

Teorik olarak Gradyan Boosting, topluluğa yinelemeli olarak yeni zayıf öğreniciler ekleyerek çalışır ve her yeni öğrenici öncekilerin yaptığı hataları düzeltir. Gradyan Boosting algoritması, tahmin edilen ve gerçek hedef değerler arasındaki farkı ölçen kayıp fonksiyonunu en aza indirmek için kullanılır (Karaköse, 2023).

Gradyan Boosting algoritması (GBA), regresyon ve sınıflandırma problemleri için kullanılan bir makine öğrenimi tekniğidir. Bu teknik, temel olarak karar ağaçları veya rastgele orman gibi zayıf tahmincilerden güçlü tahminciler oluşturmayı hedefler. GBA, adım adım ardışık olarak tahminler yaparak her adımda önceki tahminlerin hatalarını düzelterek bir sonraki tahmin modelini geliştirmeye odaklanır. Bu şekilde, bir dizi zayıf tahminciyi birleştirerek daha güçlü bir tahmin modeli elde edilir. Algoritma, Friedman ve arkadaşları tarafından geliştirilmiştir ve regresyon ile sınıflandırma problemleri için geniş bir kullanım alanına sahiptir. Özellikle karar ağaçları gibi temel öğrenme algoritmalarını kullanarak bu zayıf tahmincileri güçlü bir tahminciye dönüştürme amacı güder. Bu süreçte; her bir adımda oluşturulan yeni tahminciler, önceki tahmincilerin hatalarını düzelterek daha doğru sonuçlar üretmeye çalışır. Bu yolla; GBA, karmaşık ilişkileri yakalamak ve genellenebilir modeller oluşturmak için kullanışlı bir teknik haline gelmiştir (Şahinarslan, 2019). Boosting yöntemleri, zayıf tahmincilerin lineer olmayan bir şekilde bir araya getirilmesiyle güçlü bir tahminci oluşturmayı hedefler. Gradyan Boosting algoritması (GBA) da bu prensibi takip eder. Bu zayıf algoritmaların kullanılması rastgeleden biraz daha iyi sonuç verdikleri içindir (Freund, 1995). Karar ağaçları, bazı avantajlara sahip olmasına rağmen bazı dezavantajlar da barındırır. Özellikle, basit ağaçlar düşük karmaşıklıkta olduğu için büyük bir "bias" (yanlılık) yaratabilirler. Bu durum, modelin gerçekten karmaşık ve esnek olması gereken veri setlerini doğru bir şekilde öğrenememesine yol açabilir. Öte yandan, karmaşık ağaçlar çok fazla dallanma ve derinlikle oluşturulduğunda bu durum büyük bir varyansı (dalgalanma veya modelin verilere hassas tepkisi) artırabilir. Karmaşık ağaçlar, eğitim veri setine aşırı uyum sağlayarak (overfitting) yeni veriye genellenemeyen modeller üretebilir. Torbalama (bagging), bu varyansı azaltmak için kullanılan bir tekniktir. Bu yöntemde, bir veri kümesinden rastgele örnekler seçilir ve her örnekleme işlemi ile birbirinden farklı alt veri kümeleri oluşturulur. Bu alt kümeler üzerinde ayrı ayrı karar

ağaçları eğitilir ve bu ağaçların ortalaması ya da oylama yöntemiyle tahmin yapılır. Torbalama, her bir ağacın farklı bir alt veri kümesi üzerinde eğitilmesi sayesinde her ağacın farklı özellikleri vurgulamasını sağlar. Bu da ağaçların birbirlerini düzeltebilmesini ve genelde daha iyi bir tahmin performansı sağlayabilmesini mümkün kılar. Bu teknik aynı zamanda karar ağaçlarının oluşturduğu varyansı azaltarak daha istikrarlı ve genelleştirilebilir modeller elde edilmesine yardımcı olabilir. GBA regresyon modeli için uygulama adımları aşağıda sıralanmıştır (Kayalı, 2023):

- ✓ Regresyon ağacı oluşturulur.
- ✓ Her bir ağacın dalı için tahminler ile gözlemler arasındaki fark yani hata oranı hesaplanır.
- ✓ Yeni gözlem verileri olarak hesaplanan hata oranları kullanılmaktadır.
- ✓ Yeni ağaç oluşturularak hata oranlarını en aza indirmeyi hedefler.
- ✓ Hata oranı sıfıra yaklaşıncaya kadar bu adımlar tekrarlanır.

Diğer karar ağaçlarına benzer model bu algoritmada da kurulur. Zayıf halkalar birbirlerinin zayıf yönlerinden hataları çıkarır, ardından daha güçlü bir halka ortaya çıkar. Gradyan artırma algoritması (GBA), iteratif bir süreç izler. Bu algoritmanın ilk aşamasında, genellikle basit bir tahmin fonksiyonu veya zayıf bir öğrenme algoritması kullanılır. Bu zayıf öğrenme algoritması, genellikle karar ağaçları şeklinde olabilir ve bu ağaçlara "ağaç" adı verilir (Mustafa Berk Keleş, 2020). Sonradan oluşturulan ağaçlarla daha önce oluşturulan ağaçların hata oranları not edilir (Şahinarslan, 2019) (Hilmi Cenk Bayrakçı, 2021). Gözlemler ile tahminler arasındaki fark hesaplanır. Farklar, kayıp fonksiyonunu verir. İkinci iterasyonda ise tahmin ve kayıp fonksiyonları birleştirilerek tekrar tahmin değeri elde edilir. Gözlemlerle tahminler arasındaki fark ortaya konur. Tahmin fonksiyonuna eklemeler yapılarak başarı artırılmaya çalışılır. Bununla birlikte hata oranının sıfıra yaklaşması amaçlanır (Mustafa Berk Keleş, 2020) (Hilmi Cenk Bayrakçı, 2021).

3.4. RASTGELE ORMAN (RANDOM FOREST)

Rastgele Orman, sınıflandırma ve regresyon problemleri için kullanılan etkili bir topluluk öğrenme yöntemidir. Bu yöntem, birbirinden bağımsız olarak birçok karar ağacının bir araya gelmesiyle oluşturulan bir karar ağaçları koleksiyonudur.

Rastgele Orman modeli temelde bir koleksiyon veya topluluk olarak karar ağaçlarına dayanır. Karar ağaçları, veri setini özelliklerine göre bölerek bir dizi karar düğümü ve sonunda yaprak düğümleri (sonuçları) içeren bir yapı oluşturur. Her bir ağaç, veri setindeki değişkenleri kullanarak homojen alt gruplara ayırmayı hedefler. Bu ağaçlar, her bir düğümdeki belirli bir özellik veya değişken kullanılarak veri kümesini bölerek homojen alt gruplara ayırır. Amaç, her bir bölünmenin (her bir düğümün) olabildiğince homojen olmasını sağlamaktır. Yani o düğümdeki veriler mümkün olduğunca benzer niteliklere sahip olmalıdır. Karar ağaçları, hedef değişkeni (örneğin, sınıflandırma için bir etiket veya regresyon için bir sayısal değer) en iyi şekilde ayıran değişkenleri ve bu değişkenlerin değerlerini bulmaya çalışır. Her bir ağaç, veri setinin farklı özelliklerini ve örneklerini baz alarak birbirinden bağımsız olarak oluşturulur. Rastgele Orman modeli bu ağaçları bir araya getirerek her bir ağacın bağımsız tahminlerini birleştirir ve çoğunluk oyu alarak sınıflandırma veya regresyon yapar. Bu kolektif öğrenme yöntemi, birçok zayıf tahminciyi bir araya getirerek genellikle daha güçlü ve genelleştirilebilir bir model elde etmeye çalışır. (Luiz Pereira, 2018).

CART (Classification and Regression Tree), sınıflandırma ve regresyon problemlerine yönelik olarak kullanılan bir ağaç tabanlı modelleme yöntemidir. Bu yöntem, veriyi belirli bir hedef değişkeni bölerek homojen alt gruplara ayırmayı hedefler. Tamamını iki dala bölmeyi değil, daha karmaşık yapılar oluşturmayı amaçlar. CART, her bir ağaç yapısını bir değişken kullanarak veri setini bölerek oluşturur. Bu bölünmeler, veri setini en iyi şekilde ayırmak için belirli bir kriter kullanılarak gerçekleştirilir. Örneğin, sınıflandırma problemlerinde genellikle Gini impurity veya entropy gibi kriterler kullanılırken regresyon problemlerinde ise verinin varyansı gibi kriterler değerlendirilebilir. Veri seti içerisindeki değişkenlerin her biri, en iyi bölünmeyi sağlayacak şekilde değerlendirilir ve bu değişkenlerin değerleri ile bölünmeler yapılır. Bu işlem, ağacın dallara doğru büyümesini sağlar. Her bir bölünme noktası, homojen alt grupları oluşturacak şekilde belirlenir. CART, bu şekilde ağacı oluştururken veri setini sınıflandırır veya regresyon yapar. Veri setinin bölünmesi ve homojen alt gruplara ayrılması, hedef değişkenin tahmin edilmesini sağlar. Bu yöntem; veri analizi, sınıflandırma ve regresyon problemlerinde yaygın olarak kullanılan etkili bir modelleme yöntemidir (Silahtaroğlu, 2018). Rastgele ormanlar CART'a benzer bir karar ağacı yapısı kullanır ancak bazı temel farklılıkları vardır. Bu farklılıklar aşağıda belirtilmiştir:

Veri Alt Kümeleri ve Özellik Seçimi: Rastgele ormanlar, her bir ağacı eğitirken rastgele alt küme seçimi yapar. Bu, her ağacın farklı alt kümelerde eğitilmesini sağlar ve böylece çeşitlilik artar. Her ağaç için özelliklerin seçimi de rastgele yapılır. Her bir düğümde en iyi bölünmeyi yapmak için sadece belirli bir rastgele özellik alt kümesi değerlendirilir.

Ağaçların Bağımsızlığı: Rastgele ormanlar, her bir ağacı birbirinden bağımsız olarak eğitir. Her ağaç, kendi rastgele veri alt kümesi ve özellik alt kümesi üzerinde bağımsız olarak çalışır.

Topluluk Öğrenmesi: Rastgele ormanlar, bir topluluk öğrenme yöntemi olarak çalışır. Birden fazla karar ağacının tahminlerinin birleştirilmesiyle genellikle daha güçlü ve genelleştirilebilir bir model elde edilir.

Karar Yolu Çoğunluğu: Sınıflandırma durumunda, rastgele ormanlar genellikle çoğunluk oyu alarak tahmin yapar. Her bir ağacın sınıf tahminleri alınır ve en çok oyu alan sınıf nihai tahmin olarak seçilir.

Regresyon Durumunda Ortalama Alma: Regresyon durumunda, her bir ağacın tahminleri alınır ve bu tahminlerin ortalaması genellikle nihai regresyon tahmini olarak kullanılır. Bu farklılıklar, rastgele ormanları CART'a göre daha genelleştirilebilir, daha az aşırı uymaya meyilli ve daha kararlı hale getirir. Bu yöntem, geniş bir özellik uzayında etkili ve güçlü bir tahmin modeli oluşturmak için kullanılır.

Rastgele orman, ağacın büyütülmesinde faydalanılan eğitim verilerinin tamamını kullanmayarak bunun yerine “Torbalama” yönteminden örneklemeler yapar. Torbalama yöntemi, aynı makine öğrenme algoritmasının farklı örneklemlerle eğitildiği ve sonuçların bir araya getirildiği bir yöntemdir. Temel fikir, farklı örneklemler üzerinde eğitilmiş zayıf öğrenicileri (weak learner) bir araya getirerek daha güçlü bir model elde etmektir. Bu teknik, verilerin farklı alt kümeleri üzerinde eğitilmiş çoklu modellerin tahminlerinin ortalamasını alarak modelin varyansını azaltmaktadır (Karaköse, 2023).

$$f' = \frac{1}{B}\sum_{b}^{B} = 1fb\,(x')$$

fb: regresyon ya da sınıflandırma ağacı.

X: öğretme örneği.

B: öğretme seti (tüm ağaçları içerir.)

Ayrıca rastgelelik için önemli bir unsur ortaya konmaktadır. Ağaç yapımında düğümlerin bölünmesinde kullanılan değişkenlerin tamamen ya da kısmen seçilmesi gerekmektedir. Buna örnek olarak her düğüm bölücünün tamamen rastgele seçildiği bir ağaç gösterilebilir. Son aşamada ise karar ağacı en yüksek boyutuna dönüştürülerek budanmadan bırakılır. Hem rastgele orman hem de CART algoritmasında alt sınıflara ayırma ve bölünme yönteminde veri noksanlıkları nedeniyle bölünmeler tamamlanana kadar tekrar edilir. Büyük bir karar ağacının rastgele büyütülerek budanmaması akıllı bir modelleme uygulaması olarak gösterilmemektedir.

Rastgele orman yaklaşımdaki temel düşünce, modelin genel performansını artırmak ve varyansı azaltmak için çoklu karar ağaçlarının öngörülerini bir araya getirmektir. Tek bir karar ağacı, özellikle ağaç derin ve veriler gürültülü olduğunda, fazla uydurmaya eğilimlidir. Rastgele orman, çoklu karar ağaçlarının tahminlerinin ortalamasını alarak fazla uydurmayı azaltabilir ve genelleme performansını iyileştirebilir.

Rastgele orman modeli; ormandaki ağaç sayısı, her bölmede dikkate alınan özelliklerin sayısı, bir düğümü bölmek için gereken minimum örnek sayısı ve maksimum derinlik gibi performansını artırmak için ayarlanabilecek birkaç hiperparametreye sahiptir (Karaköse, 2023).

Rastgele Orman yöntemi, Ho ve Breiman'ın önerdiği Random Subspace ve Bagging (Bootstrap Aggregating) tekniklerinin birleşiminden faydalanır (Ho, 1998) (Breiman, Random Forests, 2001). Bagging, karar ağaçlarını bağımsız veri örnekleriyle oluşturarak her bir ağacın farklı veri alt kümeleri üzerinde eğitilmesini sağlar. Her ağaç, bağımsız bir örneklemle oluşturulur, bu da çeşitliliği artırır ve aşırı uyumu azaltır. Öte yandan Random Subspace tekniği karar ağacının her bir düğümünde en iyi bölünmeyi sağlayacak değişkenin rastgele seçilen bir alt kümesi arasından seçilmesini içerir. Bu da her bir ağacın sadece rastgele seçilen özellikler üzerinde eğitilmesini sağlar. Bu yöntem, ağaçların bağımsızlık derecesini artırarak çeşitliliği daha da artırır. Rastgele orman yöntemi, bu iki tekniği birleştirerek her ağacın bağımsız örneklem ve özellik alt kümeleri üzerinde eğitilmesini sağlayarak karar ağaçlarının çeşitliliğini artırır. Bu, her bir ağacın

diğerlerinden farklı bir bakış açısına ve modelleme şekline sahip olmasını sağlayarak daha güçlü bir öğrenici oluşturur. Bu çeşitlilik, genellikle daha iyi genelleme yapılmasına ve daha az aşırı uyuma sahip olunmasına yardımcı olur.

Rastgele orman; biyoloji, tıp, finans gibi farklı alanlarda sıklıkla kullanılmaktadır. Genelde özellik önem tahmini, anormallik tespiti ve görüntü sınıflandırma gibi görevler için kullanılmaktadır.

Rastgele Orman algoritması, birçok özelliğe sahip büyük veri kümeleri üzerindeki iyi performansıyla bilinir. Hem kategorik hem de sayısal özellikleri işleyebilir ve tek bir karar ağacına göre fazla uydurmaya daha az eğilimlidir. Ayrıca yorumlaması kolaydır ve yerleşik bir özellik önem ölçüsüne sahiptir. Bu algoritma eğitim verileri arasında rastgele alınmış bir alt kümeden bir dizi karar ağacı meydana getiren topluluk öğrenme yöntemleri arasındadır. Son tahmin, her karar ağacının tahminlerinin ortalaması alınarak yapılır (Karaköse, 2023).

Algoritma aşağıdaki adımlarda çalışır (Hastie, 2009):

- Ağaç Sayısı ve Örneklem Seçimi: Rastgele Orman modelinde, ağaç sayısı (n_estimators) ve her bir ağaç için kullanılacak örneklem sayısı (bootstrap örnekleme) gibi önemli parametreleri seçmek önemlidir. Daha fazla ağaç genellikle daha iyi genelleme performansına katkıda bulunabilir ancak belirli bir noktadan sonra artan ağaç sayısı, modelin hesaplama maliyetini artırabilir. Genellikle, artan ağaç sayısıyla birlikte modelin daha güçlü ve genelleştirilebilir hale geldiği gözlemlenir. Her bir ağacın oluşturulması için kullanılan örnekleme yöntemi olan bootstrap, veri setinden rastgele ve tekrarlı örnekler çekerek yeni alt veri setleri oluşturur. Bu, her ağacın farklı veri alt kümeleri üzerinde eğitilmesini sağlar. Bu örnekleme yöntemi, her bir ağacın birbirinden bağımsız olmasını ve çeşitliliğin artmasını sağlar. Bu parametrelerin seçimi, modelin performansını etkileyebilir. Optimal ağaç sayısı ve örneklem seçimi, genellikle çapraz doğrulama (cross-validation) gibi tekniklerle belirlenir. Veri seti için en iyi performansı elde etmek için farklı ağaç sayıları ve örneklem seçenekleri denenebilir ve performans metrikleri kullanılarak en iyi kombinasyon seçilebilir. Bu şekilde, modelin aşırı uyma veya az uyum gibi problemlerden kaçınılabilir ve daha iyi bir genelleme elde edilebilir.

Rastgele Orman algoritması, birçok karar ağacının bir araya getirilmesiyle çalışır. Her ağaç için belirli adımları izler, bu adımlar aşağıda belirtilmiştir:

- Veri Örnekleme: Her ağaç oluşturulurken bootstrap örnekleme yöntemiyle rastgele bir alt veri seti seçilir. Bu, veri setinden rastgele örnekler çekilerek alt veri setlerinin oluşturulması anlamına gelir.
- Özellik Seçimi: Rastgele Orman, her düğümde karar ağacının özelliği seçerken rastgele bir alt küme kullanır. Bu alt küme, toplam özelliklerden rastgele seçilen birkaç özelliği içerir. Böylece her ağaç farklı özellikler üzerinden eğitilir.
- Karar Ağacı Oluşturma: Örneklem veri seti ve seçilen özellik alt kümesi kullanılarak karar ağacı oluşturulur. Bu ağaç, veri setinin özniteliklerine göre tekrar tekrar bölünerek bir sınıflandırma veya regresyon modeli oluşturur.
- Tahmin: Tüm ağaçlar kullanılarak tahmin yapılır. Sınıflandırma durumunda, sınıf etiketleri çoğunluk oyu alınarak belirlenir. Regresyon durumunda ise ağaçların tahminleri ortalaması alınarak bir ortalama değer hesaplanır.
- Önem Derecelendirmesi: Her bir özelliğin model performansındaki etkisini ölçmek için bir önem derecelendirmesi yapılır. Bu, hangi özelliklerin daha bilgilendirici olduğunu ve tahminler üzerinde daha fazla etkiye sahip olduğunu belirlemeye yardımcı olur. Önem derecelendirmesi, özelliklerin bölünme ölçütleri ve hata azaltma miktarlarına dayanarak hesaplanır.
- Hiperparametre Ayarı: Rastgele Orman'ın çeşitli hiperparametreleri vardır. Ağaç sayısı, maksimum derinlik, minimum bölünme büyüklüğü gibi parametrelerin doğru ayarlanması modelin performansını ve genelleme yeteneğini etkiler. Bu parametrelerin optimize edilmesi genellikle çapraz doğrulama veya benzeri tekniklerle yapılır.

Denetimsiz Rastgele Orman: Rastgele Orman, ilk olarak denetimli öğrenmede kullanılır. Fakat bazı durumlarda denetimsiz öğrenmede de kullanılmaktadır. Bu yöntemdeki amaç verilerdeki yapı ve kalıpları veriler kullanılmadan bulmaktır. Rastgele Orman bağlamında; denetimsiz öğrenme, kümeleme veya boyut indirgeme gerçekleştirmek için kullanılabilir (Karaköse, 2023).

Kümeleme: Rastgele Orman algoritması aslen bir sınıflandırma ve regresyon tekniği olarak bilinir. Bununla birlikte, eğitim sırasında bu algoritma ağaçları veri setinin yapısını öğrenmek için kullanır. Her bir ağaç, veri noktalarını gruplamak için kullanılan

bölme ölçütleri ve dallanma kuralları oluşturur. Bu nedenle rastgele orman modeli, verileri kümelere ayırmak için dolaylı olarak kullanılabilir. Veri setindeki benzerlikleri ve farklılıkları ölçen bu yapı, özellikle veri setinin yapısını anlamak ve benzer özelliklere sahip olanları gruplamak için kullanılabilir. Ancak doğrudan bir kümeleme tekniği olarak tasarlanmamıştır ve öncelikle sınıflandırma veya regresyon amaçları için geliştirilmiştir. (Breiman, Random Forests, 2001)

Boyut azaltma: Rastgele Orman algoritması, özelliklerin önem derecelendirmesini yaparak boyut indirgeme işleminde kullanılabilir. Özelliklerin önem sırasına göre sıralanmasıyla daha az sayıda ancak daha önemli olan özellikler belirlenebilir. Bu önemli özellikler, verinin temsil edilmesi için kullanılabilir. Örneğin, en önemli özellikler seçilerek yeni bir özellik seti oluşturulabilir ve verinin boyutu azaltılabilir. Bu, veri setindeki karmaşıklığı azaltarak sınıflandırma veya regresyon gibi görevler için daha optimize edilmiş bir veri temsili sağlayabilir. Bu şekilde rastgele orman, veri setindeki önemli bilgileri koruyarak boyut indirgeme işleminde kullanılabilir. (Breiman, Random Forests, 2001).

Kernel Rastgele Ormanı: Kernel rastgele ormanı (KRF), rastgele orman algoritmasının genişletilmiş bir versiyonudur ve özellikler arasındaki ilişkileri öğrenmek için kernel yöntemlerini kullanır. Bu yöntemde, öznitelik uzayı bir kernel fonksiyonu aracılığıyla daha yüksek boyutlu bir uzaya dönüştürülür. Karar ağaçları, bu dönüştürülmüş öznitelikleri kullanarak genişletilir ve modelin karmaşıklığını artırmak için bu dönüştürülmüş uzayda işlem yapar. Bu sayede, özelleştirilmiş bir öznitelik uzayında çalışarak özgün rastgele orman algoritmasından daha karmaşık ve esnek ilişkileri modelleyebilir. Bu, özellikle özelleştirilmiş bir öznitelik uzayı gerektiren veri setlerinde daha iyi performans sağlayabilir. (Erwan Scornet, 2015).

Kernel fonksiyonları, özellik uzayını dönüştürmede kullanılan matematiksel fonksiyonlardır. Sıklıkla kullanılan kernel işlevleri arasında sigmoid çekirdeği, polinom çekirdeği ve radyal temel işlev (RTİ) kernel'i bulunmaktadır. Bu işlevin seçimi çözülmekte olan soruna ve verilerin doğasına bağlıdır.

KRF'de; öznitelik alanı bir kernel işlevi kullanılarak dönüştürülmekte ve ormandaki karar ağaçları, dönüştürülen öznitelikler kullanılarak büyütülmektedir. Karar ağaçlarındaki bölmeler, dönüştürülmüş özniteliklere dayalıdır ve tahmin tek tek ağaçlardan elde edilen tahminlerin birleştirilmesiyle yapılmaktadır (Karaköse, 2023).

Matematiksel açıdan, x bir özellik vektörü ve k(x,x') özellik uzayını dönüştüren bir kernel fonksiyonu olduğu varsayılsın. Dönüştürülen özellik alanı aşağıdaki şekilde verilebilir (Erwan Scornet, 2015):

$$\varphi(x)=[k(x,x1),k(x,x2),...,k(x,xn)]$$

Burada x1, x2, ..., xn eğitim örnekleridir. Ormandaki karar ağaçları, dönüştürülmüş öznitelikler φ(x) kullanılarak büyütülür ve tahmin, tek tek ağaçlardan gelen tahminlerin birleştirilmesiyle yapılır. Kernel rastgele ormanı, öğrenilecek veriler arasında doğrusal olmayan ilişkilere izin verdiği için doğrusal olmayan sınıflandırma ve regresyon problemleri için güçlü bir araçtır. Bununla birlikte, kernel işlevini kullanarak özellik uzayını dönüştürmeyi içerdiğinden hesaplama açısından geleneksel rastgele orman algoritmasından daha maliyetlidir.

3.5. DESTEK VEKTÖR MAKİNALARI (SUPPORT VECTOR MACHİNE-SVM)

Destek vektör makinesinden farklı sınıflandırma problemlerinin çözümünde faydalanılmaktadır (Maulud Dastan, 2020). Destek vektör makinaları, sınıflar arasındaki en iyi ayrımı yapabilmek için veri noktaları arasında bir "marj" bulmaya çalışır. Bu marj, sınıfları birbirinden en iyi şekilde ayıran hiperdüzlemi belirler. Bu hiperdüzlem, sınıflar arasındaki mesafeyi (marjı) maksimize etmeye çalışırken aynı zamanda sınırlardaki veri noktalarına denk gelen destek vektörleri de hesaba katar. Bu destek vektörleri, sınıflandırma sürecinde belirleyici olan noktalardır ve sınıflar arasındaki sınırdaki önemli veri noktalarını temsil eder. Sınıflandırma doğruluğu genellikle, sınıflar arasındaki marjın arttığı durumlarda artar. DVM, bu marjı maksimize edecek şekilde optimal bir ayrım hattı veya düzlemi oluşturarak sınıfları en iyi şekilde ayırmayı amaçlar. Hem sınıflandırma hem de regresyon problemlerinde kullanılabilen güçlü bir makine öğrenimi yöntemidir (Ibrahim Mahmood, 2021).

Destek vektör makineleri (DVM), sınıflar arasında bir ayrım oluşturmak için kenar boşlukları çizer. Bu kenar boşlukları, sınıflar arasındaki en büyük mesafeye (kenar) denk gelir. Temel amacı, sınıflar arasındaki ayrımı bu kenar boşluklarını maksimize edecek şekilde belirlemektir. Böylece sınıflar arasındaki bu maksimum mesafe, yeni veri noktalarının doğru bir şekilde sınıflandırılmasına yardımcı olur (Mahesh, 2018).

DVM'nin çalışma prensibi, veri noktalarını lineer olarak ayrılabilir hale getiremeyen durumlarda bile sınıfları ayırmak için yüksek boyutlu uzaylara dönüşüm sağlayarak çalışır. Bu, çekirdek fonksiyonları aracılığıyla gerçekleşir. DVM, lineer olarak ayrılabilir veri setleri için uygun bir hiperdüzlem bulabilir. Ancak veri seti lineer olarak ayrılamazsa çekirdek fonksiyonları kullanılarak veriler daha yüksek boyutlu bir uzaya dönüştürülür. Bu dönüşüm, orijinal uzayda lineer olarak ayrılamayan verilerin daha yüksek boyutlu uzayda lineer olarak ayrılabilir hale gelmesini sağlar. Örneğin, çekirdek fonksiyonları sayesinde doğrusal olmayan ayrımı olan veri noktaları da lineer bir hiperdüzlemde ayrılabilir hale getirilebilir. Bu dönüşüm, verilerin farklı boyutlu uzaylarda temsil edilmesini ve bu uzaylarda lineer ayrılabilirlik elde edilmesini sağlar. Bu, DVM'nin çok yönlülüğünü ve doğrusal olmayan ilişkilere sahip veri setlerinde bile etkili olabilmesini sağlar (Cristianini, 2013).

Destek vektör makineleri sınıflandırma ve regresyon işlemlerinde kullanılmakta ve genel olarak iki sınıfın üyelerinin ayırt edilebilmesi için kullanılabilecek en iyi sınıflandırma fonksiyonunu bulmayı amaçlamaktadır. Faklı bir ifade ile destek vektör makineleri marjini en yüksek doğruyu seçerek sınıflandırma hatasını en aza indirirken iki sınıfı ayırabilmek için öznitelikleri kullanarak sınıfları en uygun şekilde ayırabilecek bir düzlem veya hiper düzlem bulmaya çalışmaktadır (Görmez, 2021).

DVM'ler, iki farklı sınıfı birbirinden ayırmaya odaklanan sınıflandırma algoritmalarıdır. DVM'ler, sınıfları ayırmak için bir hiperdüzlem oluşturur. Bu hiperdüzlem, sınıflar arasındaki ayrımı en iyi şekilde gerçekleştirecek şekilde belirlenir. DVM, sınıflar arasında bir ayrım yapmak için destek vektörleri kullanır. Bu vektörler, hiperdüzleme olan uzaklıkları en küçük olan noktalardır ve sınıfları birbirinden ayıran hiperdüzlem üzerinde bulunurlar. DVM, bu destek vektörlerini kullanarak sınıfları birbirinden ayırmak için en uygun hiperdüzlemi belirler. Eğitim sırasında DVM, sınıflar arasında bir ayrım elde etmek ve sınırlayıcı bir hiperdüzlem oluşturmak için veriler arasında bir karar sınırı çizer. Bu sınıf ayrımı, verileri en iyi şekilde bölecek şekilde belirlenen bir hiperdüzlem üzerinde gerçekleşir. Eğitilen model daha sonra yeni veri noktalarını bu hiperdüzlem üzerinde sınıflandırmak için kullanılır. Bu yöntem, verilen bir girdi değerinin hangi sınıfa ait olduğunu belirlemede kullanılır ve sınıf ayrımını hiperdüzlem üzerinde gerçekleştirir (Serpil Gümüştekin Aydın, 2022).

DVM'ler genellikle başlangıçta doğrusal olarak ayrılabilen veri setlerinde kullanılır ancak çekirdek yöntemleri sayesinde doğrusal olarak ayrılamayan veri setleriyle de çalışabilirler. Doğrusal DVM, verilerin bir doğru veya düzlem ile ayrılabilir olduğu durumlarda kullanılır. Ancak gerçek dünya verileri genellikle doğrusal olarak ayrılamaz. Bu durumda çekirdek fonksiyonları devreye girer. Çekirdek fonksiyonları, verileri doğrusal olarak ayrılabilir hale getirebilmek için verileri daha yüksek boyutlu uzaylara dönüştürür. Bu, verilerin özelliklerini daha karmaşık bir şekilde ifade etmeye ve daha sonra doğrusal bir ayrım yapabilmeye olanak tanır.

Sigmoid, polinomiyal, doğrusal olmayan ve radyal tabanlı fonksiyonlar, çekirdek olarak kullanılabilen farklı fonksiyonlardır. Bu çekirdekler, verileri daha yüksek boyutlu uzaylara dönüştürerek doğrusal olarak ayrılabilir hale getirir. Böylece, doğrusal olarak ayrılamayan verileri de sınıflandırabilmek için DVM'ler bu çeşitli çekirdek fonksiyonlarını kullanabilir. Bu, DVM'nin doğrusal olarak ayrılabilen verilerden doğrusal olarak ayrılamayanlara kadar geniş bir yelpazedeki veri setleri üzerinde kullanılabilmesini sağlar (Alpaydın, 2011). Kısaca çekirdek fonksiyonları aracılığıyla doğrusal olmayan veri setlerinde büyük boyutlarda taşıma işlemleri yapılır.

Yukarıdaki açıklamalara göre destek vektör makinelerinde iki sınıfa ayırma işlemlerinden söz edilir. Ancak bu durum destek vektör makinelerinin ikiden daha fazla sınıf bulunan uygulamalar için kullanılmasının mümkün olmadığı anlamına gelmemektedir. Bundan dolayı DVM çoklu sınıftaki problemlerde de uygulanır. Bu noktada, sınıfların ayrılması için bire karşı hepsi yaklaşımıyla hareket edilebilmektedir. Şöyle ki öncelikle bir sınıf etiketi seçilerek bu sınıfa 1, diğer sınıflara -1 etiketi verilmektedir. Yani bir sınıf ile diğer tüm sınıfların karşılaştırması yapılmaktadır. Belirtilen bu işlemler sınıfların tamamı için uygulanarak sınıf sayısı ile eşit sınıflandırıcı elde edilir. Farklı bir ifade ile sınıf sayısı kadar destek vektör makinesi eğitilmekte, tahmin aşamasında ise her bir destek vektör makinesinin çıktısı karşılaştırılarak en güvenli sonuç tercih edilmektedir (Görmez, 2021).

Destek Vektör Makineleri (DVM), Vladimir Vapnik ve ekibi tarafından geliştirilen bir sınıflandırma ve regresyon modelleme tekniğidir. DVM, özellikle örneklem sayısının düşük olduğu durumlarda etkili olan istatistiksel bir öğrenme teorisine dayanır. DVM, özellikle küçük örneklem boyutlarıyla başa çıkabilen ve doğrusal olmayan ilişkileri

modellerken etkili olan bir algoritmadır (Sisodia, 2014). Destek Vektör Makineleri (DVM), karar sınırlarını belirleyen karar hiperdüzlemleriyle çalışır. Bu hiperdüzlemler, sınıfları birbirinden ayırmak için kullanılır ve veri noktalarını sınıflara göre bölen bir tür karar yüzeyidir (Silvestrov, 2016). Aşağıdaki şekilde gösterildiği üzere DVM sınıflandırma yapmak için iki sınıfı ayıran bir çizgi çizmekte ve bu çizginin ±1'ine kadar olan yeşil alana "marjin" denir. Marjin boşluğu ne kadar geniş olursa iki veya daha fazla sınıf o kadar iyi ayırır. DVM, son yıllarda örüntü sınıflandırma ve regresyon için son derece güçlü ve önemli araçlar olarak ortaya çıkmıştır. DVM, EEG sinyallerinin sınıflandırmasında yoğun olarak kullanılmaktadır. Depresyon hastaları tanınması (Zhou, 2022), insan vücudunun hareket amacını tahmin etmesi (Li, 2013) ve epileptik nöbetin sınıflandırılması (Weifeng Zhao, 2015) gibi ve daha pek çok alanda kullanılmaktadır.

Destek Vektör Makineleri, bir sınıflandırma algoritması olup istatistiksel öğrenme yöntemlerine dayanmaktadır. DVM kullanılarak gerçekleştirilen sınıflandırmalarda genellikle iki sınıfa ait örneklerin sınıf etiketleri {-1, +1} şeklinde gösterilir. Amacı, eğitim verisi kullanılarak elde edilen bir karar fonksiyonu yardımıyla bu iki sınıfın birbirinden en iyi şekilde ayrılmasını sağlamaktır. DVM, bu karar fonksiyonunu kullanarak eğitim verisini en uygun şekilde ayıracak bir hiper düzlem bulmaya çalışır. Bu hiper düzlem, sınıfları birbirinden en iyi şekilde ayırmak için kullanılmaktadır. İdeali, sınıfların birbirinden en uzak noktalardan geçen bir hiper düzlemle ayrılmasıdır. Bu hiper düzlem, destek vektörler adı verilen eğitim verisindeki önemli noktalarla belirlenir. DVM'ler sınıflar arasındaki ayrımı en iyi şekilde temsil eden ve hiper düzleme en yakın olan noktalardır (Gülşen, 2023).

DVM'ler, geniş bir uygulama yelpazesine sahip çok yönlü bir makine öğrenimi algoritmasıdır. DVM'nin bazı önemli kullanım alanları aşağıda belirtilmiştir (Chen, 2015):

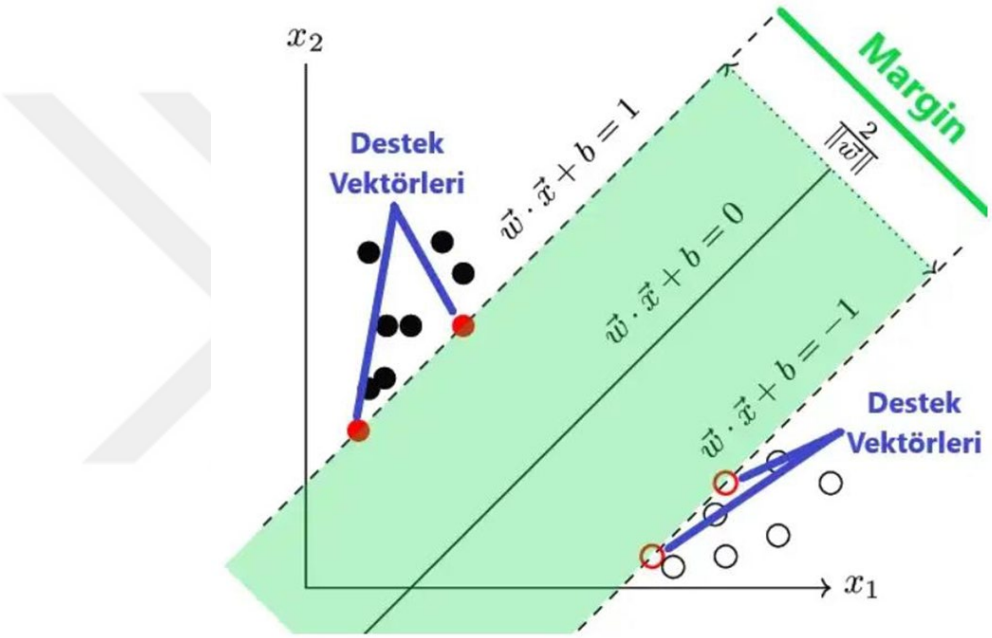
Sınıflandırma: Verileri belirli sınıflara ayırmak için kullanılır. Doğrusal olarak ayrılabilir verilerde etkili olmanın yanı sıra Kernel yöntemleriyle doğrusal olmayan verilerde de etkili sınıflandırmalar yapabilir.

Regresyon: Regresyon problemlerinde de kullanılabilir. Burada, veri noktaları arasındaki ilişkiyi tanımlamak için kullanılır.

Anormal Veri Tespiti: Anormal veri noktalarını tespit etmek için DVM kullanılabilir. Normal veri noktalarından farklı olan örüntüleri belirlemek için kullanılabilir.

Boyut İndirgeme: Özelliklerin karmaşıklığını azaltmak ve veri setlerindeki gereksiz özellikleri çıkarmak için boyut indirgeme için kullanılabilir. Özellik uzayını daha az boyutta temsil etmek için kullanılır.

Bu geniş uygulama alanı, DVM'in çok çeşitli problemlerde etkili bir şekilde kullanılmasını sağlar ve makine öğrenimi alanındaki önemini artırır.



Şekil 9. DVM algoritması kavramı (Nedir Bu Destek Vektör Makineleri? (Makine Öğrenmesi Serisi-2), 2020)

3.6. NAİVE BAYES

Naive Bayes algoritması, Bayes Teoremi'ne dayalı bir sınıflandırma algoritmasıdır. Bu algoritma, sınıf etiketlerinin olasılığını hesaplarken Bayes Teoremi'nin prensiplerini kullanır. Bu teorem; bir olayın gerçekleşme olasılığını, bu olayın meydana geldiği koşulların bilinmesi durumunda hesaplama yeteneğini sunar. Naive Bayes'in temel varsayımı, özellikler arasında bağımsızlık varsayımıdır (Kuru, 2023).

Naive Bayes (NB), istatiksel ve olasılıksal bir yöntem olup sınıflandırma algoritmasına dayalıdır. Özelliklerin tamamının son karara eşit bir biçimde katkı sunmasını sağlamadaki basitliği sebebi ile makine öğrenme uygulamaları arasında standart bir algoritmadır. Hesaplamadaki verimlilik bu basitliğe denktir ve yaklaşımı

farklı alanlar için uygun hale getirir. Naive Bayes'in temelinde sınıf koşullu olasılıkların hesaplanması yatmaktadır. Bu algoritma, büyük veri kümeleri üzerinde etkili olabilir. Çünkü hesaplamalarının basitliği ve hızı büyük veri setlerinde uygulanabilirliğini artırabilir. Ayrıca bu algoritma hem ikili sınıflandırma hem de çok sınıflı sınıflandırma problemleri için uygundur. Düşük eğitim verisi gerekliliği, Naive Bayes'in çoğu durumda diğer algoritmalara göre daha az eğitim verisi ile iyi sonuçlar alabilmesini sağlar. Bu özellik, veri toplama veya işleme aşamasında zaman ve kaynak tasarrufu sağlayabilir. Kesikli (kategorik) veya sürekli (sayısal) veri tipleriyle de çalışabilir olması, geniş bir veri yelpazesine uygulanabilirliğini artırır. Örneğin, metin tabanlı veri ile çalışabilirken aynı zamanda sayısal özelliklere de uygulanabilir. Bu algoritma spam filtreleme gibi istenmeyen e-postaları ayıklamak ve belgeleri sınıflandırmak gibi birçok uygulamada kullanılmaktadır. Naive Bayes'in basitliği ve performansı, bu tür uygulamalarda tercih edilmesine olanak sağlar (Ibrahim Mahmood, 2021).

Naive Bayes sınıflandırıcısı, özellikler arasında bağımsızlık varsayımı yapar. Bu; bir sınıftaki her özellik var olduğunda, diğer özelliklerin bu özelliğin varlığıyla doğrudan ilişkili olmadığını öngörür. Bu basit bağımsızlık varsayımı, özelliklerin bir sınıfa ait olma olasılığını hesaplarken kullanılır. Metin sınıflandırmasında sıkça kullanılmasının nedeni, metin verilerinin kelime veya terimlerle temsil edilmesi ve bu terimlerin sınıflandırma işleminde kullanılabilir olmasıdır. Örneğin; bir e-postanın spam veya spam olmayan olarak sınıflandırılması gibi durumlarda, e-postadaki belirli kelimelerin varlığı veya yokluğu sınıflandırmaya yardımcı olabilir. Kümeleme genellikle farklı bir makine öğrenimi yöntemi veya algoritma gerektirir. Bu, verilerin benzerliklerine göre gruplandırılmasını amaçlar. Naive Bayes sınıflandırıcısı, sınıflandırma problemleri için özellikle metin verileriyle çalışırken etkilidir (Mahesh, 2018).

Naive Bayes'in bellek gereksinimleri oldukça düşüktür çünkü eğitim ve sınıflandırma aşamalarında kullanılan veri miktarı oldukça sınırlıdır. Bu algoritma eğitim aşamasında, önceki ve koşullu olasılıkları depolamak için bir miktar bellek kullanır ancak bu genellikle çok fazla yer kaplamaz. Ayrıca Naive Bayes'in eksik verilerle başa çıkma becerisi oldukça iyidir. Bu; belirli bir özellik veya veri noktası için eksik değerler olduğunda, algoritmanın bu eksik verileri basitçe göz ardı etmesi ve tahmin yaparken doğal olarak bu eksikliği telafi etmesi anlamına gelir. Bu, Naive Bayes'in eksik veriyle çalışma kabiliyetini ve dayanıklılığını artırır. Ancak bu avantajlarına rağmen Naive

Bayes'in bağımsızlık varsayımı bazen gerçek dünyadaki veri setlerine tam olarak uymayabilir. Ayrıca eğer özellikler arasında güçlü bir bağımlılık varsa veya bazı özelliklerin sınıfı daha doğru bir şekilde tahmin ettiği durumlar olabilir, bu durumda Naive Bayes'in performansı düşebilir (Osisanwo, 2017).

Naive Bayes, önceden etiketlenmiş ve sınıflandırılmış veri kümesini kullanarak sınıf etiketlerinin olasılıklarını hesaplar. Bu hesaplamalar, yeni bir veri noktasının hangi sınıfa ait olma olasılığını belirlemek için kullanılır. Bu işlem, basit ve hesaplama açısından verimli olduğu için büyük veri setlerinde kullanılabilir. Naive Bayes'in gücü, verilerin karmaşıklığına rağmen etkili olabilmesi ve genellikle diğer sofistike sınıflandırma teknikleriyle rekabet edebilmesidir. Özellikle metin sınıflandırma gibi uygulamalarda, Naive Bayes'in basit yapısı ve hızlı çalışması tercih edilen bir algoritma haline gelmesine olanak tanır. Bununla birlikte bağımsızlık varsayımının gerçek veri setlerine tam olarak uymaması bazı durumlarda performansın düşmesine neden olabilir. Bu durumda, bazı durumlarda diğer daha karmaşık algoritmalar daha iyi sonuçlar verebilir (Görmez, 2021).

Naive Bayes sınıflandırma algoritması, makine öğrenimi uygulamalarında genellikle tercih edilen bir algoritmadır. Bu algoritmanın popülerliği; basit ve anlaşılabilir olmasından, hesaplama açısından verimli olmasına ve farklı alanlarda kullanılabilir olmasına dayanır. Naive Bayes'in temelinde Bayes teoremi ve sınıf koşullu olasılık yer alır. Algoritma, bir veri noktasının hangi sınıfa ait olduğunu belirlemek için bu olasılıkları hesaplar. Özellikle büyük veri kümeleriyle çalışırken hesaplama verimliliği Naive Bayes'i tercih edilebilir kılar. Bu algoritma, ikili ve çok sınıflı sınıflandırma problemleri için kullanılabilir. Ayrıca hem kesikli (discrete) hem de sürekli (continuous) veri türleriyle uyumlu çalışabilir. Eğitim verisi gereksinimi göreceli olarak düşük olduğu için Naive Bayes sınıflandırması, daha küçük eğitim veri setleriyle bile iyi performans gösterebilir. Pratikte Naive Bayes algoritması birçok alanda kullanılabilir. Örneğin; istenmeyen e-postaları filtrelemek, belgeleri sınıflandırmak, metin sınıflandırması gibi çeşitli uygulamalarda başarıyla kullanılabilir. Basitliği ve etkinliği nedeniyle, genellikle makine öğrenimi projelerinde ilk tercihlerden biri olabilir (Ibrahim Mahmood, 2021).

Naive Bayes, özellik vektörlerini ve bu vektörlere karşılık gelen sınıf etiketlerini içeren eğitim veri setlerini kullanarak özelliklerin olasılık dağılımlarını tahmin eder. Bir

veri noktası sınıflandırılmak istendiğinde, bu noktanın özelliklerine dayalı olarak her sınıfın olasılığını hesaplar ve en yüksek olasılığa sahip olan sınıfı seçer. Ancak Naive Bayes'in temel varsayımı, özellikler arasında bağımsızlık olduğunu varsayar. Yani bir özelliğin varlığı veya yokluğunun diğer özellikler üzerindeki etkisini dikkate almaz. Bu basit varsayım, yöntemi hızlı ve verimli hale getirirken gerçek dünyadaki durumlar bazen bu bağımsızlık varsayımına uymayabilir. Bununla birlikte pratikte Naive Bayes; doğal dil işleme gibi metin tabanlı problemlerde özellikle de spam filtreleme veya duygu analizi gibi alanlarda başarılı bir şekilde kullanılır. Ayrıca özellikler arasındaki gerçek bağımlılıkların bulunmadığı durumlarda da hızlı ve etkili bir sınıflandırma çözümü sunabilir (Leung, 2007).

Naive Bayes'in bellek gereksinimi oldukça düşüktür çünkü eğitim aşamasında genellikle minimum depolama alanı kullanır. Bu durum, sınıflandırma için gerekli olan önceki ve koşullu olasılıkları depolamak için gereken belleği içerir. Ancak Naive Bayes hesaplama sırasında olasılıkları basitçe göz ardı ettiğinden ve bu hesaplamaların nihai sınıflandırmaya etkisi olmadığından dolayı bu yöntem eksik değerlere karşı doğal olarak dayanıklıdır. Bu özellikler, özellikle büyük veri kümeleri veya sınıflandırma işlemleri için değerlidir. Çünkü Naive Bayes, düşük bellek kullanımı ve hesaplama açısından verimlidir. Eksik veya eksik verilere sahip olduğunda bu yöntem doğal olarak bu durumla baş edebilir ve etkili sonuçlar üretebilir. Bu, Naive Bayes'in pratik uygulamalarda tercih edilmesini sağlayan avantajlardan biridir (Osisanwo, 2017).

3.7. LOJİSTİK REGRESYON (LOGİSTİC REGRESYON)

Tahmine dayalı bir modelleme yöntemi olan regresyon analizinde bağımlı değişkenle bir ya da daha çok bağımsız değişken arasındaki ilişki araştırılır. Lojistik regresyon ise atama ve sınıflandırma işlemlerinde kullanılan bir regresyon yöntemidir. Lojistik regresyon, bir cevap değişkeninin beklenen değerlerini olasılık olarak tahmin etmek için açıklayıcı değişkenlere dayalı bir model oluşturmaktadır. Genel olarak iki kategorik sınıfın ayrımını yapmak için kullanılır ve sonucu olasılık olarak verir (Karaköse, 2023).

Lojistik regresyon; bağımlı değişkenin ikili, üçlü veya çoklu kategorilerde incelenebildiği durumlarda kullanılan bir istatistiksel modelleme yöntemidir. Bu yöntem,

bağımlı değişkenin kategorik olduğu durumlarda bağımlı değişken üzerindeki olasılıkları belirlemek için bağımsız değişkenleri kullanır (Özdamar, 2004).

Lojistik regresyon, değişkenler arasındaki ilişkileri incelemek ve bir yordama modeli oluşturmak için sıkça kullanılan bir istatistiksel yöntemdir. Doğrusal regresyonun aksine lojistik regresyon kategorik bağımlı değişkenlerle çalışabilir ve bağımsız değişkenlerin sürekli veya kategorik olabileceği bir analiz yöntemidir. Lojistik regresyon, doğrusal regresyonun aksine bazı varsayımlara bağımlı değildir. Doğrusal regresyonda olduğu gibi normallik, süreklilik, eş varyanslılık gibi varsayımlar gerektirmez. Bu özelliği, lojistik regresyonun geniş bir kullanım alanı bulmasını sağlar. Lojistik regresyon, kategorik veya sürekli özelliklere sahip bağımsız değişkenlerle çalışabilir. Ayrıca sürekli ve kategorik değişkenler bir arada kullanılabilir. Bağımlı değişken ise genellikle kategoriktir ve iki veya daha fazla sınıfa ait olabilir. Lojistik regresyon aynı zamanda esnek bir yapıya sahiptir. Örneğin, ihtiyaç halinde sürekli bir bağımlı değişken (yordanan değişken) kategorik bir forma dönüştürülebilir. Böylece lojistik regresyon bu dönüştürülmüş değişkeni kullanabilir. Bu özellikler lojistik regresyonu, geniş bir veri yelpazesi üzerinde etkili ve esnek bir sınıflandırma yöntemi haline getirir (Choi S, 2010).

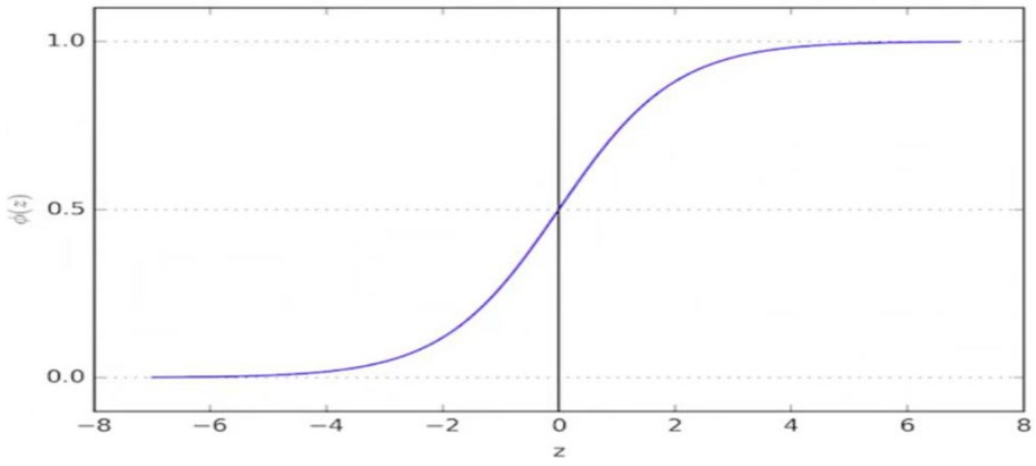
Lojistik regresyon, doğrusal olmayan ilişkileri ele alabilen bir yöntemdir. Bu model, bağımlı değişkenin (cevap değişkeni) ve bağımsız değişkenler arasındaki ilişkiyi logit fonksiyonu aracılığıyla modellemektedir. Logit fonksiyonu, doğrusal olmayan ilişkileri ifade etmek için kullanılır. Bu sayede lojistik regresyon doğrusal olmayan modeller oluşturabilir. Özellikle veri setinde doğrusal olmayan ilişkilerin var olduğu durumlarda, lojistik regresyon bu tür ilişkileri yakalamak ve modellemek için faydalıdır. Logaritmik dönüşümler gibi yöntemler kullanarak lojistik regresyon, doğrusal olmayan ilişkileri uygun şekilde ele alabilir ve modele entegre edebilir. Bu, modelin geniş bir veri yelpazesinde daha esnek ve uygun olmasını sağlar (Çokluk, 2010).

Bu algoritma isminde her ne kadar regresyon ifadesi olsa bile hem regresyon hem de sınıflandırmada kullanılmaktadır. Bu algoritma ile girdi parametresi olan öznitelikler yani bağımsız değişkenlerle örneklemlerin sınıfları yani bağımlı değişken tahmin edilmeye çalışılır. Farklı bir ifade ile lojistik regresyon algoritmasında kategorik olması gereken bağımlı değişkenlerle bağımsız değişkenler arasındaki ilişkinin bulunması ve en iyi şekilde modellenmesi amaçlanmaktadır. Ayrıca söz konusu algoritma; tahmin edilecek

bağımlı değişken iki kategoriye sahipse ikili regresyon, aralarında sıra bulunan ikiden fazla kategoriye sahipse sıralı lojistik regresyon, aralarında sıra bulunmayan ikiden fazla kategoriye sahipse çok kategorili lojistik regresyon olarak isimlendirilir (Görmez, 2021).

Lojistik regresyon, kategorik bir bağımlı değişkeni açıklamak için kullanılan bir yöntemdir. Doğrusal regresyonun aksine lojistik regresyonda bağımlı değişken kategoriktir ve genellikle iki veya daha fazla sınıfa ait olabilir. Bağımlı değişkenin bu kategorik yapısı, lojistik regresyonu doğrusal regresyondan ayıran temel özelliklerden biridir. Doğrusal regresyonda, bağımlı değişkenin değeri doğrudan kestirilirken lojistik regresyonda bağımlı değişkenin alabileceği farklı kategorilerden birinin gerçekleşme olasılığı tahmin edilir. Bu, örneğin bir olayın gerçekleşme olasılığı gibi durumlar için kullanışlıdır. Bu yöntem, belirli bir olayın meydana gelme olasılığını tahmin etmek veya bir özelliğin bir olayın gerçekleşmesi üzerindeki etkisini değerlendirmek için kullanılabilir. Lojistik regresyonun temeli aşağıdaki denklemde verilen lojistik fonksiyona dayanmaktadır ve lojistik fonksiyona ait grafik aşağıda verilmiştir (M. Erdal Balaban, 2018).

$$Ø\,(z) = \frac{1}{1+e^{-z}}$$



Şekil 10. Lojistik Fonksiyon (Logistic Function) (Logistic Regression: A Binary Classifier, 2023)

0 ile 1 arasında değer alabilen lojistik fonksiyon değerinin 0.5 ve üzerinde olması halinde 1'e yaklaştığı, aksi takdirde 0'a yaklaştığı öngörülmektedir. Böylece sınıf değerinin 1'e yakınsama olasılığı da hesaplanır. Olasılıkları hesaplanan sınıf değerlerini karşılaştırmak için üstünlük değeri (odds) hesaplanır (M. Erdal Balaban, 2018). P, bir

olayın olasılığı ise olayın üstünlük değerinin hesaplanması aşağıdaki formülle yapılmaktadır:

$$odds = \frac{\mathrm{p}}{1 - \mathrm{p}}$$

Lojistik regresyon algoritması ile ilgili dikkat edilmesi gereken konulardan biri bağımsız değişken olarak adlandırılan girdi parametrelerinin birbirinden de bağımsız olması yani aralarından yüksek düzeyde korelasyonun bulunmaması gerekliliğidir. Bu duruma dikkat edilmediği takdirde özniteliğin birbirinin fonksiyonu olarak yazılması durumunda eş doğrusallık meydana gelir. Bununla birlikte bu algoritma özniteliğin ölçeklendirilmesinde gereksinim duymaz. Fakat lojistik regresyon algoritması öznitelikler çok fazla olduğunda veya özniteliklerde kategorik değişkenler fazla olduğunda çok iyi performans gösterememektedir (Görmez, 2021).

Çok değişkenli regresyon ve lojistik regresyon arasında belirgin farklılıklar bulunmaktadır. Çok değişkenli regresyon, bağımlı değişkenin sürekli ve normal dağılıma sahip olduğu, bağımsız değişkenler arasında çoklu doğrusal ilişkilerin olmadığı, hata terimlerinin sıfır ortalamalı ve homoscedastic olduğu varsayımlarına dayanır. Bu varsayımların ihlali durumunda, çok değişkenli regresyon analizi doğru sonuçlar üretmeyebilir. Lojistik regresyon ise daha esnek bir yapıya sahiptir. Bu yöntemde, normal dağılım veya süreklilik gibi ön koşullar zorunlu tutulmaz. En büyük fark, lojistik regresyonun bağımlı değişkenin kategorik olduğu durumlarda kullanılmasıdır. Bu durumda, bağımlı değişken sınıflara ayrılır ve bu sınıflar arasında olasılık tahminleri yapılır. Doğrusal regresyonda bağımlı değişkenin değerinin tahmini yapılırken lojistik regresyonda ise bağımlı değişkenin belirli bir sınıfa ait olma olasılığı tahmin edilir. Bu özellik, lojistik regresyonun sınıflandırma problemlerinde yaygın olarak kullanılmasını sağlar (Önder Çoşkun, 2004).

Lojistik regresyon analizi, bağımsız değişkenlerin dağılımıyla ilgili özel bir varsayım içermez. Ancak kullanımıyla ilgili bazı pratik gereklilikler ve dikkat edilmesi gereken noktalar vardır. Özellikle bağımsız değişkenler arasında yüksek korelasyon veya çoklu doğrusal bağlantı olmamasına dikkat edilmelidir. Bu durum, lojistik regresyon modelinin tahmin gücünü azaltabilir veya modelin yanlılığına yol açabilir. Bu nedenle bağımsız değişkenler arasındaki ilişkiler dikkatlice incelenmeli ve gerektiğinde uygun önlemler alınmalıdır. Ayrıca veri setindeki kayıp veya uç değerler incelenmeli ve analiz

için uygun düzenlemeler yapılmalıdır. Kayıp veya uç değerler istatistiksel sonuçları etkileyebilir. Bu yüzden bu tür durumlar kontrol edilmeli ve gerekirse veri düzenlemesi yapılmalıdır. Bu adımlar, lojistik regresyon modelinin doğruluğunu ve güvenilirliğini artırmaya yardımcı olabilir.

Aşırı yayılım (heteroskedastisite), regresyon analizinde gözlenen varyansın beklenen varyanstan farklı olduğu durumu ifade eder. Bu durum, standart hataların dengesiz veya değişken olduğu anlamına gelir. Standart hataların yanlış bir şekilde küçük veya büyük olduğu durumlarda, regresyon modelindeki açıklayıcı değişkenlerin etkileri ve ilişkileri yanıltıcı olabilir. Aşırı yayılımın belirlenmesi ve düzeltilmesi önemlidir. Bu, verilerin dikkatlice incelenmesini ve aşırı yayılıma sahip gözlemlerin belirlenmesini gerektirir. Aşırı yayılımı olan gözlemlerin düzeltilmesi veya alternatif regresyon tekniklerinin kullanılması bu durumu ele almak için yaygın yöntemlerdir. Robust standart hata tahmincileri gibi yöntemler de heteroskedastisite problemini hafifletmek için kullanılabilir. Bu şekilde, regresyon modelinin daha güvenilir ve doğru sonuçlar vermesi sağlanabilir (Yılmaz Kaya, 2012). Lojistik regresyon; sosyal bilimlerde, tıpta, finansta ve pazarlamada çeşitli amaçlarla kullanılabilir. Örneğin; bir ilacın hastalık üzerindeki etkisini anlamak, bir reklam kampanyasının etkinliğini değerlendirmek veya bir ürünün satın alınma olasılığını tahmin etmek gibi birçok alanda kullanımı vardır. Bu model, değişkenler arasındaki ilişkinin doğrusal olmadığı durumlarla başa çıkabilir. Bu nedenle lojistik regresyon, ikili sonuçlarla ilişkili olup doğrusallık varsayımının sağlanmadığı durumlarda bile etkili bir şekilde kullanılabilir. Bu esneklik, gerçek dünya verilerinin çeşitliliğine uygunluğunu artırır ve birçok farklı analiz durumu için uygundur.

BÖLÜM 4: MATERYAL VE YÖNTEM

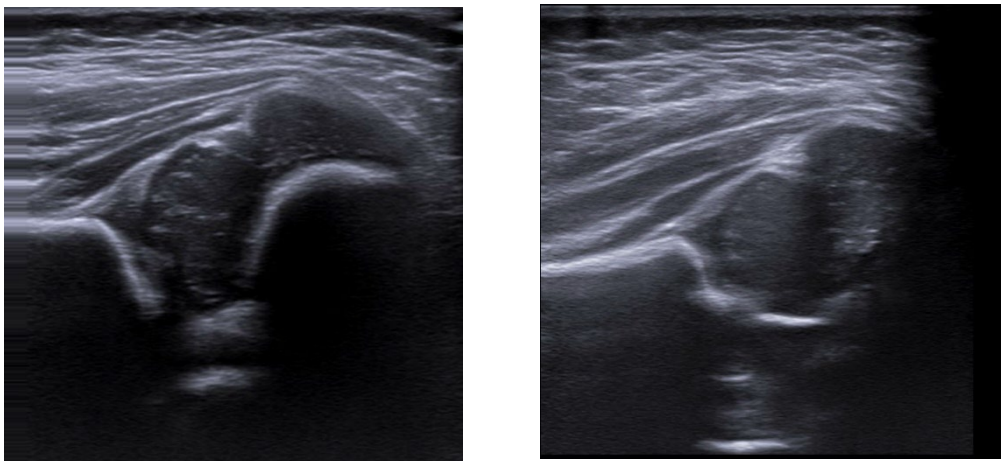
Veri seti oluştururken GKD kontrolü için gelen 100 kişinin ultrason kayıtları kullanılmıştır. Ultrason kayıtlarından alınan görüntülerin öznitelikleri önceden eğitilmiş derin öğrenme modelleri ile çıkarılmıştır. Çıkarılan özniteliklerden hazırlanan veri setinde "Doğru" ve "Yanlış" olmak üzere 2 sınıf bulunmaktadır. Bu veri seti; Random Forest, Support Vector Machine, Naive Bayes, Logistic Regresyon, KNN, Decision Tree ve Gradient Boosting makine öğrenme algoritmaları ile sınıflandırılmıştır.

Çalışma, Python dilinde Scikit-Learn (Sklearn), Pandas, Matplotlib, Seaborn kütüphaneleri kullanılarak yapılmıştır. Kullanılan bilgisayar; i7 12. nesil işlemci, 32 GB RAM, 1 TB M2 SSD donanımına sahiptir.

4.1. VERİ SETİ

Çalışmada kullanılan veri seti, bu çalışma için özel hazırlanmıştır. Yozgat Şehir Hastanesi Radyoloji ünitesinde kalça ultrasonu çekilen 100 bebeğin hem sağ hem de sol kalça ultrason kayıtları kullanılmıştır. Elde edilen 200 ultrason kayıtlarının her saniyesinden eşit aralıklarla 3 görüntü alınmıştır. Bu sayede bir bebekten 18 sağ ve 18 sol kalça görüntüsü olmak üzere 36 görüntü elde edilmiştir. Toplamda ise 3600 görüntü elde edilmiştir. Görüntülerin boyutları 581x607 pikseldir.

Çalışmada "Doğru" ve "Yanlış" olmak üzere 2 sınıf bulunmaktadır. Görüntüler radyoloji uzmanı tarafından sınıflandırılmıştır. 3600 görüntünün 1522 tanesi doğru, 2078 tanesi yanlış olarak sınıflandırılmıştır.



a) Doğru b) Yanlış

Şekil 11. Veri Setinden Görüntüler

GKD teşhisi için görüntünün doğru olabilmesi için dik iliak kemik, kemik asetabulumun en derin noktası, kıkırdak asetabulumun labrumu ve osteokondral bileşkenin görüntülenmesi gerekmektedir. Ayrıca görüntüde femur başının da görülebilmesi gerekmektedir.

4.1.1. Veri Çoğaltma

Veri çoğaltma, derin öğrenme ve makine öğrenme algoritmalarının başarısını yükseltmeyi ve kullanılan veri setlerini daha düzenli hale getirmeyi sağlayan bir yöntemdir (Jakup Nalepa, December 2019). Çalışmada veri çoğaltma yöntemi, sınıfların görüntü sayılarını eşitlemek amacıyla kullanılmıştır. 1522 olan doğru sınıfına ait olan görüntü sayısı 2078'e çıkarılarak doğru ve yanlış sınıflarına ait görüntü sayıları eşitlenmiştir.

Görüntü sayısı artırılırken veri çoğaltma yöntemlerinden dikey kaydırma, yatay kaydırma, kırpma ve büyütme teknikleri kullanılmıştır. Bu tekniklerin seçilmesindeki amaç görüntülerin, sınıflarına ait olan özelliklerini kaybetmeden kullanılmasını sağlamaktır. Bu yöntemle veri setindeki sınıflar arasındaki dengesizlik ortadan kaldırılmış ve kullanılan makine öğrenmesi algoritmalarının başarısının artırılması sağlanmıştır.

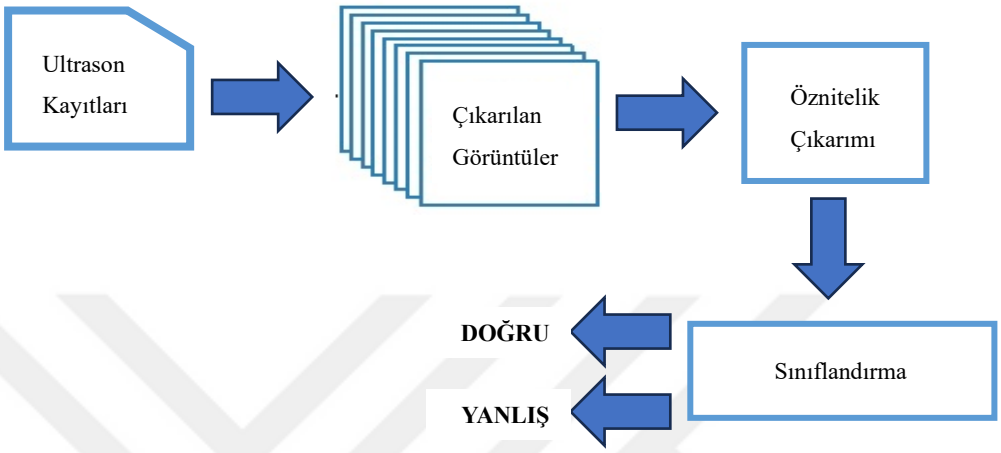
4.1.2. Öznitelik Çıkarımı

Öznitelik çıkarımı, makine öğrenmesinde yaygın olarak kullanılan bir süreçtir. Bir öğrenme algoritmasının uygulanması için verilerden elde edilebilen özelliklerin alt kümesi seçilir. Özellik seçiminin temel amacı, bir problem alanından minimal bir özellik alt kümesi belirlerken aynı zamanda problem alanını temsil etmede uygun şekilde doğruluğu korumaktır (Ladha L., 2011).

Ultrason kayıtlarından çıkarılan görüntülerin sayıları, veri çoğaltma yöntemleri kullanılarak eşitlendikten ve sayılar arasındaki dengesizlik giderildikten sonra önceden eğitilmiş derin öğrenme modelleri kullanılarak görüntülerin öznitelikleri çıkarılmıştır. Öznitelik çıkarımı için LBP yöntemi ve VGG16, VGG19, InceptionV3, MobileNet, DenseNet121, ResNet50 derin öğrenme modelleri kullanılmıştır. Her model ile görüntülerin öznitelikleri çıkarılmış ayrı dosyalar halinde kaydedilmiştir. Daha sonra her dosyaya sınıf özelliği de eklenmiştir. Bu sayede makine öğrenme algoritmaları ile kullanılacak veri setleri hazır hale getirilmiştir.

4.2. YÖNTEM

Şekil 12'de uygulama süreci gösterilmektedir. Ultrason kayıtları görüntülere çevrilmektedir. Daha sonra görüntülerin öznitelikleri çıkarılmakta ve sonrasında sınıflandırma yapılmaktadır.



Şekil 12. Uygulama Süreç Akışı

Çalışma için çalışma için özel olarak belirlenen ve ultrason cihazından alınan video kayıtları kullanılmıştır. Ultrason video kayıtları Python yazılım dilinde Opencv kütüphanesi kullanılarak her saniyeden 3 adet olacak şekilde görüntülere otomatik bir şekilde ayrılmıştır. Her ultrason kaydı adında bir klasör oluşturulmuş ve içerisine kaydedilmiştir. Bu görüntüler ortopedi uzmanı tarafından sınıflandırılmıştır. Daha sonra sınıflandırılan görüntülerin öznitelikleri LBP yöntemi ve VGG16, VGG19, InceptionV3, MobileNet, DenseNet121, ResNet50 önceden eğitilmiş derin öğrenme modelleri ile çıkarılmış ve her biri ayrı bir veri seti olarak kaydedilmiştir.

Çalışma ile GKD'nin teşhisi için en önemli ve tecrübe eksikliğinden kaynaklanan hata oranının yüksek olduğu aşama olan doğru görüntünün elde edilmesi aşamasındaki dezavantajların ortadan kaldırılması amaçlanmaktadır. Bu amaç gerçekleştirilirken başarısı yüksek ve aynı zamanda kaynak kullanımı açısından verimli bir çalışma ortaya koymak hedeflenmektedir. Bu hedef doğrultusunda görüntülerin özniteliklerinin daha doğru ve çeşitli olması, bununla birlikte Accuracy oranını yükseltmeye hizmet etmesi için önceden eğitilmiş derin öğrenme modelleri kullanılmış ve öznitelik çıkarımı gerçekleştirilmiştir. Elde edilen öznitelikler geleneksel makine öğrenmesi algoritmalarında kullanılmak üzere veri setine dönüştürülmüştür. LBP yöntemi ile 4156x11 boyutunda, VGG16 derin öğrenme modeli ile 4156x25089 boyutunda, VGG19

ile 4156x25089 boyutunda, InceptionV3 ile 4156x51201 boyutunda, MobileNet ile 4156x50177 boyutunda, DenseNet121 ile 4156x50177 boyutunda, ResNet50 ile 4156x100353 boyutunda görüntülerin özniteliklerinden oluşan veri setleri elde edilmiştir. Her veri setinin %80'i eğitim, %20'si test için kullanılmıştır.

Çalışmada; hazırlanan veri setlerinin sınıflandırılması için Random Forest, SVM, Naive Bayes, Logistic Regresyon, KNN, Decision Tree, Gradient Boosting makine öğrenmesi algoritmaları kullanılmıştır.

Random Forest algoritmasının parametreleri; n_estimators = 100, max_depth = 20, min_samples_split = 2, min_samples_leaf = 1, random_state = 42 olacak şekilde düzenlenmiştir. SVM algoritmasının parametreleri; kernel = 'linear', C = 1, gamma = 0.1 olacak şekilde düzenlenmiştir. Decision Tree algoritmasının parametreleri; criterion = 'gini', max_depth = 20, min_samples_spli t = 2, min_samples_leaf = 1, max_features = None olarak düzenlenmiştir. Gradient Boosting algoritmasının parametreleri; learning_rate = 0.1, max_depth = 7, n_estimators = 150 olarak düzenlenmiştir. Naive Bayes, Logistic Regresyon, KNN, algoritmalarının parametreleri varsayılan olarak kullanılmıştır. Algoritmalar kullanılırken çeşitli parametreler denenmiş ve en uygun parametrelerin bu şekilde olduğu görülmüştür.

Oluşturulan veri setlerinin her birine makine öğrenmesi algoritmaları uygulanmıştır. Sonuçlar; Accuracy, Precision, Recall, F1-Score metrikleri kullanılarak değerlendirilmiştir ve bunların karşılaştırmalı analizi yapılmıştır.

Çalışmada, Pandas, Scikit-Learn, Seaborn, MatplotLib, Psutil, Time kütüphaneleri kullanılmıştır. Pandas kütüphanesi ile veri setinin okunması ve ayrılması için kullanılmıştır. Scikit-Learn kütüphanesi, makine öğrenmesi algoritmalarını çalıştırmak için kullanılmıştır. Seaborn kütüphanesi, karmaşıklık matris tablosu oluşturmak için kullanılmıştır. MatplotLib kütüphanesi, karmaşıklı matris grafiğini çizdirmek için kullanılmıştır. Psutil ve time kütüphaneleri makine öğrenmesi algoritmalarının çalışma performanslarını ve sürelerini ölçmek için kullanılmıştır. Kodlama Spyder üzerinde Python dili ile yapılmıştır.

Modellerin kaynak kullanımlarının belirlenmesi amacıyla eğitim ve test aşamalarındaki işlemci kullanımları, bellek kullanımları, eğitim ve test süreleri ölçülmüştür. Çalışma Intel i7 12700H işlemcili, 32 Gb Ram ve Nvidia 3060 GPU'ya sahip

bir bilgisayar ile yapılmıştır. Bu bilgisayar üzerinden elde edilen her algoritmanın kaynak kullanımları değerlendirilmiş ve bunların karşılaştırmalı analizi yapılmıştır.

5. DENEYSEL SONUÇLAR

Çalışmada Accuracy, F1 Score, Precision, Recall metrikleri kullanılmıştır. VGG16, VGG19, InceptionV3, MobileNet, DenseNet121, ResNet50 önceden eğitilmiş derin öğrenme modelleri ve LBP yöntemi ile öznitelikleri çıkarılarak hazırlanan veri setlerinin her biri Random Forest, Support Vector Machine, Naive Bayes, Logistic Regresyon, KNN, Decision Tree ve Gradient Boosting makine öğrenmesi algoritmaları ile çalıştırılmış ve elde edilen Accuracy, F1 Score, Precision ve Recall değerleri karşılaştırılmıştır.

Öznitelik Çıkarılan Derin Öğrenme Modeli	**VGG 16**						
Model	**RF**	**SVM**	**NB**	**LR**	**KNN**	**DT**	**GB**
Accuracy	0,9014	0,8846	0,7368	0,8894	0,8726	0,8474	0,9123
Precision	0,9097	0,8953	0,6934	0,9	0,8733	0,8456	0,9153
Recall	0,9014	0,883	0,8922	0,8876	0,8853	0,867	0,9174
F1 Score	0,9055	0,8891	0,7803	0,8938	0,8793	0,8562	0,9164

Tablo 1. VGG16 Öznitelikleri ile Makine Öğrenmesi Performans Metrikleri

Tablo 1'de VGG16 derin öğrenme modeli ile öznitelikleri çıkarılan veri setine adı geçen makine öğrenmesi algoritmaları uygulandıktan sonra elde edilen performans metrikleri verilmiştir. En yüksek Accuracy oranına ulaşan makine öğrenmesi algoritması 0,9123 oranı ile Gradient Boosting olmuştur. Random Forest algoritması 0,9014 oranı ile en yüksek ikinci Accuracy oranına ulaşan algoritma olmuştur. En düşük Accuracy oranına sahip algoritma ise 0,7368 oranı ile Naive Bayes olmuştur. Precision, Recall ve F1 Score metriklerinde ise sıralama Accuracy oranına paralel olarak sonuçlar vermiştir.

Öznitelik Çıkarılan Derin Öğrenme Modeli	VGG19						
Model	**RF**	**SVM**	**NB**	**LR**	**KNN**	**DT**	**GB**
Accuracy	0,8918	0,881	0,7584	0,8822	0,875	0,8498	0,8918
Precision	0,9023	0,8965	0,7065	0,8949	0,8722	0,8526	0,9023
Recall	0,8899	0,8739	0,922	0,8784	0,8922	0,8624	0,8899
F1 Score	0,8961	0,885	0,8	0,8866	0,8821	0,8575	0,8961

Tablo 2. VGG19 Öznitelikleri ile Makine Öğrenmesi Performans Metrikleri

Tablo 2'de VGG19 derin öğrenme modeli ile öznitelikleri çıkarılan veri setine adı geçen makine öğrenmesi algoritmaları uygulandıktan sonra elde edilen performans metrikleri verilmiştir. En yüksek Accuracy oranına ulaşan makine öğrenmesi algoritmaları 0,8918 oranı ile Gradient Boosting ve Random Forest algoritmaları olmuştur. En düşük Accuracy oranına sahip algoritma ise 0,7584 oranı ile Naive Bayes olmuştur. Precision, Recall ve F1 Score metriklerinde ise sıralama Accuracy oranına paralel olarak sonuçlar vermiştir.

Öznitelik Çıkarılan Derin Öğrenme Modeli	InceptionV3						
Model	**RF**	**SVM**	**NB**	**LR**	**KNN**	**DT**	**GB**
Accuracy	0,851	0,8029	0,7272	0,8666	0,8798	0,7788	0,856
Precision	0,8697	0,8148	0,8119	0,8685	0,8767	0,7864	0,8737
Recall	0,8417	0,8073	0,6239	0,8784	0,8968	0,7936	0,8503
F1 Score	0,8555	0,8111	0,7056	0,8734	0,8866	0,79	0,8615

Tablo 3. InceptionV3 Öznitelikleri ile Makine Öğrenmesi Performans Metrikleri

Tablo 3'te InceptionV3 derin öğrenme modeli ile öznitelikleri çıkarılan veri setine adı geçen makine öğrenmesi algoritmaları uygulandıktan sonra elde edilen performans metrikleri verilmiştir. En yüksek Accuracy oranına ulaşan makine öğrenmesi algoritması 0,8798 oranı ile KNN olmuştur. Logistic Regresyon algoritması 0,8666 oranı ile en yüksek ikinci Accuracy oranına ulaşan algoritma olmuştur. En düşük Accuracy oranına sahip algoritma ise 0,7272 oranı ile Naive Bayes olmuştur. Precision, Recall ve F1 Score metriklerinde ise sıralama Accuracy oranına paralel olarak sonuçlar vermiştir.

Öznitelik Çıkarılan Derin Öğrenme Modeli	MobileNet						
Model	**RF**	**SVM**	**NB**	**LR**	**KNN**	**DT**	**GB**
Accuracy	0,8894	0,8618	0,7933	0,887	0,875	0,8077	0,9002
Precision	0,9095	0,8673	0,787	0,894	0,8739	0,8301	0,9057
Recall	0,8761	0,8693	0,8303	0,8899	0,8899	0,7959	0,9037
F1 Score	0,8925	0,8683	0,808	0,892	0,8818	0,8126	0,9047

Tablo 4. MobileNet Öznitelikleri ile Makine Öğrenmesi Performans Metrikleri

Tablo 4'te MobileNet derin öğrenme modeli ile öznitelikleri çıkarılan veri setine adı geçen makine öğrenmesi algoritmaları uygulandıktan sonra elde edilen performans metrikleri verilmiştir. En yüksek Accuracy oranına ulaşan makine öğrenmesi algoritması 0,9002 oranı ile Gradient Boosting olmuştur. Random Forest algoritması 0,8894 oranı ile en yüksek ikinci Accuracy oranına ulaşan algoritma olmuştur. En düşük Accuracy oranına sahip algoritma ise 0,8077 oranı ile Decision Tree olmuştur. Precision, Recall ve F1 Score metriklerinde ise sıralama Accuracy oranına paralel olarak sonuçlar vermiştir.

Öznitelik Çıkarılan Derin Öğrenme Modeli	DenseNet121						
Model	**RF**	**SVM**	**NB**	**LR**	**KNN**	**DT**	**GB**
Accuracy	0,8846	0,8522	0,7921	0,8918	0,8846	0,8041	0,8858
Precision	0,8917	0,8917	0,7657	0,9139	0,8846	0,8227	0,9069
Recall	0,8876	0,8876	0,8693	0,8761	0,8968	0,7982	0,8716
F1 Score	0,8897	0,8897	0,8142	0,8946	0,8907	0,8716	0,8889

Tablo 5. DenseNet121 Öznitelikleri ile Makine Öğrenmesi Performans Metrikleri

Tablo 5'te DenseNet121 derin öğrenme modeli ile öznitelikleri çıkarılan veri setine adı geçen makine öğrenmesi algoritmaları uygulandıktan sonra elde edilen performans metrikleri verilmiştir. En yüksek Accuracy oranına ulaşan makine öğrenmesi algoritması 0,8918 oranı ile Logisctic Regresyon olmuştur. Gradient Boosting algoritması 0,8858 oranı ile en yüksek ikinci Accuracy oranına ulaşan algoritma olmuştur. En düşük Accuracy oranına sahip algoritma ise 0,7921 oranı ile Naive Bayes Machine olmuştur. Precision, Recall ve F1 Score metriklerinde ise sıralama Accuracy oranına paralel olarak sonuçlar vermiştir.

Öznitelik Çıkarılan Derin Öğrenme Modeli	ResNet50						
Model	**RF**	**SVM**	**NB**	**LR**	**KNN**	**DT**	**GB**
Accuracy	0,8918	0,8822	0,6851	0,8954	0,8642	0,8257	0,887
Precision	0,8986	0,8722	0,6272	0,8939	0,8458	0,8226	0,8958
Recall	0,8945	0,9083	0,9839	0,9083	0,906	0,8509	0,8876
F1 Score	0,8966	0,8899	0,7661	0,901	0,8749	0,8365	0,8917

Tablo 6. ResNet50 Öznitelikleri ile Makine Öğrenmesi Performans Metrikleri

Tablo 6'da ResNet50 derin öğrenme modeli ile öznitelikleri çıkarılan veri setine adı geçen makine öğrenmesi algoritmaları uygulandıktan sonra elde edilen performans metrikleri verilmiştir. En yüksek Accuracy oranına ulaşan makine öğrenmesi algoritması 0,8954 oranı ile Logistic Regresyon olmuştur. Random Forest algoritması 0,8918 oranı ile en yüksek ikinci Accuracy oranına ulaşan algoritma olmuştur. En düşük Accuracy oranına sahip algoritmalar ise 0,6851 oranı ile Naive Bayes olmuştur. Precision, Recall ve F1 Score metriklerinde ise sıralama Accuracy oranına paralel olarak sonuçlar vermiştir.

Öznitelik Çıkarılan Yöntem	LBP						
Model	**RF**	**SVM**	**NB**	**LR**	**KNN**	**DT**	**GB**
Accuracy	0,8762	0,5853	0,6214	0,6478	0,8474	0,8137	0,8618
Precision	0,8805	0,5970	0,6441	0,6427	0,8440	0,8179	0,8575
Recall	0,8561	0,3965	0,4571	0,5859	0,8333	0,7828	0,8510
F1 Score	0,8681	0,4765	0,5347	0,6129	0,8386	0,8000	0,8542

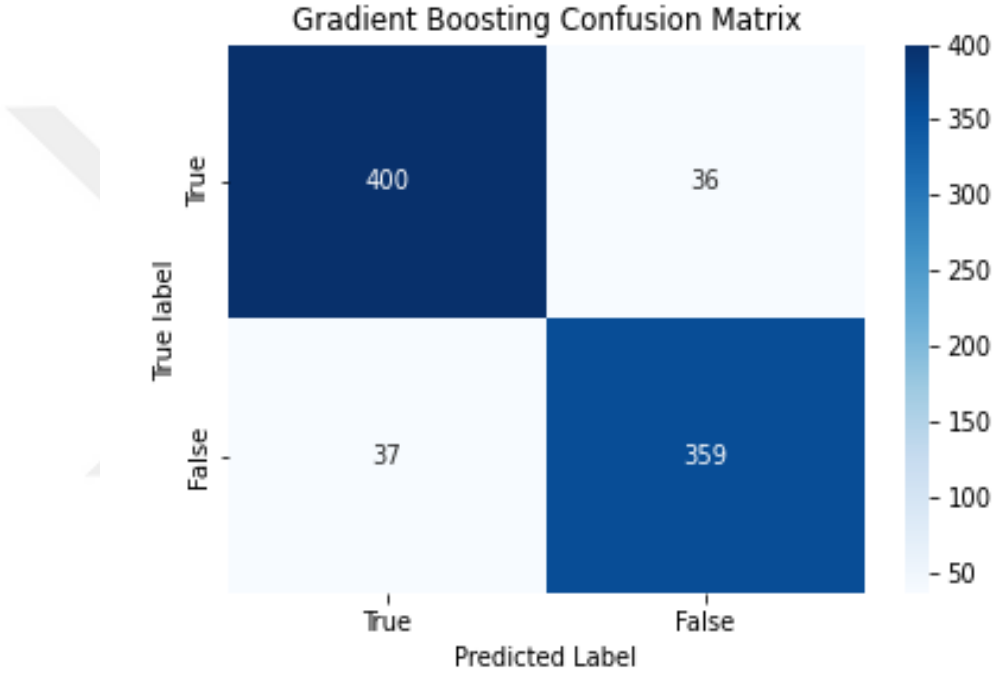
Tablo 7. LBP Yöntemi ile Makine Öğrenmesi Performans Metrikleri

Tablo 7'de LBP yöntemi ile öznitelikleri çıkarılan veri setine adı geçen makine öğrenme algoritmaları uygulandıktan sonra elde edilen performans metrikleri verilmiştir. En yüksek Accuracy oranına 0,8762 ile Random Forest algoritması ulaşmıştır. 0,8618 ile Gradient Boosting algoritması en yüksek ikinci Accuracy oranına ulaşan algoritma olmuştur. En düşük Accuracy oranına sahip algoritma ise 0,5853 ile Support Vector Machine algoritması olmuştur. Precision, Recall ve F1 Score metriklerinde ise sıralama Accuracy oranına paralel olarak sonuçlar vermiştir.

Elde edilen performanslar karşılaştırıldığında VGG16 derin öğrenme modeli ile elde edilen veri seti üzerinde çalışan Gradient Boosting algoritması 0,9123 Accuracy

oranı ile tüm çalışma içerisinde en yüksek başarının elde edildiği algoritma olmuştur. 0,9153 Precision oranı, 0,9174 Recall oranı ve 0,9164 F1 Score oranı ile de diğer performans metriklerinde en yüksek başarının sağlandığı algoritma olmuştur.

Çalışmada karmaşıklık matrisleri de oluşturulmuş ve algoritmaların performansları değerlendirilmiştir. Şekil 9'da Accuracy oranına sahip VGG16 derin öğrenme modeli ile elde edilen veri seti üzerinde çalışan Gradient Boosting algoritmasının karmaşıklık matrisi verilmiştir.



Şekil 13. VGG16 derin öğrenme modeli ile elde edilen veri seti üzerinde çalışan Gradient Boosting algoritmasının karmaşıklık matrisi

Şekil 12'da verilen karmaşıklık matrisi incelendiğinde 400 True – Negative, 36 False – Positive, 359 True – Positive, 37 False – Negative değerlerinin elde edildiği görülmektedir.

Çalışmada makine öğrenme algoritmalarının çalışırken kullandığı kaynaklar ve çalışma süreleri de değerlendirmeye tabi tutulmuştur. Algoritmaları bu açıdan da karşılaştırılmıştır.

Öznitelik Çıkarılan Derin Öğrenme Modeli	VGG16						
Model	**RF**	**SVM**	**NB**	**LR**	**KNN**	**DT**	**GB**
Eğitim Süresi (sn)	10,9119	174,267	1,536	4,6894	0,6596	15,2523	2158,968
Eğitim Bellek Kullanımı (MB)	4,4766	203,918	0	636,6445	636,2383	0,0156	0,1758
Eğitim CPU Kullanımı (%)	7	23,1	3	92,7	38,1	49,2	49,3
Test Süresi (sn)	0,1723	32,1821	0,3759	0,0221	1,1313	0,0312	0,1404
Test Bellek Kullanımı (MB)	0,0547	0,3906	0,2812	0,0195	18,4961	0	0
Test CPU Kullanımı (%)	9,1	2,5	41,7	97,7	619	100,8	55,4

Tablo 8. VGG16 Öznitelikleri ile Makine Öğrenmesi Algoritmaları Kaynak Kullanım Performansları

Tablo 8'de VGG16 derin öğrenme modeli ile öznitelikleri çıkarılan veri setine adı geçen makine öğrenmesi algoritmaları uygulandıktan sonra elde edilen kaynak kullanımları verilmiştir. En düşük öğrenme süresine sahip algoritma 1,536 saniye ile Naive Bayes olmuştur. En yüksek öğrenme süresine sahip algoritma 2158,968 saniye ile Gradient Boosting olmuştur. Öğrenmede en düşük bellek kullanımı 0 MB (Çok düşük kullanım olduğundan 0 görülmektedir.) ile Naive Bayes, en yüksek bellek kullanımı ise 636,6445 MB ile Logistic Regresyon tarafından olmuştur. Öğrenmede en düşük işlemci kullanımı %3 ile Naive Bayes, en yüksek işlemci kullanımı ise %92,7 ile Logistic Regresyon tarafından olmuştur.

Öznitelik Çıkarılan Derin Öğrenme Modeli	VGG19						
Model	**RF**	**SVM**	**NB**	**LR**	**KNN**	**DT**	**GB**
Eğitim Süresi (sn)	13,4234	144,6954	1,3838	4,6214	0,6437	20,776	73,1224
Eğitim Bellek Kullanımı (MB)	4,832	199,6602	0,0117	636,8555	636,2383	0,043	74,132
Eğitim CPU Kullanımı (%)	30,8	19,8	53,9	61,7	4,8	35,4	70,7
Test Süresi (sn)	0,1425	29,6279	0,3487	0,018	1,1273	0,034	0,0425
Test Bellek Kullanımı (MB)	0,25	0,0234	0,2812	0,0195	17,0781	0	0,55
Test CPU Kullanımı (%)	78,1	22	77,2	0	648,6	99,7	128,2

Tablo 9. VGG19 Öznitelikleri ile Makine Öğrenmesi Algoritmaları Kaynak Kullanım Performansları

Tablo 9'da VGG19 derin öğrenme modeli ile öznitelikleri çıkarılan veri setine adı geçen makine öğrenmesi algoritmaları uygulandıktan sonra elde edilen kaynak kullanımları verilmiştir. En düşük öğrenme süresine sahip algoritma 1,3838 saniye ile Naive Bayes olmuştur. En yüksek öğrenme süresine sahip algoritma ise 144,6954 saniye ile Support Vector Machine olmuştur. Öğrenmede en düşük bellek kullanımı 0,043 MB ile Decision Tree, en yüksek bellek kullanımı ise 683,8555 MB ile Logistic Regresyon tarafından olmuştur. Öğrenmede en düşük işlemci kullanımı %4,8 ile KNN, en yüksek işlemci kullanımı ise %61,7 ile Logistic Regresyon tarafından olmuştur.

Öznitelik Çıkarılan Derin Öğrenme Modeli	IncepitonV3						
Model	**RF**	**SVM**	**NB**	**LR**	**KNN**	**DT**	**GB**
Eğitim Süresi (sn)	31,1797	185,9322	3,0063	9,0104	1,2696	61,8818	217,2415
Eğitim Bellek Kullanımı (MB)	6,1367	440,7969	0,3945	1299,668	1298,441	0,0156	110,2347
Eğitim CPU Kullanımı (%)	52,1	30,3	46,9	62,5	13,6	47	54,1
Test Süresi (sn)	0,2681	36,9647	0,7218	0,036	2,1025	0,0685	0,2881
Test Bellek Kullanımı (MB)	0,4375	0,0742	1,1758	0	7,5273	0	0,6125
Test CPU Kullanımı (%)	33,4	12,5	32,6	33,2	657,4	49,6	172,4

Tablo 10. InceptionV3 Öznitelikleri ile Makine Öğrenmesi Algoritmaları Kaynak Kullanım Performansları

Tablo 10'da InceptionV3 derin öğrenme modeli ile öznitelikleri çıkarılan veri setine adı geçen makine öğrenmesi algoritmaları uygulandıktan sonra elde edilen kaynak kullanımları verilmiştir. En düşük öğrenme süresine sahip algoritma 1,2696 saniye ile KNN olmuştur. En yüksek öğrenme süresine sahip algoritma ise 217,2415 saniye ile Gradient Boosting olmuştur. Öğrenmede en düşük bellek kullanımı 0,0156 MB ile Decision Tree, en yüksek bellek kullanımı ise 1299,668 MB ile Logistic Regresyon tarafından olmuştur. Öğrenmede en düşük işlemci kullanımı %13,6 ile KNN, en yüksek işlemci kullanımı ise %62,5 ile Logistic Regresyon tarafından olmuştur.

Öznitelik Çıkarılan Derin Öğrenme Modeli	MobileNet						
Model	**RF**	**SVM**	**NB**	**LR**	**KNN**	**DT**	**GB**
Eğitim Süresi (sn)	11,4207	162,0243	2,9631	8,8754	1,4213	22,0371	2643,491
Eğitim Bellek Kullanımı (MB)	6,5312	377,3516	0,3906	1275,133	1272,473	0,0273	5,3164
Eğitim CPU Kullanımı (%)	57,6	25,2	65,8	96,7	35,2	60,2	67,9
Test Süresi (sn)	0,2946	32,9665	0,6986	0,036	1,9979	0,068	0,4323
Test Bellek Kullanımı (MB)	0,0039	1,1914	1,1523	0,0195	21,3438	0	0
Test CPU Kullanımı (%)	5,3	54,9	91,1	0	691,9	75,6	71,5

Tablo 11. MobileNet Öznitelikleri ile Makine Öğrenmesi Algoritmaları Kaynak Kullanım Performansları

Tablo 11'de MobileNet derin öğrenme modeli ile öznitelikleri çıkarılan veri setine adı geçen makine öğrenmesi algoritmaları uygulandıktan sonra elde edilen kaynak kullanımları verilmiştir. En düşük öğrenme süresine sahip algoritma 1,4213 saniye ile KNN olmuştur. En yüksek öğrenme süresine sahip algoritma ise 2643,491 saniye ile Gradient Boosting olmuştur. Öğrenmede en düşük bellek kullanımı 0,0273 MB ile Decision Tree, en yüksek bellek kullanımı ise 1275,133 MB ile Logistic Regresyon tarafından olmuştur. Öğrenmede en düşük işlemci kullanımı %25,2 ile Support Vector Machine, en yüksek işlemci kullanımı ise %96,7 ile Logistic Regresyon tarafından olmuştur.

Öznitelik Çıkarılan Derin Öğrenme Modeli	DenseNet121						
Model	**RF**	**SVM**	**NB**	**LR**	**KNN**	**DT**	**GB**
Eğitim Süresi (sn)	755,969	1780,474	34,703	43,070	14,992	203,282	234,337
Eğitim Bellek Kullanımı (MB)	5,543	3568,281	0,3828	12,741	12,724	0,039	101,839
Eğitim CPU Kullanımı (%)	47,7	45,296	45,347	60,5	50,0	45,6	36,7
Test Süresi (sn)	0,2331	315,585	0,7718	0,0708	19,327	0,071	0,6013
Test Bellek Kullanımı (MB)	0,0391	0,6836	0,8984	0,6289	394,844	0,5212	319,457
Test CPU Kullanımı (%)	66,8	45,478	48,0	12,40	53,05	40,10	38,50

Tablo 12. DenseNet121 Öznitelikleri ile Makine Öğrenmesi Kaynak Kullanım Performansları

Tablo 12'de DenseNet121 derin öğrenme modeli ile öznitelikleri çıkarılan veri setine adı geçen makine öğrenmesi algoritmaları uygulandıktan sonra elde edilen kaynak kullanımları verilmiştir. En düşük öğrenme süresine sahip algoritmalar 14,992 saniye ile KNN olmuştur. En yüksek öğrenme süresine sahip algoritma ise 1780,474 saniye ile Support Vector Machine olmuştur. Öğrenmede en düşük bellek kullanımı 0,3828 MB ile Naive Bayes, en yüksek bellek kullanımı ise 3568,281 MB ile Support Vector Machine tarafından olmuştur. Öğrenmede en düşük işlemci kullanımı %36,7 ile Gradient Boosting; en yüksek işlemci kullanımı ise %60,05 ile Logistic Regresyon tarafından olmuştur.

Öznitelik Çıkarılan Derin Öğrenme Modeli	ResNet50						
Model	**RF**	**SVM**	**NB**	**LR**	**KNN**	**DT**	**GB**
Eğitim Süresi (sn)	177,801	2659,644	82,137	261,842	39,925	41,677	4530,091
Eğitim Bellek Kullanımı (MB)	69,375	843,357	30,742	2547,043	2544,945	0,78	93,017
Eğitim CPU Kullanımı (%)	45,433	45,341	45,502	52,68	45,556	34,5	45,535
Test Süresi (sn)	0,4032	536,107	17,295	0,1243	49,673	0,2573	15,143
Test Bellek Kullanımı (MB)	0,7344	12,03	0,039	0,0234	42,652	0,280	0,0312
Test CPU Kullanımı (%)	34,6	45,547	45,562	57,3	51,39	25,0	41,20

Tablo 13. ResNet50 Öznitelikleri ile Makine Öğrenmesi Algoritmaları Kaynak Kullanım Performansları

Tablo 13'te ResNet50 derin öğrenme modeli ile öznitelikleri çıkarılan veri setine adı geçen makine öğrenmesi algoritmaları uygulandıktan sonra elde edilen kaynak kullanımları verilmiştir. En düşük öğrenme süresine sahip algoritma 39,921 saniye ile KNN olmuştur. En yüksek öğrenme süresine sahip algoritma ise 4530,091 saniye ile Gradient Boosting olmuştur. Öğrenmede en düşük bellek kullanımı 0,78 MB ile Decision Tree, en yüksek bellek kullanımı ise 2547,043 MB ile Logistic Regresyon tarafından olmuştur. Öğrenmede en düşük işlemci kullanımı %34,5 ile Decision Tree tarafından olmuştur. En yüksek işlemci kullanımı ise %52,26 ile Logistic Regresyon tarafından olmuştur.

Öznitelik Çıkarılan Yöntem				**LBP**			
Model	**RF**	**SVM**	**NB**	**LR**	**KNN**	**DT**	**GB**
Eğitim Süresi (sn)	1,76	0,18	0,01	0,03	0,01	0,6	7,27
Eğitim Bellek Kullanımı (MB)	5,281	0,7343	0,2	0,234	0,128	0,2	1,984
Eğitim CPU Kullanımı (%)	21,2	0,5	0,6	50,4	0,4	54,4	60,4
Test Süresi (sn)	0,01	0,03	0,02	0,01	0,01	0,001	0,002
Test Bellek Kullanımı (MB)	0,01	0,023	0,01	0,03	0,039	0,01	0,0234
Test CPU Kullanımı (%)	0,01	0,01	0,01	0,01	10,4	0,01	97,7

Tablo 14. LBP Öznitelikleri ile Makine Öğrenmesi Algoritmaları Kaynak Kullanım Performansları

Tablo 14'te LBP yöntemi ile öznitelikleri çıkarılan veri setine adı geçen makine öğrenmesi algoritmaları uygulandıktan sonra elde edilen kaynak kullanımları verilmiştir. En düşük öğrenme süresine sahip algoritma 0,1 saniye ile KNN ve Naive Bayes olmuştur. En yüksek öğrenme süresine sahip algoritma ise 7,27 saniye ile Gradient Boosting olmuştur. Öğrenmede en düşük bellek kullanımı 0,2 MB ile Decision Tree ve Naive Bayes, en yüksek bellek kullanımı ise 5,281 MB ile Random Forest tarafından olmuştur. Öğrenmede en düşük işlemci kullanımı %0,4 ile KNN tarafından olmuştur. En yüksek işlemci kullanımı ise %60 ile Gradient Boosting tarafından olmuştur.

TARTIŞMA

GKD’nin doğru teşhisinde, doğru görüntünün alınması en önemli ve ilk aşamadır. Bu çalışma, GKD teşhisinde doğru görüntünün tespiti için makine öğrenmesi algoritmalarının başarıyla kullanılabileceğini göstermektedir. Ancak bu bulguların uygulanabilirliği ve genel geçerliliği üzerine bazı tartışmalar bulunmaktadır.

Çalışmada kullanılan özgün veri setinin boyutu ve çeşitliliği, sonuçların genelleştirilebilirliğini etkileyebilir. Veri setinin boyutunun sınırlı olması, makine öğrenmesi algoritmalarının genel performansını etkileyebilir ve sonuçların güvenilirliğini sınırlayabilir. Gelecekteki araştırmalar için daha büyük ve çeşitli veri setleri kullanılarak başarı ve güvenirlilik artırılabilir.

Çalışmanın sonuçları, GKD teşhisi için makine öğrenmesi tekniklerinin klinik uygulamalarda kullanılmasını desteklemektedir. Ancak bu tekniklerin gerçek dünya uygulamalarına entegrasyonu ve doktorlar tarafından günlük pratikte kullanılabilirliği hakkında daha fazla araştırmaya ihtiyaç vardır. Ayrıca bu teknolojilerin maliyeti ve hastaların bu teknolojilere erişimine etkisi gibi pratik faktörler de dikkate alınmalıdır.

GKD teşhisinde klasik yöntemler, yıllardır kullanılan ancak bazı sınırlamalara sahip olan görüntüleme ve fiziki muayene gibi yöntemleri içerir. Bu yöntemler, genellikle doktorların deneyim ve görsel değerlendirme yeteneklerine dayanır ve doğru teşhis için öznel değerlendirmelere bağlıdır. Ancak bu yaklaşımın doğruluğu ve güvenilirliği zaman zaman sınırlı olabilir, bu da teşhisin gecikmesine veya hatalı teşhise yol açabilir. Bu çalışma ile GKD teşhisinde klasik yöntemlerle elde edilen görüntülerin doğruluğunu artırmak için makine öğrenmesi algoritmalarının potansiyeli incelenmiştir. Makine öğrenmesi algoritmalarının kullanılmasıyla doğru görüntülerin tespit edilmesi ve GKD'nin erken teşhisi için daha objektif ve hassas bir yaklaşım sunulmuştur.

Sonuç olarak bu çalışma, GKD tanısı için ultrason görüntülerinin değerlendirilmesinde makine öğrenmesi algoritmalarının başarıyla kullanıldığını göstermiştir. Bu, doktorların doğru görüntüleri tespit etmelerine yardımcı olabilir ve erken teşhiste başarı oranını artırabilir. Gelecekteki çalışmalarda, görüntüler farklı görüntü işleme teknikleri ile işlenerek ve daha büyük veri setleri ile farklı algoritmalar kullanılarak bu sistemin performansı daha da iyileştirebilir. Ayrıca ultrason cihazlarına entegre edilerek GKD tanısında yapılabilecek yanlış değerlendirmelerin önüne geçilebilir.

SONUÇ

GKD teşhisinde ilk adım olan doğru görüntünün tespit edilmesi en önemli adımdır. Bu adımın yanlış atılması yanlış teşhise ya da teşhisin gecikmesine neden olacaktır. Teşhisin gecikmesi ise acılı, zor ve maliyetli tedavileri gerektirmektedir. Ayrıca teşhisin geç yapılması hastalarda kalıcı ortopedik rahatsızlıklara yol açabilmektedir. Bu açıdan bakıldığında doğru görüntünün teşhis edilmesi ilk ve en önemli adımdır.

Çalışmada ortaya konulan makine öğrenmesi yöntemleri ile amaç, GKD teşhisinde doğru görüntünün erken ve yüksek başarı ile tespit edilmesin için doktorlara yardımcı olmaktır. VGG16 derin öğrenme modeli ile ultrason kayıtlarından elde edilen görüntülerin özelliklerinin çıkarıldığı veri seti üzerinde çalışan Gradient Boosting makine öğrenmesi algoritması %91,23 başarıya ulaşmıştır. Bu oran GKD'nin erken teşhisinde oldukça önemli bir başarıdır.

Çalışmada 6 farklı önceden eğitilmiş derin öğrenme modeli ve LBP öznitelik çıkarma yöntemi ile görüntülerin özellikleri çıkarılmış ve veri seti hazırlanmıştır. En başarılı sonuç VGG16 modeli ile elde edilen veri setinde yakalanmıştır. Bu veri seti 4156x25089 boyutundadır. Hazırlanan her veri seti üzerinde yedi makine öğrenme algoritması çalıştırılmıştır. Her algoritmanın sonuçları birbiri ile karşılaştırılmıştır. En başarılı sonuç Gradient Boosting algoritması ile elde edilmiştir.

Çalışmada önceden eğitilmiş derin öğrenme modellerinin özellik çıkarma performansı, LBP özellik çıkarma yöntemi ile karşılaştırılmıştır. LBP yöntemi ile elde edilen veri seti üzerinde çalışmada kullanılan makine öğrenmesi algoritmaları uygulanmış ve en yüksek Accuracy oranı Random Forest ile %87,62 elde edilmiştir. Derin öğrenme modellerinden VGG16 ve Gradient Boosting makine öğrenmesi kombinasyonunun %91,23 başarı oranının altında kaldığı görülmüştür.

Çalışma için özel hazırlanan özgün bir veri seti kullanılmıştır. Çalışma, bu veri setinin makine öğrenmesi algoritmaları ile doğru ve yanlış şeklinde sınıflandırılmasının yapıldığı ilk çalışma olması açısından önemlidir. Veri seti oluşturulmak için kullanılan ultrason kayıtlarının sayısının az olması bu çalışmanın dezavantajı olarak görülebilir.

Sonuç olarak bu çalışma GKD teşhisinde doğru görüntünün tespit edilmesi amacıyla makine öğrenmesi algoritmalarının başarıyla kullanıldığını göstermiştir. Çalışma doktorların doğru görüntünün tespitinde yardımcı olabilir ve GKD'nin erken teşhisinde Accuracy oranını artırabilir. Gelecekte yapılacak çalışmalarda farklı

sınıflandırma yöntemleri kullanılarak Accuracy oranı artırılabilir. Ayrıca daha büyük veri setleri oluşturularak da sistemin performansı yükseltilebilir. Ultrason makinelerine bu sistem entegre edilerek dünya genelinde GKD'nin erken teşhisindeki başarı artırılabileceği gibi yanlış değerlendirmelerin de önüne geçilebilir.

KAYNAKÇA

A. R. Hareendranantha, M. M. (2022). Can AI Automatically Assess Scan Quality of Hip Ultrasound? *Applied Sciences, 42*(8), 4072.

Abdulselam BATU, C. C. (2018). Gelişimsel Kalça Displazisi Taramasında Ultrasonagrafinin Yeri. *Medical Research Reports 1(2)*, s. 36-39.

Abhilash Rakkunedeth Hareendranathan, A. T. (2022). Domain-aware contrastive learning for ultrasound hip image analysis. *Computers in Biology and Medicine*, s. 106004.

Ahmet Sinan Sarı, O. K. (2020). Is experience alone sufficient to diagnose developmental dysplasia of the hip without the bony roof (alpha angle) and the cartilage roof (beta angle) measurements? *Medicine (Baltimore)*, s. 19677.

Ahuja, S. A. (2017). Machine learning and its applications: A review. *International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*, (s. 57-60). Chirala, Andhra Pradesh, India.

Alex Krizhevsky, I. S. (2017). ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM 60(6)*, s. 84-90.

Alexander Kolb, E. B. (2017). Measurement Considerations on Examiner-dependent Factors in the Ultrasound Assessment of Developmental Dysplasia of the Hip. *International Orthopaedics 41(6)*, s. 1-6.

Alpaydın, E. (2011). *Introduction to Machine Learning Second Edition.* The MIT Press.

Anand Pillai, J. J. (2011). Diagnostic accuracy of static graf technique of ultrasound evaluation of infant hips for developmental dysplasia. *Arch Orthop Trauma Surg*, s. 53-58.

Andrew G. Howard, M. Z. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *Computer Vision and Pattern Recognition*, s. 2-6.

Animesh Urgiriye, R. B. (2020). Review of Machine Learning Algorithm on Cancer Data Set. *International Journal of Scientific Research & Engineering Trends 6(6)*, s. 3259-3267.

Anthony J. Myles, R. N. (2004). An Introduction to Decision Tree Modeling. *Journal of Chemometrics*, s. 275-285.

Atik, Ş. Ö. (2022). Derin Öğrenme Yaklaşımlarıyla Çevresel İzlemeye Yönelik Çok Sınıflı Sahne Sınıflandırma . *Avrupa Bilim ve Teknoloji Dergisi,41*, s. 307-314.

Babar, V. S. (2021). A Review of Machine Learning and Its Applications. *International Journal of Engineering Applied Sciences and Technology*.

Bahar Kural, E. D. (2019, Aug). Risk Factor Assessment and a Ten-Year Experience of DDH Screening in a Well-Child Population. *National Library of Medicine*.

Bahbibi Rahmatullah, A. P. (2011). Automated Selection of Standardized Planes from Ultrasound Volume. K. W. Suzuki içinde, *Machine Learning in Medical Imaging* (s. 35-42). Berlin, Heidelberg: Springer Berlin Heidelberg.

Bas H.M. van der Velden, H. J. (2022). Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. *Medical Image Analysis, 79*, 102470.

Baumgartner, C. F. (2016). Real-Time Standard Scan Plane Detection and Localisation in Fetal Ultrasound Using Fully Convolutional Neural Networks. S. J. Ourselin içinde, *Medical Image Computing and Computer-Assisted Intervention* (s. 203--211). MICCAI: Springer International Publishing.

Bingxuan Huang, B. X. (2023). Artificial Intelligence-Assisted Ultrasound Diagnosis on Infant Developmental Dysplasia of the Hip Under Constrained Computational Resources. *Journal of Ultrasound in Medicine, 42*(6), 1235-1248.

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, s. 123-140.

Breiman, L. (2001). Random Forests. *Machine Learning 45*, s. 5 -32.

Carl Kingsford, S. L. (2008). What Are Decision Trees? *Nature Biotechnology, 26*, s. 1011-1013.

Carol Dezatuex, K. R. (2007). Developmental Dysplasia oft the Hip. *The Lancet 369*, s. 1541 - 1552.

Chen, T. H. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1 - 4.

Choi S, J. Z. (2010, January). Cardiac sound murmurs classification with autoregressive spectral analysis and multi-support vector machine technique. *Comput Biol Med*, s. 8 - 20.

Christian SZEGEDY, V. V. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, s. 2818-2826.

Christian Szegedy, V. V. (2016). Rethinking the Inception Architecture for Computer Vision, Zbigniew Wojna. *Conference on Computer Vision and Pattern Recognition (CVPR)* (s. 2818-2826). Las Vegas: IEEE.

Civelek, S. (2023). Makine Öğrenmesi Yöntemleri ile Öğrenci Mezuniyet Notu Tahmini. *Yüksek Lisans Tezi*. Bolu Abant İzzet Baysal Üniversitesi Lisanüstü Eğitim Enstitüsü.

Cristianini, N. (2013). *Support Vector Machines.* London: Cambridge University Press.

Çam, K. (2021). Küçük ve orta ölçekli firmalar için makine öğrenmesi destekli karar destek sistemi. *Yüksek Lisans Tezi*. Dokuz Eylül Üniversitesi, Sosyal Bilimler Enstitüsü.

Çokluk, Ö. (2010). Lojistik Regresyon Analizi: Kavram ve Uygulama. *Kuram ve Uygulamada Eğitim Bilimleri*, s. 1357 - 1407.

Dangeti, P. (2017). *Statistics for Machine Learning: Techniques for exploring supervised, unsupervised, and reinforcement learning models with Python and R.* Packt Publishing.

Derin Öğrenme (Deep Learning) Nedir? (2022). https://www.beyaz.net/tr/: https://www.beyaz.net/tr/yazilim/makaleler/derin_ogrenme_deep_learning_nedir.html (Erişim Tarihi: 11.10.2023) adresinden alındı

E A Simon, F. S. (2004). Inter-observer agreement of ultrasonographic measurement of alpha and beta angles and the final type classification based on the Graf method. *Swiss Med Wkly. 134(45-46)*, s. 671-677.

Erdal Taşçı, A. O. (2017). K-En Yakın Komşu Algoritması Parametrelerinin Sınıflandırma Performansı Üzerine Etkisinin İncelenmesi. *Akademik Bilişim 1(1)*, s. 4-18.

Erol Egrioglu, C. H. (2009). A new approach based on artificial neural networks for high order multivariate fuzzy time series. *Expert Systems with Applications*, s. 10589 - 10594.

Erwan Scornet, G. B.-P. (2015). Consistency of Random Forests. *The Annals of Statistics*, s. 1716-1741.

Ezel, E. (2018). Derin Öğrenme Yöntemi Kullanılarak Görüntü tabanlı Türk İşaret Dili tanıma. *Yüksek Lisans Başlığı*. Konya: Selçuk Üniversitesi Fen Bilimleri Enstitüsü.

Freund, Y. (1995). Boosting a Weak Learning Algorithm by Majority. *Information and Computation*, s. 256-285.

Furnes O., L. S. (2001). Hip disease and the prognosis of total hip replacements. A review of 53,698 primary total hip replacements reported to the Norwegian Arthroplasty Register 1987-99. *Bone Joint Surg Br*.

Gamze Kaba, S. B. (2022). Kardiyovasküler Hastalık Tahmininde Makine Öğrenmesi Sınıflandırma Algoritmalarının Karşılaştırılması. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, s. 183-193.

Gao Huang, Z. L. (2016). Densely Connected Convolutional Networks. *Computer Vision and Pattern Recognition*.

Gareth James, D. W. (2013). *An Introduction to Statistical Learning with Applications in R.* Springer.

Gizem Dilki, Ö. D. (2020). İşletmelerin İflas Tahmininde K-En Yakın Komşu Algoritması Üzerinden Uzaklık Ölçütlerinin Karşılaştırılması. *Fen Bilimleri Dergisi*, s. 224 - 233.

Goh, A. T. (1995). Back-propagation neural networks for modeling complex systems. *Artificial Intelligence in Engineering*, s. 143-151.

Görmez, B. (2021). Adli Bilişimde Makine Öğrenmesi: Makine Öğrenmesi Algoritmaları ile Terör Olaylarının Tahmin Edilmesi Çalışması. *Yüksek Lisans Tezi*. Ankara Üniversitesi, Sağlık Bilimleri Enstitüsü.

Graf. (1984). Classification of hip joint dysplasia by means of sonography. *Arch Orthop Trauma Surg*, s. 117-133.

Graf, R. (1984). Fundamentals of Sonographic Diagnosis of Infant Hip Dysplasia. *Journal of Pediatric Orthopaedics*, s. 735-740.

Graf, R. M. (2013). Hip sonography update. Quality-management, Catastrophes Tips and Tricks. *Medical ultrasonography 15(4)*, s. 299-303.

Guanfang Dong, Y. M. (2021). Feature-Guided CNN for Denoising Images From Portable Ultrasound Devices. *IEEE Access, 9*, 28272-28281.

Gülşen, P. (2023). Makine Öğrenmesi Algoritmaları Kullanılarak Disfonik Seslerin İncelenmesi. *Yüksek Lisans Tezi*. Erciyes Üniversitesi Fen Bilimleri Enstitüsü.

H. Atalar, U. S. (2007). Indicators of Successful Use of the Pavlik Harness in iIfants with Developmental Dysplasia of the Hip. *International Prthopeaedics 31(2)*, s. 145-150.

H. Ömeroğlu, A. B. (2001). Assessment of variations in the measurement of hip ultrasonography by the Graf method in developmental dysplasia of the hip. *J. Pediatr Orthop B.*, s. 89-95.

Hao Chen, L. W.-Z.-A. (2017). Ultrasound Standard Plane Detection Using a Composite Neural Network Framework. *IEEE Transactions on Cybernetics*, s. 1576-1586.

Harrington, P. (2012). *Machine learning in action.* Simon and Schuster.

Hasan Basri Sezer, A. S. (2019). Automatic Segmentation and Classification of Neonatal Hips According to Graf's Sonographic Method: A Computer-Aided Diagnosis System. *Applied Soft Computing vol. 82*.

Hastie, T. R. (2009). An introduction to statistical learning.

Haydar Hoşgör, H. G. (2022). Sağlıkta Yapay Zekanın Kullanım Alanları Üzerine Nitel Bir Araştırma. *Avrupa Bilim ve Teknoloji Dergisi (35)*, s. 395-407.

Hilmi Cenk Bayrakçı, R. S. (2021, 12 31). Tarım Arazilerinde Harcanan Su Miktarını Yapay Zekâ Teknikleri Kullanarak Belirlenmesi. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, s. 237 - 250.

Ho, T. K. (1998). The Random Subspace Method for Constructing Decision Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, s. 832-844.

Ian Goodfellow, Y. B. (2016). *Deep Learning.* MIT Press.

Ibrahim Mahmood, A. M. (2021). The Role of Machine Learning Algorithms for Diagnosing Diseases. *Journal of Applied Science and Technology Trends*, s. 10-19.

Ihsan Ullah, M. H.-u.-H. (2018). An Automated System for Epilepsy Detection Using EEG Brain Signals Based on Deep Learning Approach. *Expert Systems with Applications 107(1)*, s. 61-71.

Jacob L Jaremko, M. M. (2014). Potential for change in US diagnosis of hip dysplasia solely caused by changes in probe orientation: patterns of alpha-angle variation revealed by using three-dimensional US. *Radiology*, s. 870-878.

Jakup Nalepa, M. M. (December 2019). Data Agumentation for Brain Tumor Segmentation: A rewiev. *Frontiers in Computational Neuroscience, 13*(83), 1-18.

Jiawei Han, M. K. (2022). *Data mining: concepts and techniques.* Morgan Kaufmann.

Jingnan He, T. C. (2023). Analysis of the results of hip ultrasonography in 48 666 infants and efficacy studies of conservative treatment. *Kournal of Clinical Ultrasound, 51*(4), 656-662.

Kadir ÖZTÜRK, M. E. (2018). Yapay Sinir Ağları ve Yapay Zekâ'ya Genel Bir Bakış. *Takvim-i Vekayi*, s. 25-36.

Kalaycı, T. E. (2018). Kimlik hırsızı Web Sitelerinin sınıflandırılması için Makine öğrenmesi yöntemlerinin karşılaştırılması. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, s. 870-878.

Karaköse, M. (2023). Teknik Göstergeleri Kullanarak İstanbul Şehir Endeksi Hareketinin Tahmini: Makine Öğrenmesine Dayalı Bir Yaklaşım. *Yüksek Lisans tezi*. Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü.

Kasthurirangan Gopalakrishnan, S. K. (2017). Deep Convolutional Neural Networks with Transfer Learning for Computer Vision-based Data-driven Pavement Distress Detection. *Construction and Building Materials 157(30)*, s. 322-330.

Kavuncu, S. K. (2018). Makine Öğrenmesi ve Derin Öğrenme: Nesne Tanıma Uygulaması. *Yüksek Lisans tezi*. Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü.

Kaya, Ö. Ş. (2023). Makine Öğrenmesi Algoritmaları ile Yazılım Hata Kestirimi. *Yüksek Lisans tezi*. Eskişehir: Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü.

Kayalı, S. (2023). Makine Öğrenmesi Yöntemleriyle Öğrencilerin Akademik Performanslarının Farklı Öznitelik Seçim Teknikleri uygulanarak Sınıflandırılması. *Yüksek Lisans Tezi*. Çankırı Karatekin Üniversitesi Fen Bilimleri Enstitüsü.

Kuru, E. (2023). Makine Öğrenmesi Algoritmalarıyla Gıda Sektöründe Karar Destek Sistemlerinin Oluşturulması. *Yüksek Lisans tezi*. Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü.

Ladha L., D. T. (2011). Feature Selection Methods And Algorithms. *International Journal on Computer Science and Engineering, 3*(5), 1787-1797.

Lamya Ann Athew, J. H. (2013). Multimodality Imaging of Developmental Dysplasia of the Hip. *PediatricnRadiology*, 166-171.

Leung, K. M. (2007). Naive Bayesian Classifier. *Polytechnic University*, s. 123-156.

Li, K. Z. (2013). A SVM Based Classification of EEG for Predicting the Movement Intent of Human Body. *2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, s. 402-406.

Logistic Regression: A Binary Classifier. (2023). https://rasbt.github.io/: https://rasbt.github.io/mlxtend/user_guide/classifier/LogisticRegression/ (Erişim Tarihi: 11.10.2023) adresinden alındı

Luiz Pereira, S. B. (2018). Predicting the Ripening of Papaya Fruit with Digital Imaging and Random Forests. *Computers and Electronics in Agriculture*, s. 76-82.

M. Erdal Balaban, E. K. (2018). *Veri Madenciliği ve Makine Öğrenmesi.* Çağlayan Kitabevi.

Mahesh, B. (2018). Machine Learning Algorithms -A Review. *International Journal of Science and Research*, s. 381-386.

Maki Kinugasa, A. I. (2023). Diagnosis of Developmental Dysplasia of the Hip by Ultrasound Imaging Using Deep Learning. *Journal of Pediatric Orthopaedics, 43*(7), 538-544.

Manali Shaha, M. P. (2018). Transfer Learning for Image Classification. *2018 Second International Conference on Electronics* (s. 656-660). India: Communication and Aerospace Technology (ICECA), Coimbatore.

Manoj Mannil, J. v. (2018). Texture Analysis and Machine Learning for Detecting Myocardial Infarction in Noncontrast Low-Dose Computed Tomography: Unveiling the Invisible. *Investigative Radiology, 53*(6), 338-343.

Mariana S. Silva, A. R. (2019). Radiography, CT, and MRI of Hip and Lower Limb Disorders in Children and Adolescents. *National Library of Medicine 39(3)*, s. 779-794.

Mark Cicero, A. B. (2017). Training and Validating a Deep Convolutional Neural Network for Computer-Aided Detection and Classification of Abnormalities on Frontal Chest Radiographs. *Investigative Radiology, 52*(5), 281-287.

Maulud Dastan, A. M. (2020). A Review on Linear Regression Comprehensive in Machine Learning. *Journal of Applied Science and Technology Trends 1(4)*, s. 140-147.

Meryem Pulat, D. B. (2023). Kayıt Sistemi Süreç İyileştirmesinde Simülasyon Tekniğinin Kullanımı ve Bir Uygulaması. *The Black Sea Journal of Sciences 13(1)*, s. 42-59.

Mingyuan Luo, X. Y. (2021). Self Context and Shape Prior for Sensorless Freehand 3D Ultrasound Reconstruction. *Computer Vision and Pattern Recognition*, s. 201-210.

Mitchell, T. (1997). *Machine Learning.* McGraw Hill.

MobileNet. (2023, 30 01). https://www.codingninjas.com/: https://www.codingninjas.com/studio/library/mobilenet (Erişim Tarihi: 10.10.2023) adresinden alındı

Mustafa Berk Keleş, A. K. (2020). Yapay Zekâ Teknolojisi ile Uçuş Fiyatı Tahmin Modeli Geliştirme. *Turkish Studies - Information Technologies and Applied Sciences*, s. 511-520.

Na, J. W. (2021). An Extended K Nearest Neighbors-Based Classifier for Epilepsy Diagnosis. *IEEE Access*, s. 73910-73923.

Nedir Bu Destek Vektör Makineleri? (Makine Öğrenmesi Serisi-2). (2020, 08 25). https://medium.com/: https://medium.com/deep-learning-turkiye/nedir-bu-destek-vekt%C3%B6r-makineleri-makine-%C3%B6%C4%9Frenmesi-serisi-2-94e576e4223e (Erişim Tarihi: 11.10.2023) adresinden alındı

Nerys F. Woolacott, M. A. (2005). Ultrasonography in Screening for Developmental Dysplasia of the Hip in Newborns. *BMJ. 330(7505)*, s. 1413.

Niamul Quader, E. S. (2018). A Systematic Review and Meta-analysis on the Reproducibility of Ultrasound-based Metrics for Assessing Developmental Dysplasia of the Hip. *Journal of Pediatric Orthopaedics*, s. 305-311.

Noble, M. M. (2017). A framework for analysis of linear ultrasound videos to detect fetal presentation and heartbeat. *Medical Image Analysis*, s. 22-36.

Osisanwo, F. A. (2017). Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology (IJCTT)*, s. 128-138.

Önder Çoşkun, S. K. (2004). Lojistik Regresyon Analizinin İncelenmesi ve Dişhekimliğinde Bir Uygulaması. *Cumhuriyet Üniversitesi Diş Hekimliği Fakültesi Dergisi*.

Özdamar, D. Y. (2004). Analitik Hiyerarşi Süreci Yöntemi: Bir Satın Alma İhalesinde Uygulanması. *Yüksek Lisans Tezi*. Ankara Üniversitesi Sosyal Bilimler Enstitüsü.

Paluszek Michael, T. S. (2017). An Overview of Machine Learning. In: MATLAB Machine Learning. *MATLAB Machine Learning* (s. 3-15). içinde Berkeley, CA: Apress.

Paserin, O. M. (2018). Real Time RNN Based 3D Ultrasound Scan Adequacy for Developmental Dysplasia of the Hip. *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2018* (s. 365-373). içinde Springer International Publishing.

Quinlan, J. (1986). Induction of Decision Trees. *Machine learning*, s. 81-106.

Randall T. Loder, E. N. (2011). The epidemiology and demographics of hip dysplasia. *International Scholarly Research Notices*, s. 238607.

Raphael Prevost, M. S. (2018). 3D Freehand Ultrasound without External Tracking Using Deep Learning. *Medical Image Analysis*, s. 187-202.

Recep Sinan Arslan, H. U. (2023). Sensitive deep learning application on sleep stage scoring by using all PSG data. *Neural Computing & Application, 35*(10), 7495-7508.

Rizvanche, S. (2020). Talep Tahmininde Makine Öğrenmesi ve Zaman Serileri Temelli Bir Karar Destek Sistemi. *Yüksek Lisans Tezi*. Eskişehir Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü.

Ruhan Liu, M. L. (2021). NHBS-Net: A Feature Fusion Attention Network for Ultrasound Neonatal Hip Bone Segmentation. *IEEE Transactions on Medical Imaging, 40*(12), 3446-3458.

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Networks 61*, s. 85-117.

Scott Yang, N. Z. (2019). Developmental Dysplasia of the Hip. *Pediatrics 143(1)*.

Sebastian Ruder, B. P. (2018). Strong Baselines for Neural Semi-supervised Learning under Domain Shift. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (s. 1044-1054). Melbourne, Australia: Association for Computational Linguistics.

Seda Sahin, E. A. (2017). A Novel Computer-Based Method for Measuring the Acetabular Angle on Hip Radiographs. *National Library of Medicine 51(2)*, s. 155-159.

Serpil Gümüştekin Aydın, G. A. (2022). Makine Öğrenmesi Algoritmaları Kullanılarak Türkiye ve AB Ülkelerinin CO2 Emisyonlarının Tahmini. *Avrupa Bilim ve Teknoloji Dergisi 37(22)*, s. 42-46.

Sharmila Ashok, G. P. (2016). DWT Based Detection of Epileptic Seizure From EEG Signals Using Naive Bayes and k-NN Classifiers. *IEEE Access*, s. 7716-7727.

Shaw, B. A. (2016). Evaluation and Referral for Developmental Dysplasia of the Hip in Infants. *Pediatrics 138(6)*.

Shichao ZHANG, X. L. (2017). Learning k for kNN Classification. *ACM Transactions on Intelligent Systems and Technology 8,(3)*, s. 1-19.

Si Cheng Zhang, H. L. (2023). Clinical study on ultrasonic artificial intelligence-assisted diagnosis of developmental hip dysplasia in children. *Journal of Ultrasound in Medicine, 42*(6), 1235-1248.

Si Wook Lee, H. U. (2021). Accuracy of New Deep Learning Model-Based Segmentation and Key-Point Multi-Detection Method for Ultrasonographic Developmental Dysplasia of the Hip (DDH) Screening. *Diagnostics (Basel), 11*(7), 1174.

Si-Cheng Zhang, J. S.-B.-H.-T. (2020). Clinical application of artificial intelligence-assisted diagnosis using anteroposterior pelvic radiographs in children with developmental dysplasia of the hip. *Bone Joint J, 102*(11), 1574-1581.

Siddharth Sharma, S. S. (2020). Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*, s. 310-316.

Sie Heng Sharon Tan, K. L. (2019). The Earliest Timing of Ultrasound in Screening for Developmental Dysplasia of the Hips. *Ultrasonography 38(4)*, s. 321-326.

Siegel, M. J. (2011). *Pediatric Sonography :Fourth Edition.* Institute of Clinical and Translational Sciences.

Silahtaroğlu, G. (2018). *Veri Madenciliği.* İstanbul: Papatya Yayınları.

Silvestrov, S. ,. (2016). *Engineering Mathematics II: Algebraic, Stochastic and Analysis Structures For Networks, Data Classification and Optimization.* Springer.

Simonyan, K. A. (2014). Very Deep Convolutional Networks for Large-scale Image Recognition. *Computer Vision and Pattern Recognition*.

Sisodia, D. S. (2014). Fast and Accurate Face Recognition Using SVM and DCT. *II. International Conference an Soft Computing for Problem Solving.*

Soni Singh, K. R. (2021). Machine Learning Techniques and Implementation of Different ML Algorithms. *2021 2nd Global Conference for Advancement in Technology (GCAT)*, 1-6.

Suat TORAMAN, B. D. (2020). Evrişimsel Sinir Ağları Kullanılarak Normal ve Göğüs Kanseri Hücreleri İçeren Genomların Sınıflandırılması. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi 11(1)*, s. 81-90.

Suat Toraman, İ. T. (2018). Kolon Kanseri Hastaları ve Sağlıklı Kişilerin FTIR Spektrogram Görüntülerinin Derin Öğrenme ile Sınıflandırılması. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi 3*.

Suat TOROMAN, B. D. (2020). Evrişimsel Sinir Ağları Kullanılarak Normal ve Göğüs Kanseri Hücreleri İçeren Genomların Sınıflandırılması. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi,11(1)*, s. 81-90.

Sun Choi, Y. J. (2021). Artificial neural network models for airport capacity prediction. *Journal of Air Transport Management, 97*, s. 102146.

Susan T. Mahan, J. N.-J. (2009). To Screen or Not to Screen? A Decision Analysis of the Utility of Screening for Developmental Dysplasia of the Hip. *J. Bone Joint Surg Am.*, s. 1705-1719.

Suzuki, K. (2013). Suzuki, Kenji, ed. Artificial neural networks: Architectures and applications. *BoD–Books on Demand*.

Şahinarslan, F. V. (2019). Makine Öğrenmesi Algoritmaları ile Nüfus Tahmini: Türkiye Örneği. *Yüksek Lisans Tezi*. İstanbul: İstanbul Teknik Üniversitesi.

Şenel, F. A. (2020). Makine Öğrenmesi Algoritmaları Kullanılarak Kayısı İç Çekirdeklerinin Sınıflandırılması. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi 9(2)*, s. 807-815.

Şengül, Z. (2022). Makine Öğrenmesi Algoritmalarını Kullanarak Bitcoin Fiyat Tahmini. *Yüksek Lisans Tezi*. Edirne: Trakya Üniversitesi Sosyal Bilimler Enstitüsü.

Şeyda Demirel, S. G. (2019). Karar ağacı algoritmaları ve çocuk işçiliği üzerine bir uygulama. *Sosyal Bilimler Araştırma Dergisi*, s. 52-65.

T. Cover, P. H. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, s. 21 - 27.

Tao Chen, Y. Z. (2022). Development of a Fully Automated Graf Standard Plane and Angle Evaluation Method for Infant Hip Ultrasound Scans. *Diagnostics (Basel), 12*(6), 1423.

Taşkın Kavzoğlu, İ. Ç. (2010). Karar Ağaçları İle Uydu Görüntülerinin Sınıflandırılması. *Harita Teknolojileri Elektronik Dergisi 2(1)*, s. 36-45.

Terence W O'Neill, P. S. (2018). Update on the epidemiology, risk factors and disease outcomes of osteoarthritis. *Best Pract Res Clinical Rheumatology, 32*(2), 312-326.

Thanh-Tu Pham, M.-B. L. (2021). Assessment of hip displacement in children with cerebral palsy using machine learning approach. *Medical & Biological Engineering & Computing, 59*(9), 1877-1887.

Tianmei Gua, J. D. (2017). Simple Convolutional Neural Network on Image Classification. *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)* (s. 721-724). IEEE.

Tim P.C. Kane, J. H. (2003). Radiological outcome of innocent infant hip clicks. *J Pediatr Orthop B.*, s. 259-263.

Tuğçe Yüce, M. K. (2021). Makine Öğrenmesi Algoritmaları ile Detay Üretim Alanları İçin İş Merkezi Kırılımında Üretim Süresi Tahminleme. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi 37(1)*, s. 47-60.

Umut Kaya, A. Y. (2019). Sağlık Alanında Kullanılan Derin Öğrenme Yöntemleri. *Avrupa Bilim ve Teknoloji Dergisi (16)*, s. 792-808.

V. Bialik, G. M. (1999, Jan). Developmental Dysplasia of the Hip: A New Approach to Incidence. *Pediatrics 103(1)*, s. 93-99.

Vaquero-Picado, A. (2023). Developmental Dysplasia of the Hip: Update of Mangement. *EFORT Open Rewiews 52(5)*, s. 281-287.

Vyawahare, D. (2022, August). Machine Learning: A Solution Approach for Complex Problems. *Internatıonal Journal Of Scıentıfıc Research In Engıneerıng And Management*, s. 6.

Weifeng Zhao, J. Q. (2015). Classification of Seizure in EEG Signals Based on KPCA and SVM. *SpringerLink*, s. 201-207.

Weize Xu, L. S. (2022). A Deep-Learning Aided Diagnostic System in Assessing Developmental Dysplasia of the Hip on Pediatric Pelvic Radiographs. *Frontiers in Pediatric, 9*, 785480.

Xiaohui Ding, Q. S. (2016). Classification of Motor Imagery EEG Signals with Support Vector Machines and Particle Swarm Optimization. *Computational and Mathematical Methods in Medicine*.

Xindi Hu, L. W. (2021). Joint Landmark and Structure Learning for Automatic Evaluation of Developmental Dysplasia of the Hip. *IEEE Journal of Biomedical and Health Informatics, 26*(1), 345-358.

Yann LeCun, Y. B. (2015). Deep learning. *Nature*, s. 436-444.

Yan-Yan SONG, Y. L. (2015). Decision tree methods: applications for classification and prediction. *Shanghai Arch Psychiatry 27(2)*, s. 130-135.

Yılmaz Kaya, A. Y. (2012). İki Durumlu Karışımlı Lojistik Regresyona İlişkin Bir Uygulama. *Bilişim Teknolojileri Dergisi*, s. 53-58.

Zhou, L. (2022). Recognition of depression patients with electroencephalogram. *International Journal of Biometrics 14(3-4)*, s. 481-491.