T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

VERİ MADENCİLİĞİ İLE OTİZM BELİRLENMESİ

YÜKSEK LİSANS TEZİ

Elif ÖZTAD

Bilgisayar Mühendisliği Ana Bilim Dalı

Bilgisayar Mühendisliği Programı

Eylül 2020

T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

VERİ MADENCİLİĞİ İLE OTİZM BELİRLENMESİ

YÜKSEK LİSANS TEZİ

Elif ÖZTAD
(Y1813.010005)

Bilgisayar Mühendisliği Ana Bilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Dr. Öğr. Üyesi Peri GÜNEŞ

Eylül 2020

ONUR SÖZÜ

Yüksek Lisans tezi olarak sunduğum "Veri Madenciliği ile Otizm Belirlenmesi" adlı çalışmanın süreçlerinde bilimsel ahlak ve etik geleneklere aykırı düşecek bir davranışımın olmadığını, çalışmadaki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bu eserlere atıf yaparak yararlanmış olduğumu belirtir ve onurumla beyan ederim.

Elif ÖZTAD

ÖNSÖZ

Bu çalışmanın yürütülmesi sırasında bana çok destek olan değerli danışmanım Dr. Öğr. Üyesi Peri Güneş'e, hayatım boyunca hep yanımda olan çok sevdiğim aileme, engin bilgisi ve deneyimleriyle her zaman daha iyi olmamız için bize destek olan sayın hocam Prof. Dr. Ali Güneş'e, yaptığım çalışmada psikoloji alanında desteklerini esirgemeyen çok sevgili arkadaşım Klinik Psikolog Şebnem Aydın'a sevgilerimi ve teşekkürlerimi sunarım.

Her ihtiyaç duyduğumda sabırla sorularımı cevaplayan ve yanımda olan, okulun bana katmış olduğu değerli arkadaşım Othmane Elmeziani'ye sonsuz teşekkürlerimi sunarım.

Ve son olarak, bilgi ve birikimlerini benimle paylaşan, her zaman destek olan değerli arkadaşım Öğr. Gör. Çağdaş Özer'e sonsuz sevgi, saygı ve teşekkürlerimi sunarım.

Ağustos, 2020 Elif ÖZTAD

İÇİNDEKİLER

KISALTMALAR

QCHAT **:** Yeni yürümeye başlayan çocuklarda otizm için nicel kontrol listesi (Quantitative Checklist for Autism in Toddlers)
OSB **:** Otistik Spektrum Bozukluğu
AC **:** İlişkilendirme Sınıflandırması (Association Classification)
WCBA **:** İlişkilendirmeye dayalı ağırlıklı sınıflandırma (Weighted classification based on association)
CNN **:** Konvolüsyonlu Sinir Ağı (Convolutional Neural Network)
NB **:** Naif Bayes (Naive Bayes)
LR **:** Lojistik Regresyon (Logistic Regression)
DT **:** Karar Ağaçları (Decision Trees)
SVM **:** Destek Vektör Makinesi (Support Vector Machine)
RF **:** Rastgele Orman (Random Forest)
BTA **:** Başka Türlü Adlandırılamayan
YGB **:** Yaygın Gelişimsel Bozukluk

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

VERİ MADENCİLİĞİ İLE OTİZM BELİRLENMESİ

ÖZET

Bu çalışmada, otizm konusunda net bir teşhisin hemen konulamaması, Türkiye'de ve dünyada otizm teşhisi konan çocuk sayısının hızla artması ancak farkındalığın çok az olması problemlerinin çözümüne fayda sağlayabilmek amacıyla makine öğrenmesi ve veri madenciliği tekniklerinden yararlanılmıştır. Otizm için kesin tanı koyan bir test henüz geliştirilmediği için tanı koymada en etkili sonuç elde etmeyi sağlayan QCHAT adı verilen testin sonuçlarına göre veriler analiz edilip, sonuçların görsel açıdan kolay anlaşılabilmesi için veri görselleştirme de yapılarak veri madenciliği algoritmalarının hangisinin daha yüksek doğruluk değeri verdiği karşılaştırılmıştır. Ayrıca bu çalışmada kullanılan Lojistik Regresyon, Karar Ağaçları, Naif Bayes, Destek Vektör Makinesi ve Rastgele Orman algoritmaları RapidMiner programında analiz edilip çıkan sonuçlar karşılaştırılmıştır. Hangi algoritmanın en yüksek doğruluğu verdiğini ölçmek adına Doğruluk (Accuracy Analizi), sınıflandırma hatasının hangi algoritmada en fazla olduğunu bulmak adına Sınıflandırma Hatası (Classification Error) Analizi, hassasiyetin hangi algoritmada en fazla olduğunu bulmak adına Hassasiyet (Precision) Analizi yapılmıştır. Aynı zamanda Geri Çağırma, Testin Doğruluğu, Gerçek Pozitifler Oranı ve Belirlilik Analizleri de yapılıp, sonuçlar algoritma bazında karşılaştırılmıştır. Kaggle sitesinden alınan otizm veri seti ile Jupyter Notebook'ta yapılan analize göre en yüksek doğruluğu Lojistik Regresyon algoritması vermiştir.

Anahtar Kelimeler: Otizm, Veri Madenciliği, Osb, Derin Öğrenme, Sınıflandırma

AUTISM DIAGNOSIS WITH DATA MINING

ABSTRACT

In this study, machine learning and data mining techniques were utilised to help solving the problems such as inability to make a clear diagnosis in autism, rapid increase of the number of children diagnosed with autism in Turkey and in the world. Since a test with a definitive diagnosis for autism has not been developed yet, the data was analyzed according to the results of the test called QCHAT, which provides the most effective result in making a diagnosis, and which data mining algorithms give higher accuracy value was compared by performing data visualization in order to understand the results easily. In addition, Logistic Regression, Decision Trees, Naive Bayes, Support Vector Machine and Random Forest algorithms were analyzed in RapidMiner and the results were compared. Accuracy Analysis was performed to measure which algorithm gives the highest accuracy, Classification Error Analysis to find out which algorithm has the highest classification error, and Precision Analysis to find out which algorithm has the highest sensitivity. At the same time; Recall, F-Measure, Sensitivity and Specificity Analyzes will be performed and the results were compared based on the algorithm. According to the analysis performed in Jupyter Notebook with the autism data set taken from the Kaggle website, the Logistic Regression algorithm gave the highest accuracy.

Keywords: Autism, Data Mining, Asd, Deep Learning, Classification

I. GİRİŞ

Makine öğrenmesi ve Veri Madenciliği tıpta farklı alanlarda kullanılmıştır ve kullanılmaya devam etmektedir. Örneğin Parkinson hastalığı ile ilgili çalışmalarda, epileptik nöbetlerin tespitinde, farmakoloji, patoloji ve radyoloji alanlarında makine öğrenmesi ve veri madenciliği tekniklerinin kullanımı her geçen gün artmaktadır. Bu alanlarda bu denli başarı elde etmesinden yola çıkılarak otizm spektrum bozukluğu için veri madenciliğindeki algoritmalar kullanılıp hangi algoritmanın daha yüksek doğruluğa ulaşacağı karşılaştırılmak istenmiştir.

Otizm spektrum bozukluğu; Otizm, Asperger Sendromu, Yaygın Gelişimsel Bozukluk (YGB) ve Başka Türlü Adlandırılamayan yaygın gelişimsel bozukluk (BTA) olmak üzere üçe ayrılmaktadır (Url-1). Otizm terimi Almanca “autismus” yani "kendisi" anlamına gelir ve ilk olarak yetişkin hastalarda şizofrenik semptomları tanımlamak için 1911'de İsviçreli psikiyatrist Eugen Bleuler tarafından tanıtılmıştır (Tordjman, 2011). Otizme neyin sebep olduğu kesin olarak bulunamamış olup, genel olarak beyin yapısını veya işlevini etkileyen bir takım sinir sorunlarından kaynaklandığı düşünülmektedir (Url-1).

Otizm teşhisini doktorlar belirli kriterlere göre koymaktadır. Bu kriterlerden biri olan davranışsal bozukluğu ise psikologlar çeşitli testler yaparak ölçeklendirmektedir. Denver, Q-Chat gibi testler 0-6 yaş arası çocuklarda çocuğa bakım veren kişi (anne, baba vs.) ile uygulanmakta, 6 yaşından büyük çocuklarda ise birebir çocuk ile uygulanmaktadır. Örneğin çocuk 3 yaşındaysa annesine veya bakım sağlayan kişiye çocuktan küpleri üst üste dizmesini söylemesi isteniyor. Nasıl dizdiğine göre, annesine verdiği tepkilere, göz teması kurup kurmamasına göre yaşıtlarından kaç ay geri olduğu ölçülebilmektedir.

A. Problem

Bu tezde, çözümlenmesi istenilen sorunlar şunlardır:

- Veri Madenciliği algoritmaları kullanarak otizm tahmin edilebilir mi?

- Bu çalışmada yapılan tahmin otizm için yapılan testlere katkı sağlayabilir mi?
- En yüksek doğruluğu hangi algoritma verecektir?

1. Motivasyon

Türkiye'de ve dünyada otizm teşhisi konan çocuk sayısının hızla artması ancak farkındalığın çok az olması problemlerinden yola çıkılarak bu probleme veri madenciliği yöntemlerinden yararlanarak bir çözüm sunup, en doğru ve en hızlı sonucu veren algoritmayı tespit edebilmek bu çalışmanın motivasyonunu oluşturmaktadır.

2. Hedefler

Bu çalışmada ulaşılmak istenen hedefler aşağıdaki gibidir:

- Otizmli bir bireye ait veri setinin işlenmesi, verilerin eğitim ve test olarak ayrıştırılması
- Veri setindeki veriler işlendikten sonra veri madenciliği algoritmaları kullanılarak hangi algoritmanın en yüksek doğruluğu vereceğinin tespit edilmesi
- Her bir algoritma için sonuçların karşılaştırılması
- Doğruluk oranını arttırmak ve otizm tespitinde kullanılabilmesi için yapılabilecek çalışmalar

II. LİTERATÜR TARAMASI

Bu bölümde, Veri Madenciliği algoritmalarının hangi alanlarda kullanıldığı, bu algoritmalar ile nasıl sonuçlar alındığı anlatılmıştır. Ayrıca otizm konusunda Veri Madenciliği'nden yararlanılarak yapılan uluslararası çalışmalara da örnekler verilmiştir.

A. Otizm ve Veri Madenciliği

Otizm, sosyal etkileşimlerin bozulması ve stereotipik aktivitelerle ilişkili iletişim yeteneğindeki değişikliklerle karakterize bir gelişimsel bozukluk olarak tanımlanır (Chamak ve Cohen). Otizme neden olan çeşitli faktörler olduğu bilinmektedir ancak kesin bir bağlantı kurulamamıştır. Araştırmalar, otizm teşhisi konmuş kişilerin beyinlerinde kimyasal ve yapısal bir takım farklılıklar olduğuna dair kanıt bulmuşlardır(Url-2). Literatürde, Sylvie Tordjman otizmin genetik faktörlerden kaynaklanıp kaynaklanmadığını ölçmek için ikizler ile çalışma yapmış olup, olası faktörleri oluşturacak çevresel faktörlerin ve epigenez mekanizmalarının otistik bozukluklarının gelişimindeki rolünü açıkça vurgulamıştır (Url-3). Bir diğer çalışmada ise otizmin güçlü bir genetik bozukluk olduğu ve muhtemelen birden fazla gen nedeniyle ortaya çıktığı; tek çocuklu ailelerde tekrarlama oranlarının yüksek olduğu sonucuna varılmıştır (Url-4). Bir çocukta otizmin en belirgin işareti sosyal olarak etkileşime geçmemesidir. Bebekler sesli uyaranlara veya ebeveynlerine tepki vermeyebilir. Çocuklar başkalarını gözleriyle takip edemeyebilir, hatta göz teması kuramayabilir. Araştırmacılar yıllardır otizm konusunda çeşitli araştırmalar yapmışlardır ve günümüzde otizmin olgunlaşmamış beynin gelişimsel bir bozukluğu olduğu görüşünü öne sürmektedirler. Şu anda otizm bozukluğunun kesin tanısını koymak için net bir bilimsel yöntem bulunmamaktadır. Otizm tanısı koyabilmek için birden fazla test ve tanı sistemleri mevcuttur.

Aşağıda, Hastalık Kontrol ve Önleme Merkezleri (Centers for Disease Control and Prevention-CDC)'nin internet sitesinden alınan, OSB'li bireylerin sahip olabileceği semptomlardan bazıları gösterilmiştir (Url-1).

- İlgisini çektiğini belli edici davranışlarda bulunmama (örneğin, üzerinden geçen bir uçağı işaret etmeme)
- Başkasının işaret ederek gösterdiği nesnelere bakmama
- Başkalarıyla ilişkilerde sorun yaşama ya da başkalarıyla hiç ilgilenmeme
- Göz temasından kaçınma ve yalnız kalmak isteme
- Başkalarının duygularını anlamakta ve kendi duygularını anlatmakta güçlük çekme
- Sarılmaktan kaçınma veya sadece kendi istediğinde sarılma
- İnsanlar konuşurken farkında değilmiş gibi görünme ancak diğer seslere tepki verebilme
- İnsanlarla çok ilgili olmak ancak onlarla nasıl konuşulacağını, oynanacağını veya nasıl ilişki kurulacağını bilmeme
- Bir rutin değişikliğinde uyum sağlayamama
- Bir zamanlar sahip oldukları becerileri kaybetmek

Literatürde veri madenciliği yöntemleriyle yapılan birçok çalışma mevcuttur. Karar Ağaçları kullanarak yaptıkları çalışmada (DUYGU, KOCAOĞLU ve COŞKUN.) veri setindeki otizm üzerinde etkili faktörlerin ilişkisini incelenmiş ve özellikle etnik yapının bu hastalık üzerinde gözle görülür bir etkisi olup olmadığını araştırmıştır ancak sonuç olarak büyük bir etkisinin olmadığı görülmüştür.

Başka bir çalışmada (Alwidian, Elhassan ve Ghnemat, 2020), İlişkilendirme Sınıflandırması (AC) tekniğini kullanarak bireyin otizmi olup olmadığını tespit etmeye çalışmıştır. Erken teşhisin önemi göz önünde bulundurularak, yedi farklı algoritma ile AC tekniğinin performansının analizini ve değerlendirmesini yapmışlardır ve sonuç olarak algoritmalar arasında karşılaştırmalı performans analizi kullanılmıştır. Sonuç olarak bu yedi algoritma içinden WCBA algoritması en yüksek doğruluğu vermiştir.

Andrius Vabalas, Emma Gowen, Ellen Poliakoff ve Alexander J. Casson "Otizm Teşhisini Tahmin Etmek İçin Bir Hareket Taklit Görevinin Kinematik ve Göz Hareketi Özelliklerine Makine Öğrenimini Uygulama" isimli çalışmalarında basit bir taklit

görevinin otistik ve otistik olmayan bireyleri ayırt edip edemeyeceğini ve otizme özgü motor farklılıkları karakterize edip edemeyeceğini araştırmışlardır. Bu çalışmada, bir hareket taklidi görevinden alınan hareket ve göz izleme verileri, 22 otistik ve 22 otistik olmayan yetişkini sınıflandırmak için denetimli makine öğrenme yöntemleri kullanmışlardır. Amaç, güvenilir bir makine öğrenmesi uygulaması elde etmektir. Kullanılan modeller tanıyı kinematik özelliklerden%73, göz hareketi özelliklerinden%70 doğruluk ve birleşik özelliklerden%78 doğrulukla tahmin etmiştir. Otizmdeki hareket taklidi farklılıklarını daha iyi tahmin edebilmek için en önemli olan özellikler araştırılmıştır. Davranışsal sonuçlarla tutarlı olarak, ayırt edici özelliklerin çoğu, otistik olmayan bireylerin olağandışı hareket kinematiğini başarılı bir şekilde taklit etme eğilimindeyken otistik bireylerin başarısız olma eğiliminde olduğu görülmüştür. Makine öğrenimi sonuçları, gelecekteki çalışmaların mevcut kalitatif testleri tamamlayacak nicel testler sağlayarak teşhis sürecine yardımcı olacağı sonucu elde edilmiştir.

M. S. Mythili ve A. R. Mohamed Shanavas "Sınıflandırma Teknikleri Kullanılarak Otizm Spektrum Bozuklukları Üzerine Bir Çalışma" isimli çalışmalarında veri madenciliği sınıflandırma algoritmaları yardımıyla otizm düzeylerini tespit etmeyi amaçlamışlardır. Otizm ve çeşitli otizm bozukluklarından bahsedilip popüler makine öğrenme yöntemlerinin etkinliğini Yapay Sinir Ağı (algılayıcı), Destek Vektör Makinesi ve bulanık mantıkla karşılaştırmışlardır. Bu algoritmaların, otizmli öğrencilerin tahmin düzeyini ele almak için çok kullanışlı olduğu sonucuna varmışlardır. Gelecekte, otizm bozukluğu sınıflandırması için bulanık bilişsel harita ve arı kovanı sürüsü optimizasyonu tekniğinin kullanılmasını önermişlerdir.

Milan N. ve diğerlerinin yapmış olduğu « Optimize Edilmiş Makine Öğrenimi Modelleri ve Kişisel Karakteristik Verilerle Otizm Teşhisini Geliştirmek » çalışmasında, günümüzde OSB teşhisi için tek klinik yöntem, daha uzun teşhis süresi ve artan tıbbi maliyetler gerektiren standartlaştırılmış ASD testleri olduğunu vurgulamış ve amaçlarının, OSB'nin önceki tanı modellerini iyileştirmek için büyük, iyi karakterize edilmiş bir veri setinden kişisel karakteristik verilerin (PCD) tahmin gücünü araştırmak olduğunu belirtmişlerdir. Otizm Beyin Görüntüleme Veri Değişimi (ABIDE) veritabanındaki 851 kişiden altı kişisel özellik (yaş, cinsiyet, el tercihi ve üç bireysel IQ ölçümü) çıkarmışlardır. ABIDE, 17 araştırma ve klinik enstitüsünden çok sayıda ASD hastasından ve tipik ASD olmayan kontrollerden veri toplayan

uluslararası bir ortak projedir. Bu kamuya açık veritabanını, dokuz denetimli makine öğrenimi modelini test etmek için kullanmışlardır. Tipik ASD olmayan kontroller ile ASD hastaları arasında sınıflandırma için bu makine öğrenimi modellerini eğitmek ve test etmek için bir çapraz doğrulama stratejisi uygulamışlardır. Alıcı çalışma karakteristik eğrisi (AUC) altındaki doğruluk, duyarlılık, özgüllük ve alanı kullanarak sınıflandırma performansını değerlendirmişlerdir. Altı kişisel özellik kullanarak test ettikleri dokuz modelden, sinir ağı modeli en iyi performansı 0.646 (0.005) ortalama AUC (SD) ile ve ardından 0.641 (0.004) ortalama AUC (SD) ile k-en yakın komşu ile performans gösterdiğini tespit etmişlerdir. Bu çalışma, PCD ile optimal bir ASD sınıflandırma performansı oluşturmuştur. Ek ayırt edici özelliklerle (ör. Nörogörüntüleme), makine öğrenimi modellerinin otizmin otomatik klinik teşhisini sağlayabileceği sonucuna ulaşmışlardır.

Kayleigh K. ve diğerleri "Otizm Spektrum Bozukluğu Araştırmasında Denetimli Makine Öğreniminin Uygulamaları: Bir Gözden Geçirme" çalışmalarında, ASD'de denetimli makine öğrenimini kullanan, sınıflandırma ve metin analizi için algoritmalar dahil olmak üzere 45 makalenin kapsamlı bir incelemesini yapmışlardır. Makalenin amacı, ASD literatüründeki denetimli makine öğrenimi eğilimlerini belirlemek ve açıklamak, ayrıca ASD verilerinin madenciliği için klinik, hesaplamalı ve istatistiksel olarak sağlam yaklaşımların gövdesini genişletmek isteyen araştırmacıları bilgilendirmek ve yönlendirmektir. Yaptıkları literatür taraması Endnote'tan 27 ve Google Scholar'dan 94 yayın üretmiştir. İncelenen makalelerden 35'inin ASD araştırmasında denetimli makine öğreniminin kullanıldığını tespit etmişlerdir.

Devika Varshini G and Chinnaiyan R yaptıkları "Otizm Spektrum Bozukluğunun Tahmini için Optimize Edilmiş Makine Öğrenimi Sınıflandırma Yaklaşımları" çalışmasında yeni yürümeye başlayan çocuklarda ve yetişkinlerde erken otizm özelliklerini tahmin etmek için kullanılan tıbbi veri setlerinin sınıflandırılması görevi için çeşitli makine öğrenme algoritmalarının ve ön işleme tekniklerinin etkinliğini değerlendirmişlerdir. Bu yöndeki önceki bazı çalışmalar, etkili bir klasikleştirme için karmaşık ön işleme ve makine öğrenimi tekniklerini kullanmıştır. Bununla birlikte, bu deney, uygun veri kodlaması ve lojistik regresyon, KNN ve Random Forest gibi farklı sınıflandırıcı algoritmalarıyla birleştirilen basit bir ön işleme adımlarının, son teknoloji ile karşılaştırılabilir sonuçlar doğurduğunu ortaya koymaktadır.

Fatiha Nur Büyükoflaz Ali Öztürk "Makine Öğrenmesi Algoritmaları ile Çocuklarda Erken Otizm Teşhisi" çalışmasında UCI 2017 Autistic Spectrum Disorder Screening Data for Children veri kümesi üzerinde, Naive Bayes, IBk (k-en yakın komşu), Radyal Temel Fonksiyon Ağı (RBFN) ve Rasgele Orman(RO) olmak üzere dört farklı sınıflandırma yöntemi kullanmışlardır. Elde ettikleri sonuçta Rasgele Orman yönteminin Naive Bayes, IBk ve RBFN yöntemlerinden daha başarılı olduğunu tespit etmişlerdir. Veriyi 244 tanesi eğitim, 58 tanesi ise test verisi olacak şekilde ikiye bölmüşlerdir. Kullandıkları sınıflandırıcıların performans analizinden elde ettikleri sonuçlara göre veri seti üzerinde IBk %89.65, Naive Bayes %96.55, RBFN %98.27, Rasgele Orman ise %100 başarı sağlamıştır.

Yeni Yürümeye Başlayan Çocuklarda Otizm Taraması için Makine Öğrenimi Stratejisi çalışması ile Luke E. K. Achenie ve diğerleri ileri beslemeli sinir ağını (fNN) kullanarak, ASD taramasının önündeki engellerin üstesinden gelmek için otomatik bir makine öğrenimi (ML) yöntemi geliştirmeyi denemişlerdir. Bu çalışmada fNN tekniği 14.995 yeni yürümeye başlayan çocuğun (16-30 ay, %46.51 erkek) arşivlenmiş M-CHAT-R verileri kullanılarak uygulanmıştır. Örnek, alt grup farklılıklarını incelemek için ırk (yani, beyaz ve siyah), cinsiyet (yani, erkek ve kız) ve anne eğitimi (yani 15 yıllık eğitimin altı ve üstü) olarak alt gruplara ayrılmıştır. Her alt grup, en iyi performans gösteren fNN modelleri için değerlendirilmiştir ve sonuç olarak toplam örneklem için, en iyi sonuçlar 18 madde kullanılarak % 99.72 doğru sınıflandırma verdiği görülmüştür. Bu nedenle MO, insan hatasını ortadan kaldıran ve önceki tarama yöntemlerine göre bir avantaj sağlayan, verimli puanlamanın uygulanmasında yararlı bir araç olabilir şeklinde yorumda bulunmuşlardır.

Azian Azamimi Abdullah, Saroja Rijal ve Satya Ranjan Dash'ın yapmış oldukları Otizm Spektrum Bozukluğunun (ASD) Sınıflandırılmasına Yönelik Makine Öğrenimi Algoritmalarının Değerlendirilmesi isimli çalışmalarında ASD'yi sınıflandırmak için daha yüksek potansiyele sahip modeller oluşturmak için Otizm Spektrum Soruları'nı kullanma üzerine bir deneme geliştirilmiştir. Bu araştırmada, Rastgele Orman, Lojistik Regresyon ve K-En Yakın Komşular olmak üzere 3 denetimli makine öğrenme algoritması için en önemli özellikleri seçmek üzere özellik seçim yöntemi olarak Ki-kare ve En Az Mutlak Çekme ve Seçim Operatörü (LASSO) seçilmiştir. Performans, Ki-kare seçim yöntemine göre seçilen 13 özelliğe sahip model kullanılarak Lojistik Regresyonun% 97.541 ile en yüksek doğruluğu puanladığı sonuçlarda

değerlendirilmiştir. Bu yüzden Lojistik Regresyon'un performansının, ASD'nin gelecekteki tespiti için uygulanabilecek davranış gözlemi amacıyla yüksek doğruluk sağladığı sonucuna varılmıştır.

Ayşe DEMİRHAN yapmış olduğu "Otizm Spektrum Bozukluk Vakalarını Belirlemede Makine Öğrenme Yöntemlerinin Performansı" isimli çalışmada, ASD ergen tarama verileri kullanarak destek vektör makineleri (SVM), k-en yakın komşu (kNN) ve rastgele orman (RF) algoritmalarını kullanarak ASD durumunun hızlı ve doğru teşhisi için analizler yapmıştır. SVM, kNN ve RF yöntemleri kullanılarak 10 kat çapraz doğrulama (CV) ile ikili sınıflandırma sonucunda sırasıyla %95, %89 ve %100 doğruluk oranları elde etmiştir. Ayrıca RF yöntemi ile yapılan sınıflandırmadan %100 duyarlılık ve özgüllük değerleri elde etmiştir. Bu çalışma ile ASD erişkin tarama verileri kullanılarak RF yöntemi ile sınıflandırma sonucunda OSB vakalarının tam bir başarı ile tespit edilebileceği sonucuna varmıştır.

1. Veri Önişleme

Gereksiz bilgileri temizlemek için veri setini temizlek oldukça önemlidir. Kullanmayacağımız özelliklerin elimine edilmesi gerekmektedir. Bu bölümde, verilerin tahmine dayalı analize uygun olması için veri ön işleme ve dönüştürme işlemleri yapılmıştır. Model için girdi olarak 10 soru yanıtı, yaş, cinsiyet ve sarılık özellikleri kullanılmıştır. Kullanılan veri kümesinin açıklaması Çizelge 1'de gösterilmiştir.

Çizelge 1. Veri Setindeki Özellikler, Tip ve Tanımları (Url-5)

Özellik	Tip	Tanım
A1	Binary (0,1)	Çocuğunuz siz seslendiğinizde size bakıyor mu?
A2	Binary (0,1)	Göz teması kuruyor mu?
A3	Binary (0,1)	Bir şey istediğinde eliyle işaret ediyor mu?
A4	Binary (0,1)	İlgisini çeken bir şey olduğunda size gösteriyor mu? (İlginç bir manzarayı işaret ederek size göstermek gibi)
A5	Binary (0,1)	Yap-inan oyunları oynuyor mu(Bebeklerle evcilik oyunu, telefonda taklit yapma gibi)
A6	Binary (0,1)	Çocuğunzu baktığınız yeri takip ediyor mu?
A7	Binary (0,1)	Eğer siz veya ailede başka biri üzgün görünüyorsa, çocuğunuz onu rahatlatmak istediğine dair bir işaret gösteriyor mu? (Sarılmak gibi)
A8	Binary (0,1)	Çocuğunuzun ilk kelimeleri sıradan mıydı?
A9	Binary (0,1)	Çocuğunuz basit jest ve mimikler yapıyor mu? (El sallamak gibi)
A10	Binary (0,1)	Çocuğunuz görünürde bir amacı olmadan hiçbir şeye bakmıyor mu?
Yaş	Integer	Ay cinsinden
Q-Chat-10 skoru	Integer	1-10(3'ten küçük veya eşitse OSB tanısı konmaz. 3'ten büyükse OSB tanısı konur.
Cinsiyet	K ya da E	Kız yada Erkek
Irk	String	Orta doğulu, Beyaz Avrupalı Hispanik, Asyalı, Güney Asyalı, Siyahi, Latin, Yerli Hintli, Siyahi ve Diğer
Doğuştan sarılık var mı?	Boolean (Evet ya da Hayır)	Evet ya da Hayır
Ailede OSB var mı?	Boolean (Evet ya da Hayır)	Evet ya da Hayır
Soruları kim cevaplıyor?	String	Kendisi ya da Aile Üyesi
OSB Tanısı	String	Evet ya da Hayır

a. Özellik Seçimi

Özellik seçimi genelde doğruluk ve ölçeklenebilirlik amacıyla kullanılmaktadır (Yazıcı, vd.). Veri setini işlerken özellik seçimi büyük önem taşımaktadır. Özellik seçim teknikleri bir dizi kritere göre kategorize edilebilmektedir. Popüler bir kategorizasyon, özniteliklerin değerini değerlendirmek için kullanılan metriğin yapısını tanımlamak için filtre ve sarmalayıcı terimlerini icat etmiştir (Berminghan, vd. 2015). Veri madenciliğinde veri setindeki boyut fazla olduğunda doğruluk elde etmek zorlaşabilir. Örneğin elimizde 10 öğrencinin verisi var, matematik notlarını sınıflandırmak istiyoruz, eğer veri setinde bu öğrencilerin kaç beden pantolon giydiği bilgisi var ise bu hem boyut fazlalığına yol açar hem de asıl amacımızdan uzaklaştırabilir.

Burada amaç, makine öğrenme sisteminde öğrenmeyi olumlu şekilde etkileyen nitelikleri seçip, olumsuz yönde sistem zarar veren nitelikleri de ortadan kaldırmaktır. (URL-6).

B. Sınıflandırma ve Algoritmalar

Sınıflandırma, önemli veri sınıflarını tanımlayan modelleri çıkaran bir veri analizi biçimidir (Han, vd. 2011). Aynı zamanda yeni bir gözlemin hangi kategorilerden hangisine ait olduğunu belirlemeyi amaçlar. Ayhan ve Erdoğmuş 2014 yılında konuyla ilgili yaptıkları çalışmada algoritmanın genelleme performansının sınıflandırma problemlerinin çözümü için geliştirilen makine öğrenimi algoritmasının seçiminde dikkat edilecek en önemli kriterlerden biri olduğunu vurgulamıştır (Ayhan ve Erdoğmuş, 2014).

1. Sınıflandırma Algoritmaları

Her bir hedef sınıfı belirlemede kullanılabilecek daha iyi sınır koşulları elde etmek için eğitim veri setini kullanırız. Sınır koşulları belirlendikten sonra, bir sonraki görev hedef sınıfı tahmin etmektir. Örneğin: Bilgisayar aksesuarları satın alıp almayacağını tahmin etmek için müşteri verilerinin analizi (Hedef sınıf: Evet veya Hayır) Meyveleri renk, tat, boyut, ağırlık gibi özelliklerden sınıflandırmak (Hedef sınıflar: Elma, Portakal, Kiraz, Muz) Cinsiyet saç uzunluğuna göre sınıflandırma (Hedef sınıflar:

Erkek veya Kadın) (URL-7). Bu projede kullanılan sınıflandırma algoritmaları aşağıdaki gibidir:

- Lojistik Regresyon (LR)
- Karar Ağaçları (DT)
- Naif Bayes (NB)
- Destek Vektör Makinesi (SVM)
- Rastgele Orman (RF)

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import RandomForestClassifier
```

Şekil 1. Sınıflandırma Algoritmaları Kod Görüntüsü

III. MATERYAL VE METOTLAR

Bu kısımda, veri setinde veri ön işleme hakkındaki bilgilere, kullanılan kütüphanelere ve hangi algoritmaların kullanıldığına değinilecektir.

A. Otizm Veri Seti

Bu projede kullanılan veri seti, 1054 satır ve 18 sütundan oluşmaktadır ve Kaggle internet sitesinden elde edilmiştir. Bu veri setinde; çocuğun veya ona bakım veren ebeveynin cevaplaması gereken, Q-Chat-10 testinde yer alan 10 soru (A1'den A10'a kadar) yer almaktadır. Sorular; çocuğunuz siz seslendiğinizde size bakıyor mu, göz teması kuruyor mu gibi 10 sorudan oluşmaktadır. Veri setindeki "Evet" yanıtı string değerden binary değere yani "1" e, "Hayır" yanıtı ise "0" a eşlenmiştir. Ayrıca veri setinde ırk, cinsiyet, aile bireylerinde ASD olup olmadığı, sarılık bilgisi, sorulara kimin cevap verdiği bilgisi mevcuttur. Doğruluğu arttırmak için veri setinde öncelikle gereksiz özelliklerin elenmesi için ön işleme yapılmıştır.

Ön işleme sonrasında veri setinin ekran görüntüsü Şekil-2'de gösterilmiştir:

	Case_No	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Age_Mons	Qchat-10-Sc	Sex	Ethnicity	Jaundice	Family_mem	Who comple	Class
2	1	0	0	0	0	0	0	1	1	0	1	28	3	f	middle easte	yes	0	family mem	No
3	2	1	1	0	0	0	1	1	0	0	0	36	4	m	White Europ	yes	0	family mem	Yes
4	3	1	12	0	0	0	0	1	1	0	1	36	4	m	middle easte	yes	0	family mem	Yes
5	4	1	1	1	1	1	1	1	1	1	1	24	10	m	Hispanic	no	0	family mem	Yes
6	5	1	1	0	1	1	1	1	1	1	1	20	9	f	White Europ	no	1	family mem	Yes
7	6	1	1	0	0	1	1	1	1	1	1	21	8	m	black	no	0	family mem	Yes
8	7	1	0	0	1	1	1	0	0	1	0	33	5	m	asian	yes	0	family mem	Yes
9	8	0	1	0	0	1	0	1	1	1	1	33	6	m	asian	yes	0	family mem	Yes
10	9	0	0	0	0	0	0	1	0	0	1	36	2	m	asian	no	0	family mem	No
11	10	1	1	1	0	1	1	0	1	1	1	22	8	m	south asian	no	0	Health Care I	Yes
12	11	1	0	0	1	0	1	1	0	1	1	36	6	m	Hispanic	yes	1	family mem	Yes
13	12	1	1	1	1	0	1	1	1	0	1	17	8	m	middle easte	yes	0	family mem	Yes
14	13	0	0	0	0	0	0	0	0	0	0	25	0	f	middle easte	yes	0	family mem	No
15	14	1	1	1	1	0	0	1	0	1	1	15	7	f	middle easte	yes	0	family mem	Yes
16	15	0	0	0	0	0	0	0	0	0	0	18	0	m	middle easte	no	0	family mem	No
17	16	1	1	1	0	1	0	1	1	0	1	12	7	m	black	no	0	family mem	Yes
18	17	0	0	0	0	0	0	0	0	0	0	36	0	m	middle easte	no	1	family mem	No
19	18	1	1	1	0	1	1	1	1	0	1	12	8	f	middle easte	yes	0	family mem	Yes
20	19	1	0	0	0	1	0	0	0	0	1	29	3	f	middle easte	no	0	family mem	No

Şekil 2. Veri Setindeki İlk 20 Kaydın Görüntüsü

B. Kullanılan Kütüphaneler

Aşağıda, bu çalışma içerisinde yararlanılan kütüphanelerin bir listesi verilmiştir.

• NumPy: Numerik Python'ın kısaltmasıdır. Bu kütüphane aynı zamanda temel doğrusal cebir fonksiyonlarını, Fourier dönüşümlerini, gelişmiş rasgele sayı

yeteneklerini ve Fortran, C ve C ++ gibi diğer düşük seviyeli dillerle entegrasyon için araçlar içerir.

• Matplotlib: Histogramlardan çizgi grafiklerine ve ısı çizimlerine kadar çok çeşitli grafikleri çizmek için kullanılır. Bu çizim özelliklerini satır içi kullanmak için ipython not defterinde (ipython notebook –pylab = satır içi) Pylab özelliği kullanılabilir.

• Pandas: Veri işleme ve hazırlama için yaygın olarak kullanılır.

• Seaborn: İstatistiksel veri görselleştirme için kullanılır. Python'da çekici ve bilgilendirici istatistiksel grafikler oluşturmak için kullanılır (Url-8).

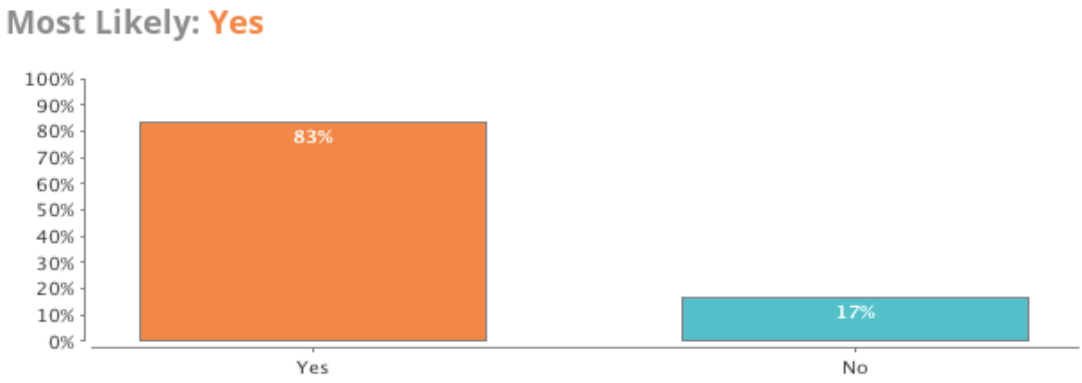
C. Metodoloji

Bu bölümde, projede kullanılan algoritmalardan bahsedilip, grafik, Çizelge ve şekiller sunulmuştur. Çalışma Python dilinde, Jupyter Notebook ortamında yapılmış olup, grafikler için Matplotlib fonksiyonu, şekiller ve çizelgeler için ise RapidMiner (RapidMiner Studio 9.4.001) programı kullanılmıştır. Öncelikle veri setinde temizleme işlemi yapılıp sadece kullanılacak verilerin kalması sağlanmıştır. Veri setinin bir kısmı eğitim, bir kısmı da test için ayrılmıştır.

1. Lojistik Regresyon (LR)

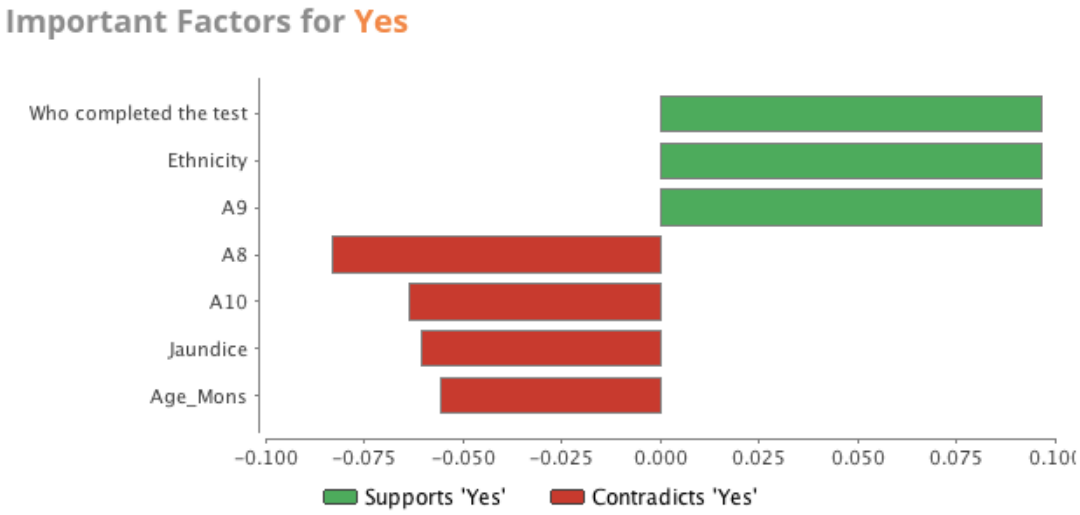
Lojistik regresyon, birden fazla bağımsız değişken ile kategorik bir bağımlı değişken arasındaki ilişkiyi analiz eder ve verileri bir lojistik eğriye uydurarak bir olayın meydana gelme olasılığını tahmin eder (Park, 2013). Lojistik regresyon gibi popüler istatistiksel prosedürler, nadir olayların olasılığını keskin bir şekilde hafife alabilmektedir (King ve Zeng, 2001).

Lojistik regresyon sınıflandırma amacı için tasarlanmıştır ve en çok birkaç bağımsız değişkenin tek bir sonuç değişkeni üzerindeki etkisini anlamak için yararlıdır ancak yalnızca tahmin edilen değişken ikili olduğunda çalışması, tüm öngörücülerin birbirinden bağımsız olduğunu ve verilerin eksik değerler içermediğini varsayması bir dezavantajdır (Url-9).



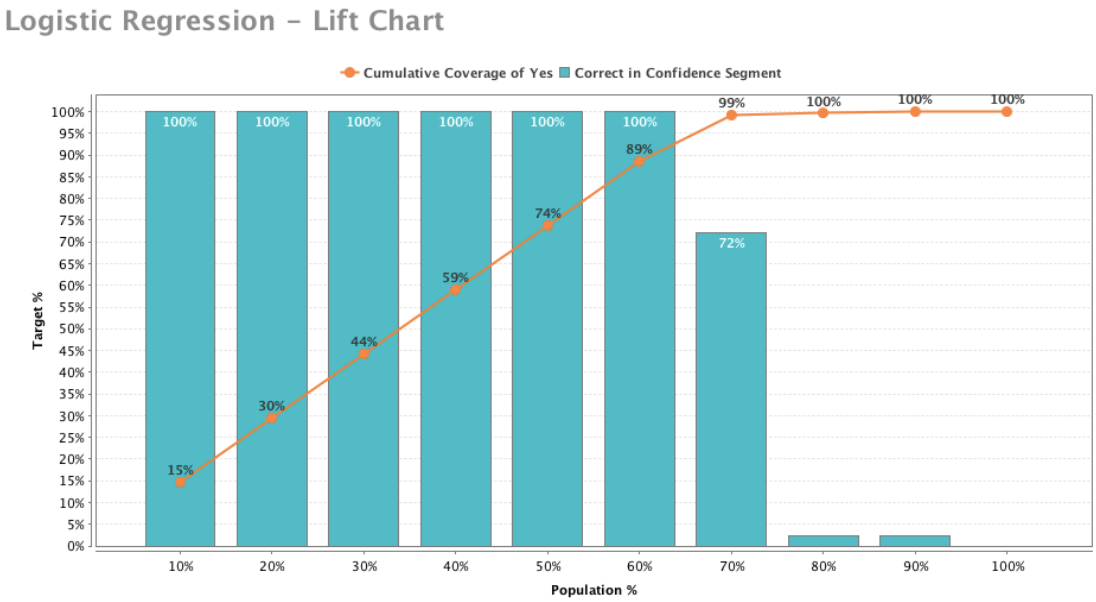
Şekil 3. Lojistik Regresyon için Simülasyon Sonucu

OSB Tanısı "Evet" için önemli olan faktörler Şekil 4'deki gibidir.



Şekil 4. OSB Tanısı Evet için Önemli Faktörler (LR)

Lojistik regresyon algoritmasının yükseliş grafiği Şekil 5'de gösterilmiştir.



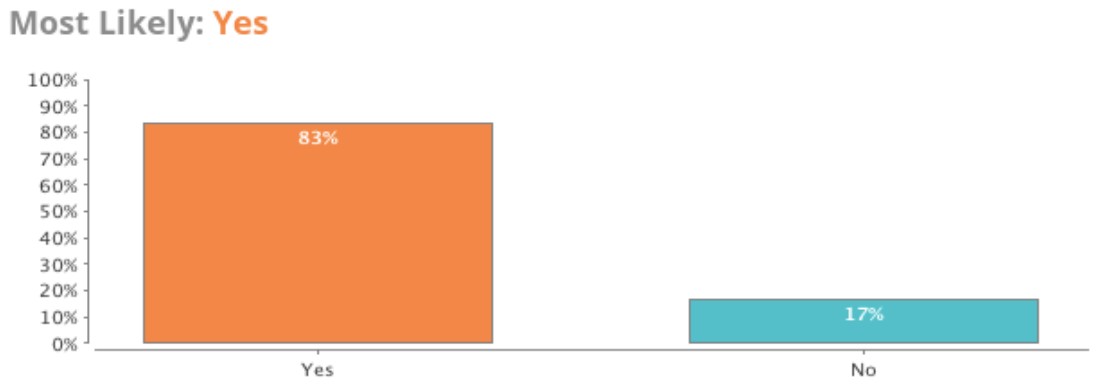
Şekil 5. Lojistik Regresyon Algoritması Yükseliş Grafiği

2. Karar Ağacı (Decision Tree)

Karar ağacı metodolojisi, çoklu değişkenlere dayalı sınıflandırma sistemleri oluşturmak veya bir hedef değişken için tahmin algoritmaları geliştirmek için yaygın olarak kullanılan bir veri madenciliği yöntemidir (Url-10). Karar ağacı oluşturma, çeşitli sınıflardaki (en az iki sınıf) verileri sınıflandırmak için popüler bir tekniktir (Liu, vd.).

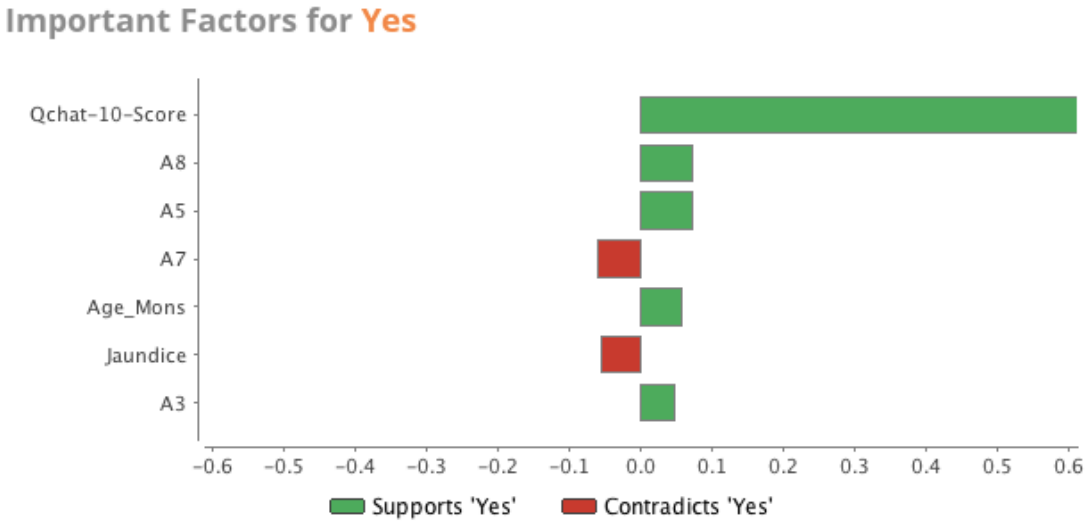
Karar ağacı modellerinin, makul doğruluk elde ettikleri ve hesaplamaları diğer modellere göre daha ucuz olduğu için veri madenciliği alanında en yararlı modeller olduğu görülmüştür (Du ve Zhan, 2002).

Simulasyonda ortaya çıkan gruplama sonucu aşağıda şekil 6'daki gibidir.



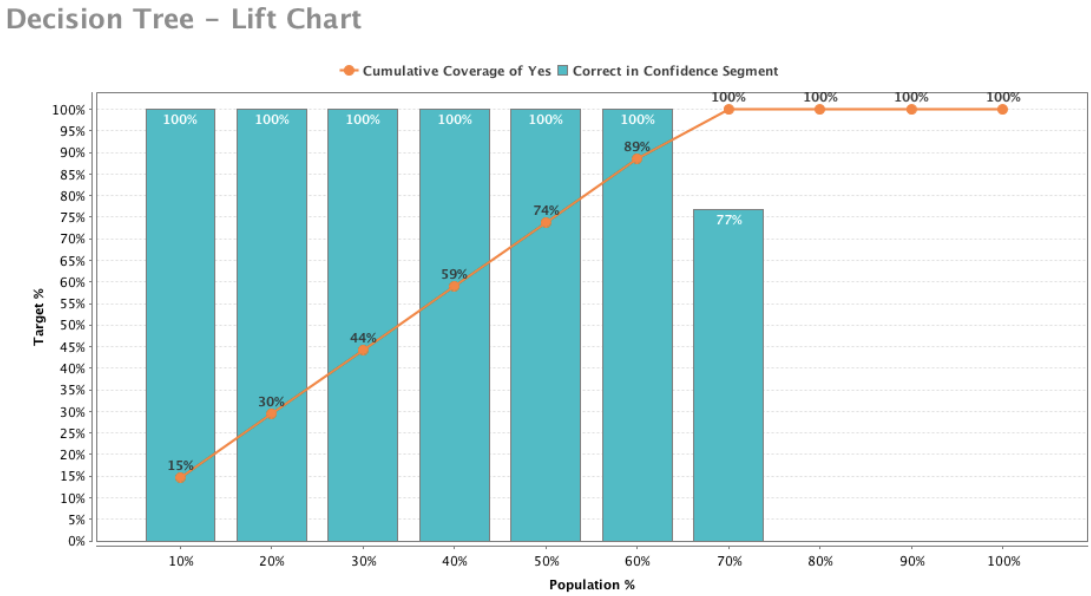
Şekil 6. Karar Ağaçları için Simülasyon Sonucu

OSB Tanısı “Evet” için önemli olan faktörler Şekil 7’deki gibidir.



Şekil 7. OSB Tanısı “Evet” için Önemli Faktörler(Karar Ağaçları)

Genelleştirilmiş doğrusal model algoritması için yükseliş grafiği Şekil 8’deki gibidir.



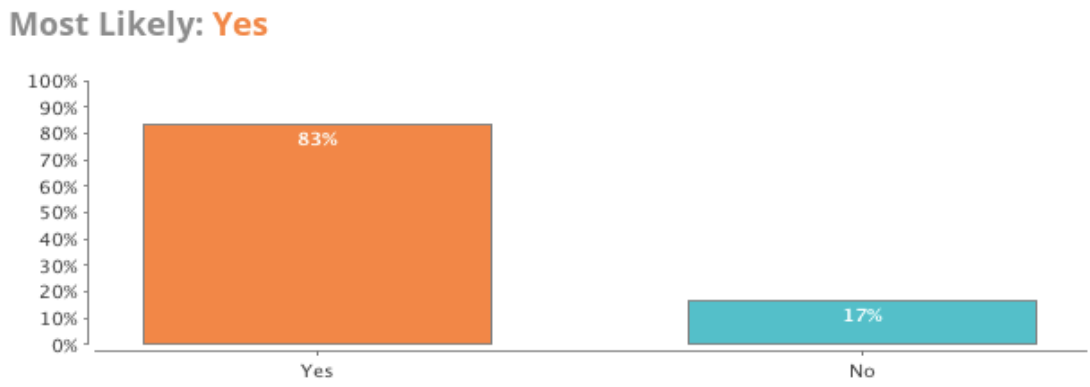
Şekil 8. Karar Ağaçları Algoritması Yükseliş Grafiği

3. Naif Bayes (NB)

Naif Bayes, özelliklerin verilen sınıftan bağımsız olduğunu varsayarak öğrenmeyi büyük ölçüde basitleştiren bir algoritmadır (Url-11). Sadeliğine rağmen iyi performans sergileyebilmesi makine öğrenimi araştırmacılarını şaşırtmıştır (Frank, vd.). Naif

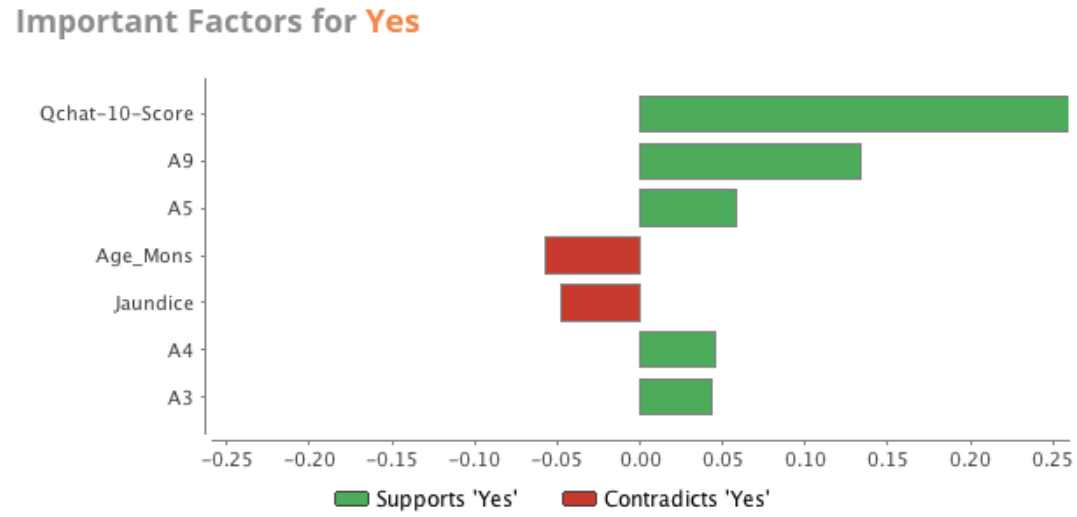
Bayes modelleri, basitlikleri, verimlilikleri ve doğrulukları nedeniyle sınıflandırma ve kümeleme için oldukça popüler hale gelmiştir (Lowd ve Domingos).

Naif Bayes için simülasyon gruplaması sonucu Şekil 9'da gösterilmiştir.



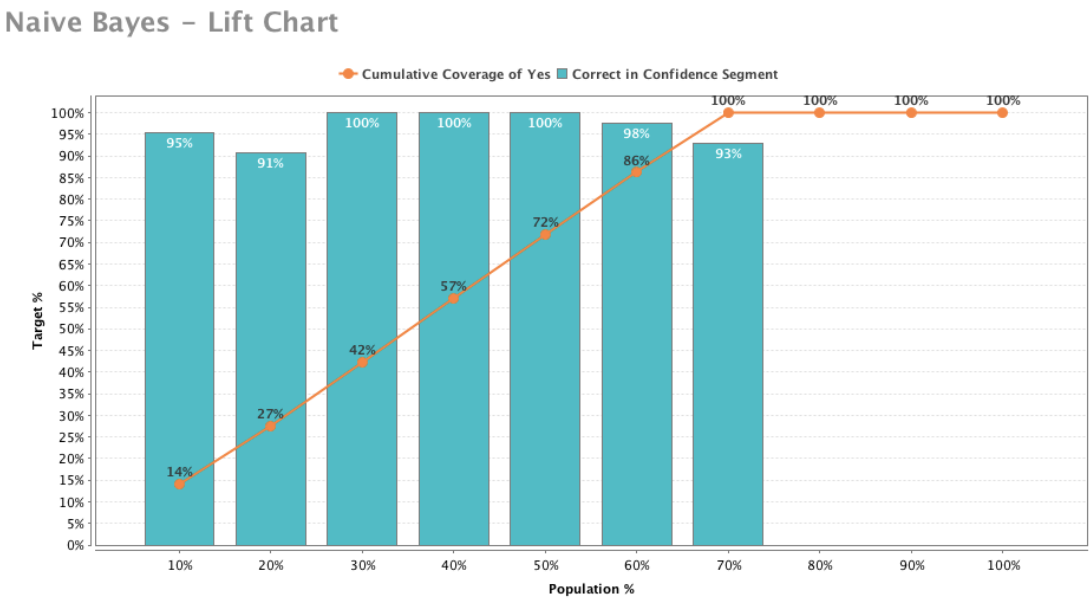
Şekil 9. Naif Bayes için Simülasyon Sonucu

OSB Tanısı "Evet" için önemli olan faktörler Şekil 10'daki gibidir.



Şekil 10. OSB Tanısı "Evet" için Önemli Faktörler(NB)

Bu algoritma için yükseliş grafiği ise Şekil 11'deki gibidir.



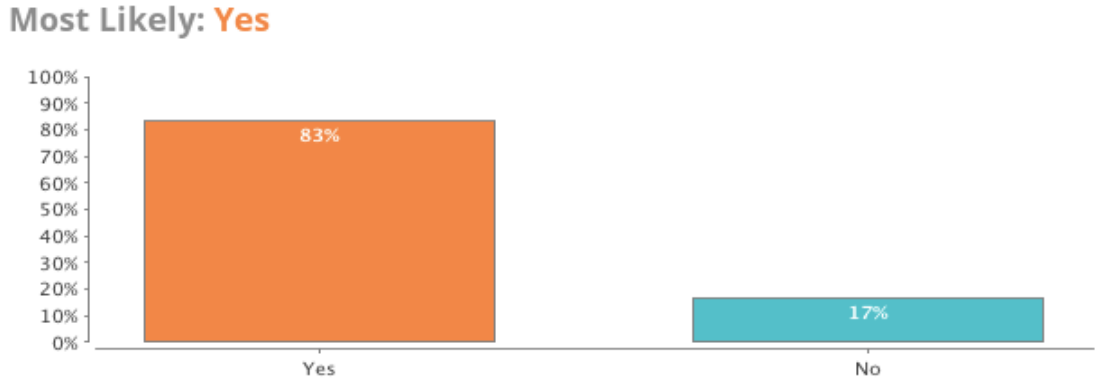
Şekil 11. Naif Bayes Algoritması Yükseliş Grafiği

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

Destek Vektör Makinesi (SVM), nesnelere etiket atayarak öğrenen bir bilgisayar algoritmasıdır (Url-12). Destek vektör makinesi (SVM), Vapnik vd. tarafından önerilen yeni bir evrensel öğrenme makinesidir ve hem regresyon hem de örüntü tanımaya uygulanır (Zhang vd, 2004). Güçlü teorik temellere ve mükemmel ampirik başarılara sahip olmakla birlikte, el yazısı rakam tanıma, nesne tanıma ve metin sınıflandırması gibi görevlere uygulanmıştır (Tong ve Koller, 2001)

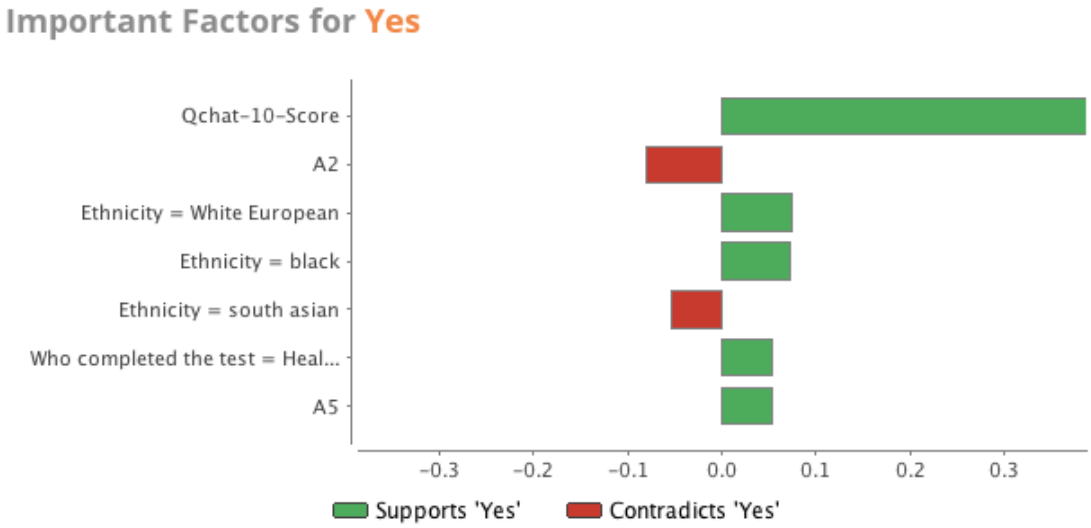
Bu algoritmanın optimizasyon yöntemi, iyi çalışılmış ve anlaşılmış bir matematiksel programlama tekniği olan ikinci dereceden programlamadır (Sebald, 2000).

Simülasyon gruplaması sonucu Şekil 12'de gösterilmiştir.



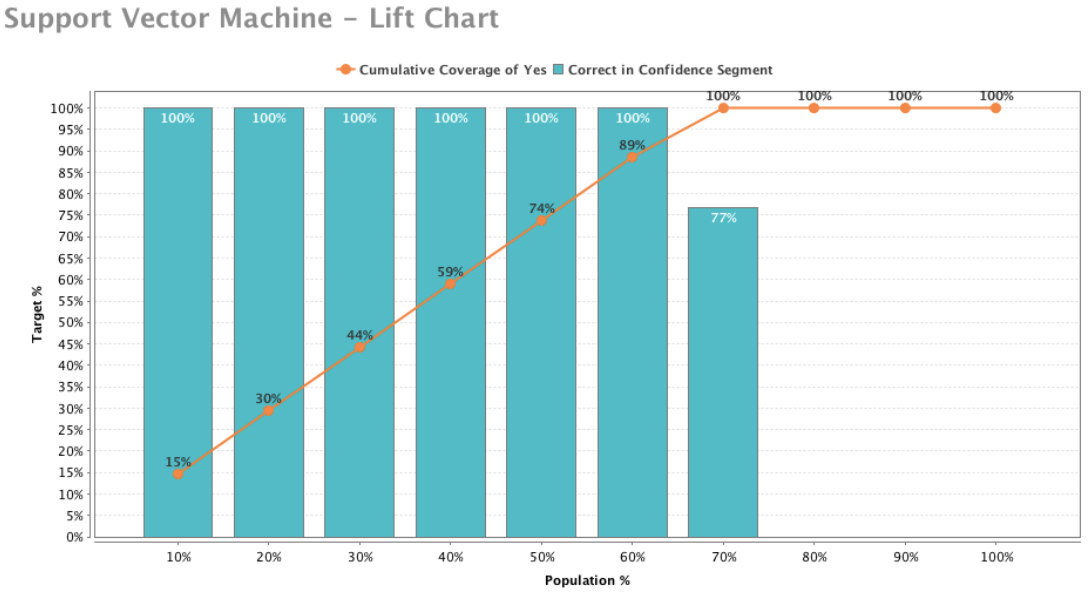
Şekil 12. Destek Vektör Makinesi için Simülasyon Sonucu

OSB Tanısı “Evet” için önemli olan faktörler Şekil 13’teki gibidir.



Şekil 13. OSB Tanısı Evet için Önemli Faktörler (SVM)

Bu algoritma için yükseliş grafiği Şekil 14’deki gibidir.



Şekil 14. Destek Vektör Makinesi Yükseliş Grafiği

Yüksek boyutlu alanlarda etkili olması ve karar verme işleminde eğitim noktası kullanması avantajlarından bazılarıdır. Doğrudan olasılık tahmini sağlamaması ise bir dezavantajdır.

5. Rastgele Orman (Random Forest)

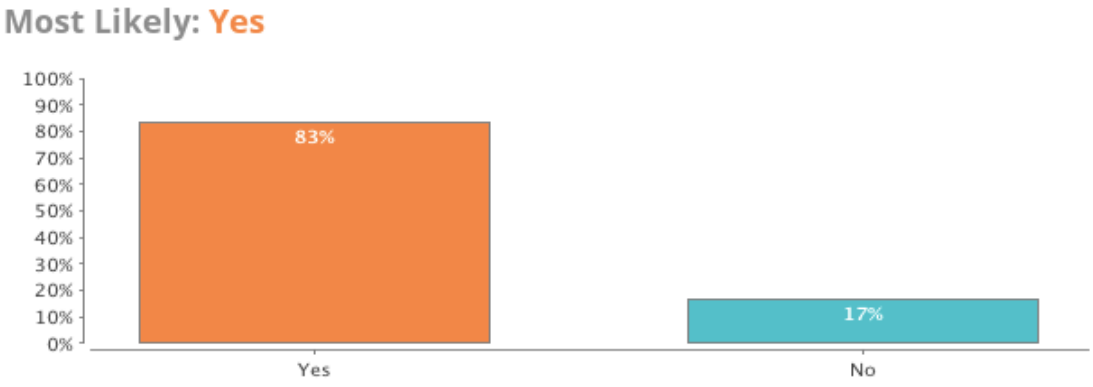
İlk olarak Breiman (2001) tarafından tanıtılan rastgele orman algoritması, günümüzde yanıt değişkeni ile ilişkilerinin şekli hakkında herhangi bir ön varsayımda

bulunmadan, çeşitli yordayıcı değişkenlere dayalı tahmin kuralları oluşturmak için standart bir parametrik olmayan sınıflandırma ve regresyon aracına dönüşmüştür (Url-13).

Rastgele orman, tahmin edici değişkenlerin çoğu gürültü olduğunda bile mükemmel performans gösterir, değişkenlerin sayısı gözlem sayısından çok daha fazla olduğunda ve ikiden fazla sınıfı içeren problemlerde kullanılabilir (Uriarte ve Andrez, 2006). Büyük veri kümelerinde hızlı bir şekilde çalışabilen, hesaplama açısından verimli bir tekniktir ve son zamanlarda birçok araştırma projesinde kullanılmıştır (Baranauskas, vd. 2012).

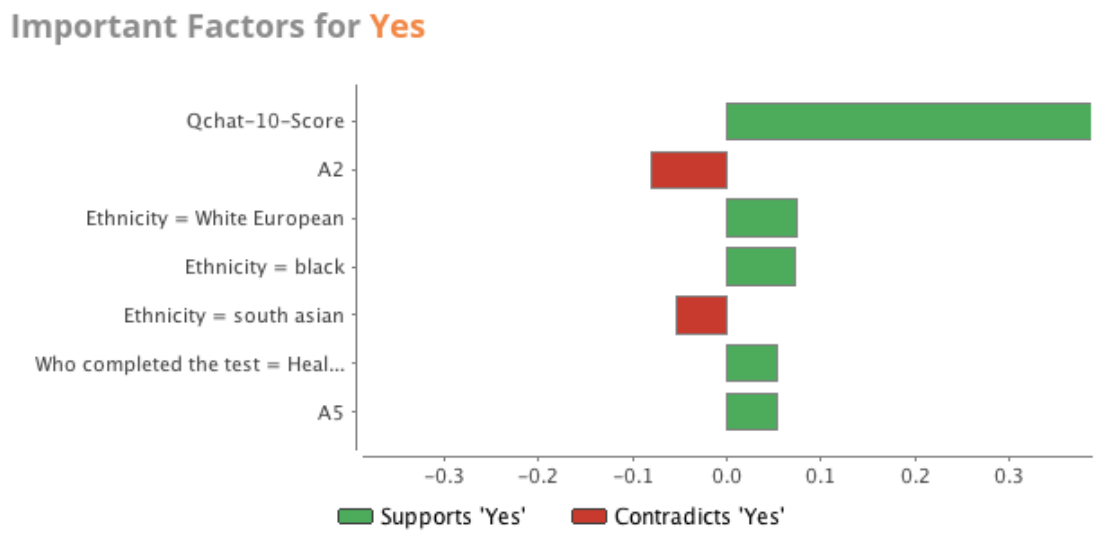
Bu yöntem, bazı rastgele seçimlerden oluşan bir karar ağaçları koleksiyonuna dayanır ve diğer birçok sınıflandırma ve regresyon yöntemi gibi, uygulamaya uygun eğitim örnekleri temelinde bir rastgele orman oluşturulur. (Albert, vd. 2007).

Simülasyon sonucu Şekil 15'te gösterilmiştir.



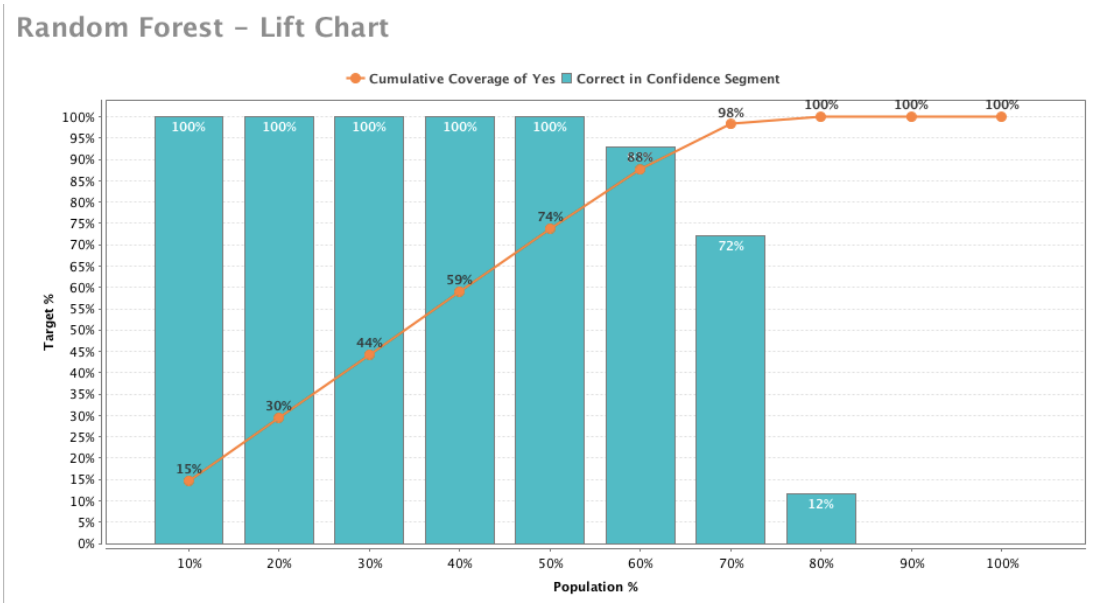
Şekil 15. Rastgele Orman için Simülasyon Sonucu

OSB Tanısı "Evet" için önemli olan faktörler Şekil 16'daki gibidir.



Şekil 16. OSB Tanısı Evet için Önemli Faktörler (RF)

Bu algoritma için yükseliş grafiği Şekil 17'deki gibidir.



Şekil 17. Rastgele Orman (Random Forest) Yükseliş Grafiği

IV. BULGULAR

A. Lojistik Regresyon (Logistic Regression – LR)

Lojistik regresyon modelinde doğruluk değerinin 99.0%, hata.oranının ise 1,0% olduğu aşağıda Çizelge 1'de gösterilmiştir. Belirlilik ve Hassasiyet kriterleri ise 100.0% olarak görülmektedir.

Lojistik Regresyon performans sonucu Çizelge 2'deki gibidir.

Çizelge 2 – Lojistik Regresyon performans sonucu

Kriter	Değer	Standart Sapma
Doğruluk	99.0%	±0.9%
Sınıflandırma Hatası	1.0%	±0.9%
Eğrinin Altındaki Alan	99.7%	±0.7%
Hassasiyet	100.0%	±0.0%
Geri Çağırma	98.6%	±1.3%
Testin Doğruluğu	99.3%	±0.7%
Gerçek Pozitifler Oranı	98.6%	±1.3%
Belirlilik	100.0%	±0.0%

B. Karar Ağaçları (Decision Trees – DT)

Karar Ağaçları modelinde yüzde yüz doğruluk saptanmıştır. Dolayısıyla hata oranı 0'dır. Hassasiyet, Belirlilik, Testin Doğruluğu kriterleri de yüzde yüzdür.

Karar Ağaçları performans sonucu Çizelge 3'teki gibidir.

Çizelge 3 - Karar Ağaçları performans sonucu

Kriter	Değer	Standart Sapma
Doğruluk	100.0%	±0.0%
Sınıflandırma Hatası	0.0%	±0.0%
Eğrinin Altındaki Alan	100.0%	±0.0%
Hassasiyet	100.0%	±0.0%
Geri Çağırma	100.0%	±0.0%
Testin Doğruluğu	100.0%	±0.0%
Gerçek Pozitifler Oranı	100.0%	±0.0%
Belirlilik	100.0%	±0.0%

C. Naif Bayes (Naive Bayes – NB)

Naif Bayes modelinde 3.0% oranında hata bulunmaktadır. Doğruluk ise 97.0%'dir.

Naif Bayes performans sonucu Çizelge 4'teki gibidir.

Çizelge 4 – Naif Bayes performans sonucu

Kriter	Değer	Standart Sapma
Doğruluk	97.0%	±0.7%
Sınıflandırma Hatası	3.0%	±0.7%
Eğrinin Altındaki Alan	99.8%	±0.2%
Hassasiyet	98.1%	±1.1%
Geri Çağırma	97.6%	±1.7%
Testin Doğruluğu	97.9%	±0.6%
Gerçek Pozitifler Oranı	97.6%	±1.7%
Belirlilik	95.1%	±2.9%

D. Destek Vektör Makinesi (Support Vector Machine – SVM)

Destek Vektör Makinesi modelinin performansının oldukça başarılı olduğu aşağıda Çizelge 4'te görülmektedir. Hiç hata bulunmamaktadır ve diğer kriterler 100.0%dür.

Destek Vektör Makinesi performans sonucu Çizelge 5'teki gibidir.

Çizelge 5 – Destek Vektör Makinesi performans sonucu

Kriter	Değer	Standart Sapma
Doğruluk	100.0%	±0.0%
Sınıflandırma Hatası	0.0%	±0.0%
Eğrinin Altındaki Alan	100.0%	±0.0%
Hassasiyet	100.0%	±0.0%
Geri Çağırma	100.0%	±0.0%
Testin Doğruluğu	100.0%	±0.0%
Gerçek Pozitifler Oranı	100.0%	±0.0%
Belirlilik	100.0%	±0.0%

E. Rastgele Orman (Random Forest – RF)

Rastgele Orman modelinde 3.7% oranında hata bulunup, doğruluk 96.3% olarak görülmektedir.

Rastgele Orman performans sonucu Çizelge 6'daki gibidir.

Çizelge 6 – Rastgele Orman performans sonucu

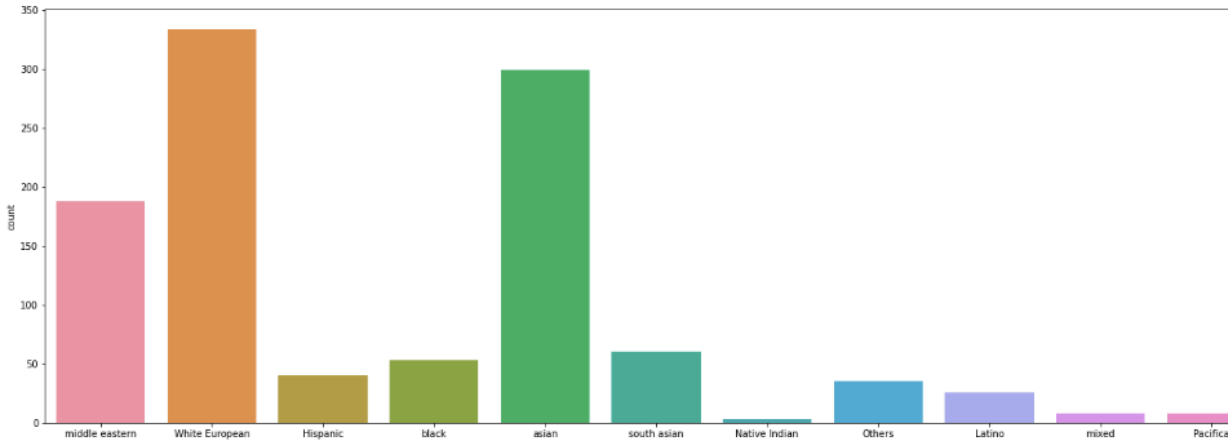
Kriter	Değer	Standart Sapma
Doğruluk	96.3%	±0.7%
Sınıflandırma Hatası	3.7%	±0.7%
Eğrinin Altındaki Alan	99.0%	±0.5%
Hassasiyet	98.6%	±2.1%
Geri Çağırma	96.2%	±2.8%
Testin Doğruluğu	97.4%	±0.7%
Gerçek Pozitifler Oranı	96.2%	±2.8%
Belirlilik	96.6%	±4.8%

.

V. SONUÇ VE ÖNERİLER

Bu bölümde, Jupiter Notebook'ta Kaggle'den alınan veri seti kullanılarak eğitim ve test için ayrılarak alınan sonuç ve RapidMiner programında yapılan Doğruluk, Sınıflandırma Hatası, Eğrinin Altında Kalan Alan, Hassasiyet, Geri Çağırma, Testin Doğruluğu, Gerçek Pozitifler Oranı ve Belirlilik analizlerinin sonuçları aktarılmıştır. Yapılan analizler grafikler ile gösterilmiştir.

Matplotlib fonksiyonu kullanılarak OSB tanısı konan çocukların "Irk" özelliğine göre grafiği aşağıda şekil 18'de gösterilmiştir.



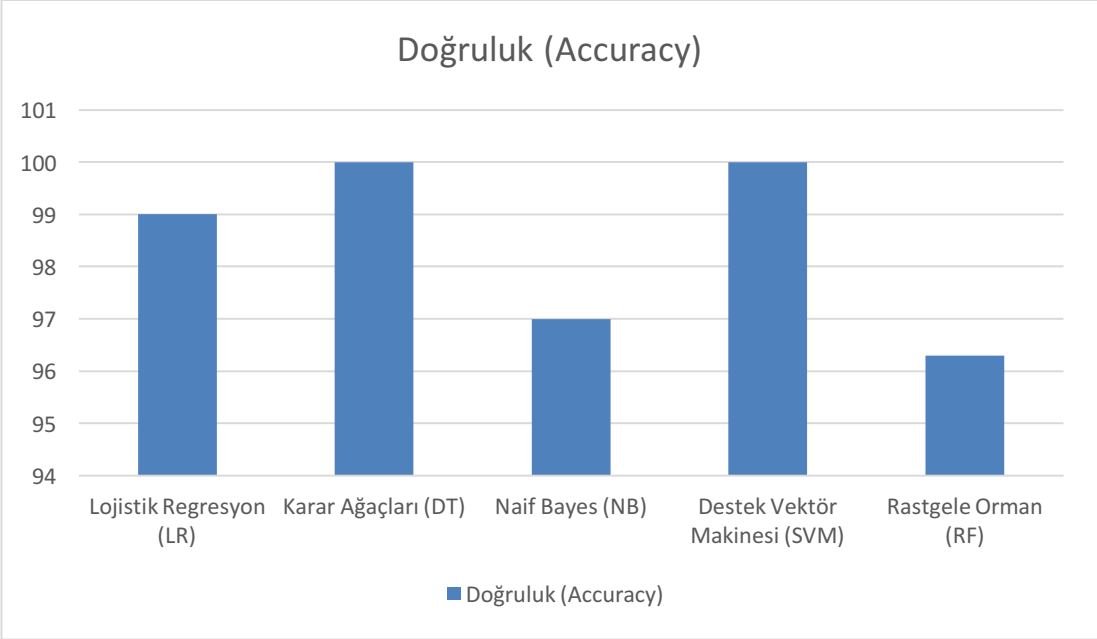
Şekil 18. OSB Tanısı Konan Çocukların Irklara Göre Dağılımı

Aşağıda, bu çalışmada kullanılan algoritmaların doğruluk sonuçları şekil 19'da gösterilmiştir. Lojistik regresyon algoritması %100 doğruluk ile en iyi sonucu elde etmiştir. En düşük doğruluk oranı ise %63 ile Rastgele Orman algoritmasına aittir.

```
Logistic Regression: 1.0
Decision Tree      : 0.933649289099526
Naive Bayes        : 0.952606635071090
SVM                : 0.8530805687203792
Random Forest      : 0.6398104265402843
```

Şekil 19. Kullanılan Algoritmaların Doğruluk (Accuracy) Sonuçları

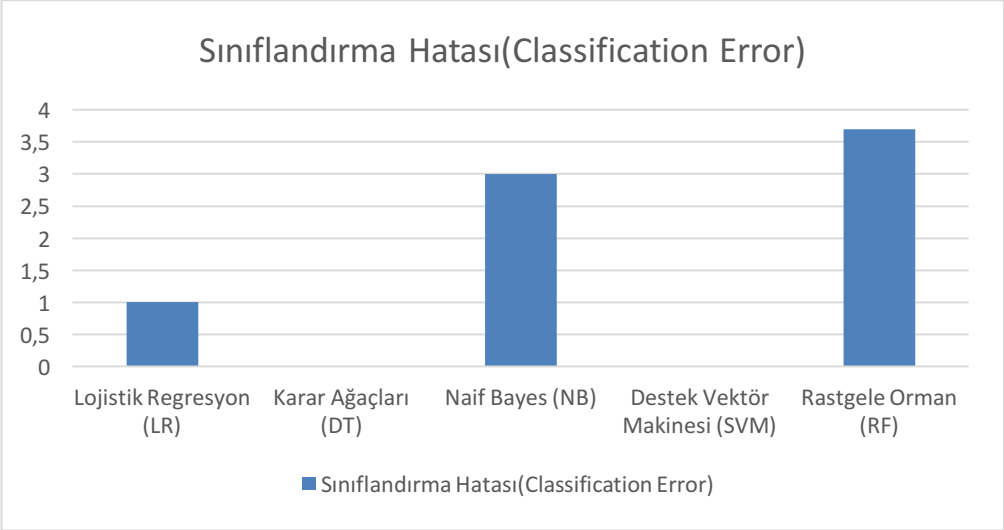
A. Doğruluk (Accuracy) Analizi



Şekil 20. Doğruluğa Dayalı Performans Analizi

Doğruluğa Dayalı Performans Analizi'nde Karar Ağaçları (DT) ve Destek Vektör Makinesi (SVM) 100,0% doğruluk oranı ile en başarılı performansı göstermişlerdir. Ardından 99,0% ile Lojistik Regresyon gelmektedir. Bu analizde Naif Bayes (NB) 97,0% doğruluk oranı gösterirken Rastgele Orman (RF) ise 96,3% doğruluk oranı göstermiştir.

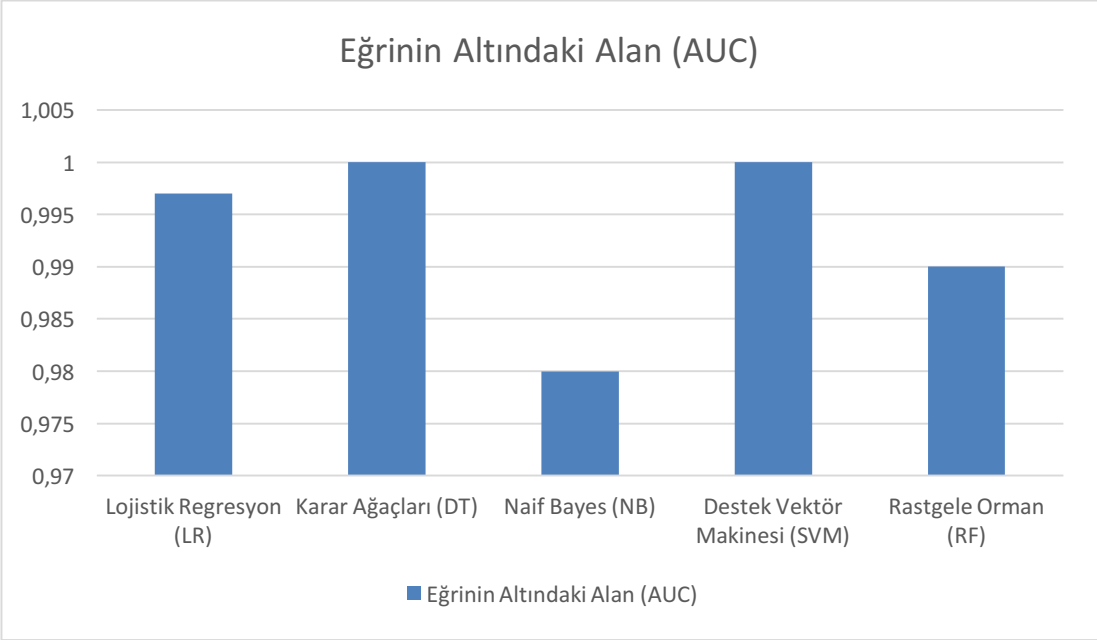
B. Sınıflandırma Hatası (Classification Error) Analizi



Şekil 21. Sınıflandırma Hatasına Dayalı Performans Analizi

Sınıflandırma Hatasına Dayalı Performans Analizi'nde Karar Ağaçları (DT) ve Destek Vektör Makinesi (SVM) 0,0% ile en başarılı performansı göstermiştir yani en az hata oranına sahiptir. En fazla hata oranına sahip olan algoritma ise 3,7% ile Rastgle Orman (RF) algoritmasıdır. Naif Bayes 3,0%, Lojistik Regresyon ise 1,0% hata oranına sahiptir.

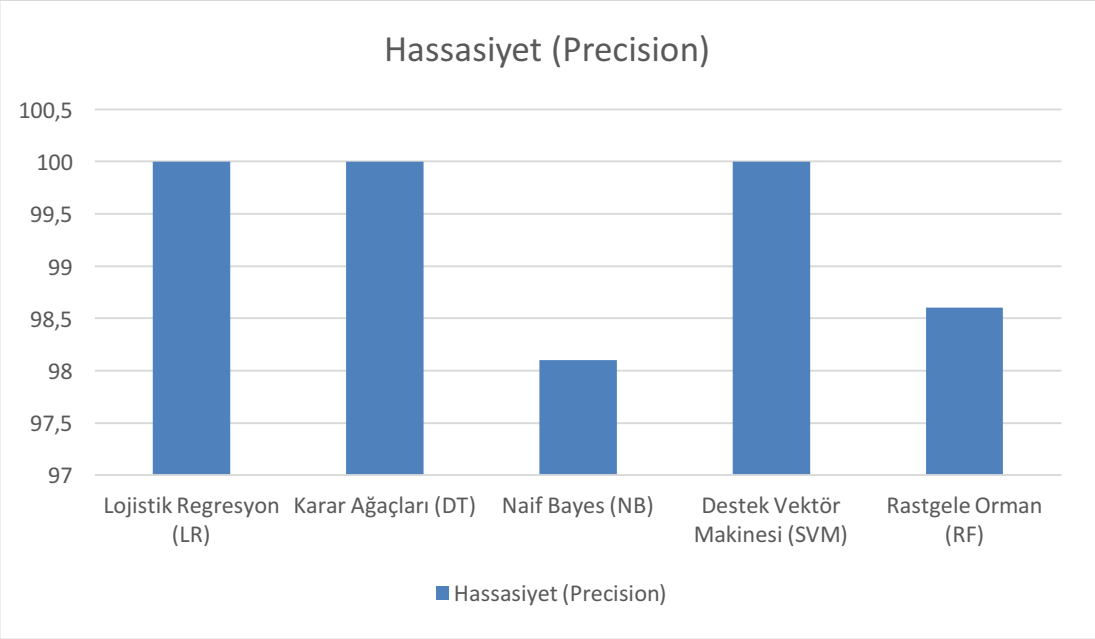
C. Eğrinin Altındaki Alan (AUC) Analizi



Şekil 22. Eğrinin Altındaki Alana Dayalı Performans Analizi

Eğrinin Altındaki Alana Dayalı Performans Analizi'nde Karar Ağaçları (DT) ve Destek Vektör Makinesi (SVM) en başarılı performansı göstermişlerdir.

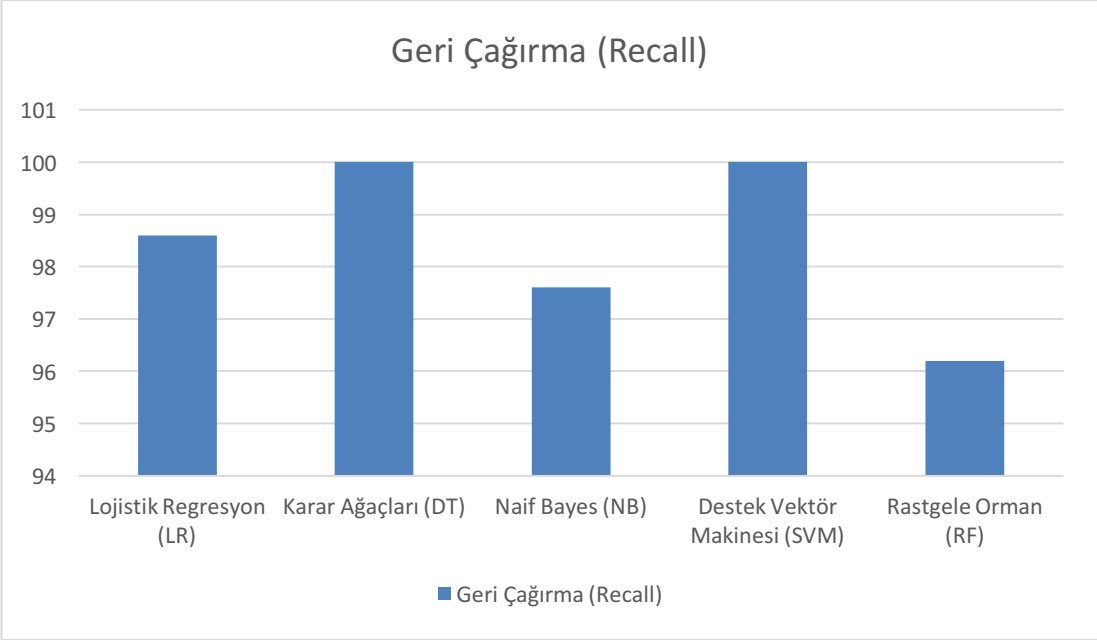
D. Hassasiyet (Precision) Analizi



Şekil 23. Hassasiyete Dayalı Performans Analizi

Hassasiyete Dayalı Performans Analizi'nde en başarılı algoritmalar 100,0% oran ile Lojistik Regresyon (LR), Karar Ağaçları (DT) ve Destek Vektör Makinesi (SVM) olmuştur.

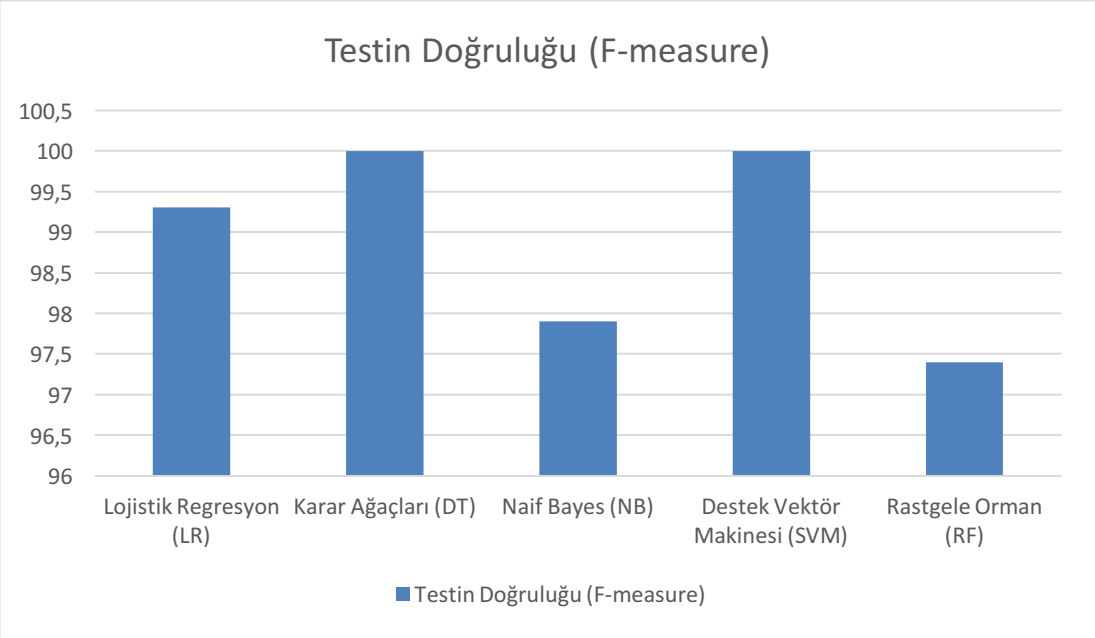
E. Geri Çağırma (Recall) Analizi



Şekil 24. Geri Çağırmaya Dayalı Performans Analizi

Geri Çağırmaya Dayalı Performans Analizi'nde Karar Ağaçları (DT) ve Destek Vektör Makinesi (SVM) en başarılı performansı göstermiştir.

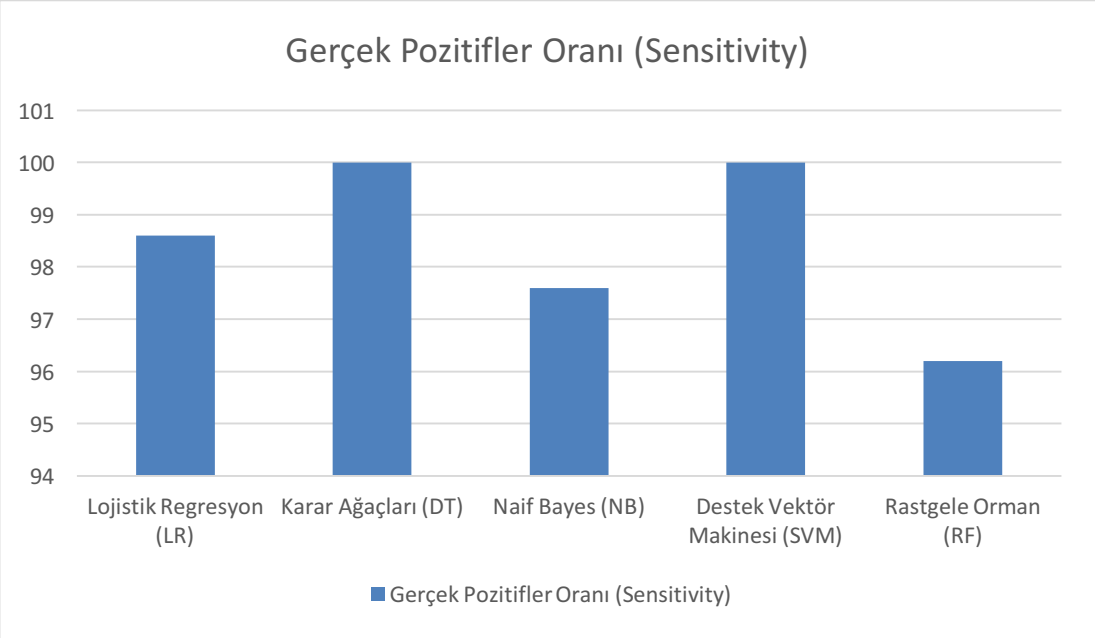
F. Testin Doğruluğu (F-measure) Analizi



Şekil 25. Testin Doğruluğuna Dayalı Performans Analizi

Testin Doğruluğuna Dayalı Performans Analizi'nde Karar Ağaçları (DT) ve Destek Vektör Makinesi (SVM) en iyi performansı göstermişlerdir.

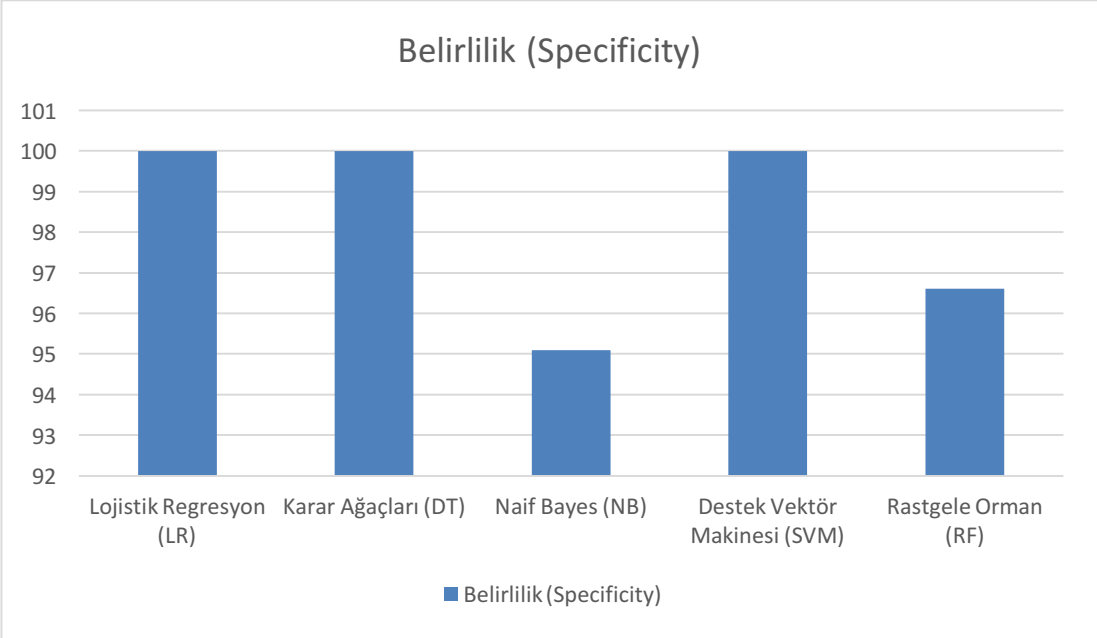
G. Gerçek Pozitifler Oranı (Sensitivity) Analizi



Şekil 26. Gerçek Pozitifler Oranına Dayalı Performans Analizi

Gerçek Pozitifler Oranına Dayalı Performans Analizi'nde Karar Ağaçları (DT) ve Destek Vektör Makinesi (SVM) en iyi performansı göstermişlerdir. Ve 96,2% oranla Rastgele Orman (RF) en az orana sahiptir.

H. Belirlilik (Specificity) Analizi



Şekil 27. Belirliliğe Dayalı Performans Analizi

Belirliliğe Dayalı Performans Analizi'nde 3 algoritma 100,0% performans göstermiştir. Bu algoritmalar Lojistik Regresyon (LR), Karar Ağaçları (DT) ve Destek Vektör Makinesi (SVM)'dir.

I. Tartışma ve Öneriler

Klasik istatistiklerden veya makine öğrenmesinden elde edilen çok sayıda algoritma, veri madenciliği projelerinin yürütülmesini mümkün kılmaktadır. Doğruluk değerinin yüksek çıkması için algoritma seçimi da oldukça önemlidir. Çalışmada kullanılan kod ve modüller ekte paylaşılmış, açıklaması yapılmıştır. Seçilen metotlar, yapılan analizler açıklanarak anlatılmıştır. Q-Chat-10 testinin 10 sorusu, yaş, ırk, cinsiyet, sarılık hastalığı geçirip geçirmediği, ailede otizmli birey olup olmadığı, otizm tanısı, Q-Chat-10 testinin rakam olarak sonucu ve bu testi kimin tamamladığı bilgilerinin olduğu veri kümesi üzerinde 5 adet algoritma ile çalışılmıştır. Doğruluk değerinin artması adına veri setinde gereksiz özelliklerin elenmesi için ön işleme de yapılmıştır Sonuç olarak en iyi performansı sağlayan algoritma Lojistik Regresyon algoritması olmuştur. Daha sonra onu sırasıyla Naif Bayes, Karar Ağaçları, Destek Vektör

Makinesi ve Rastgele Orman takip etmiştir. Veri Madenciliği'nde yapılan çalışmalarda özellik seçimi, İngilizce adıyle feature selection oldukça büyük önem taşımaktadır. Bu çalışmada algoritmaların sağlıklı bir şekilde karşılaştırılmasının yapılabilmesi için "Osb Tanısı" özelliği seçilmiştir. Bu özellik Q-Chat-10 testindeki rakamsal sonucun 3'ten küçük ve eşitse "Hayır", sonuç 3'ten büyükse "Evet" olduğunu yansıtmaktadır. Dolayısıyla yeni gelen veri kümelerinde uygulanabilirlik açısından bu özellik bizim için ayırt edicidir. RapidMiner programında veri kümesini işleme alıp "Osb Tanısı" özelliğini seçerek tahmin yaptığımızda yüzde yüz doğruluğu Destek Vektör Makinesi algoritmasının verdiği görülmüştür. Hata oranının ise en fazla Rastgele Orman algoritmasında olduğu görülmüştür.

Veri madenciliği kullanılarak yapılan çalışmalarda kullanılan sınıfların ve özellik seçiminin önemi aşikardır. Bu yüzden aradaki farkın görülebilmesi adına kullanılan veri setinde bulunan ve korelasyonu düşük olup, daha çok text içeren, stabilitesi düşük olan değerler çıkartılıp sonuçlar aşağıdaki gibi karşılaştırılmıştır.

Çizelge 7 – Doğruluk "accuracy" değerinin algoritma bazında karşılaştırılması

Model	Doğruluk	Standart Sapma	Kazanım	Toplam Süre	Eğitim Süresi	Sonuç Süresi
Naif Bayes	1,0	0,0	172,0	574505,0	93,9	294,5
Lojistik Regresyon	1,0	0,0	172,0	737492,0	131,9	85,5
Karar Ağaçları	1,0	0,0	172,0	732061,0	82,5	109,3
Rastgele Orman	1,0	0,0	172,0	762874,0	76,9	237,5
Destek Vektör Makinesi	1,0	0,0	172,0	759253,0	258,1	370,5

Çizelge 8 – Sınıflandırma hatasının algoritma bazında karşılaştırılması

Model	Doğruluk	Standart Sapma	Kazanım	Toplam Süre	Eğitim Süresi	Sonuç Süresi
Naif Bayes	0,0	0,0	172,0	574505,0	93,9	294,5
Lojistik Regresyon	0,0	0,0	172,0	737492,0	131,9	85,5
Karar Ağaçları	0,0	0,0	172,0	732061,0	82,5	109,3
Rastgele Orman	0,0	0,0	172,0	762874,0	76,9	237,5
Destek Vektör Makinesi	0,0	0,0	172,0	759253,0	258,1	370,5

Çizelge 9 – Eğrinin altında kalan alanın algoritma bazında karşılaştırılması

Model	Doğruluk	Standart Sapma	Kazanım	Toplam Süre	Eğitim Süresi	Sonuç Süresi
Naif Bayes	1,0	0,0	172,0	574505,0	93,9	294,5
Lojistik Regresyon	1,0	0,0	172,0	737492,0	131,9	85,5
Karar Ağaçları	1,0	0,0	172,0	732061,0	82,5	109,3
Rastgele Orman	1,0	0,0	172,0	762874,0	76,9	237,5

Destek Vektör Makinesi	1,0	0,0	172,0	759253,0	258,1	370,5

Çizelge 10 – Hassasiyet değerinin algoritma bazında karşılaştırılması

Model	Doğruluk	Standart Sapma	Kazanım	Toplam Süre	Eğitim Süresi	Sonuç Süresi
Naif Bayes	1,0	0,0	172,0	574505,0	93,9	294,5
Lojistik Regresyon	1,0	0,0	172,0	737492,0	131,9	85,5
Karar Ağaçları	1,0	0,0	172,0	732061,0	82,5	109,3
Rastgele Orman	1,0	0,0	172,0	762874,0	76,9	237,5
Destek Vektör Makinesi	1,0	0,0	172,0	759253,0	258,1	370,5

Çizelge 11 – Geri çağırma değerinin algoritma bazında karşılaştırılması

Model	Doğruluk	Standart Sapma	Kazanım	Toplam Süre	Eğitim Süresi	Sonuç Süresi
Naif Bayes	0,0	0,0	172,0	574505,0	93,9	294,5
Lojistik Regresyon	0,0	0,0	172,0	737492,0	131,9	85,5
Karar Ağaçları	0,0	0,0	172,0	732061,0	82,5	109,3
Rastgele Orman	0,0	0,0	172,0	762874,0	76,9	237,5
Destek Vektör Makinesi	0,0	0,0	172,0	759253,0	258,1	370,5

Çizelge 12 – Testin doğruluğunun algoritma bazında karşılaştırılması

Model	Doğruluk	Standart Sapma	Kazanım	Toplam Süre	Eğitim Süresi	Sonuç Süresi
Naif Bayes	1,0	0,0	172,0	574505,0	93,9	294,5
Lojistik Regresyon	1,0	0,0	172,0	737492,0	131,9	85,5
Karar Ağaçları	1,0	0,0	172,0	732061,0	82,5	109,3
Rastgele Orman	1,0	0,0	172,0	762874,0	76,9	237,5
Destek Vektör Makinesi	1,0	0,0	172,0	759253,0	258,1	370,5

Çizelge 13 – Gerçek pozitifler oranının algoritma bazında karşılaştırılması

Model	Doğruluk	Standart Sapma	Kazanım	Toplam Süre	Eğitim Süresi	Sonuç Süresi
Naif Bayes	1,0	0,0	172,0	574505,0	93,9	294,5
Lojistik Regresyon	1,0	0,0	172,0	737492,0	131,9	85,5
Karar Ağaçları	1,0	0,0	172,0	732061,0	82,5	109,3
Rastgele Orman	1,0	0,0	172,0	762874,0	76,9	237,5
Destek Vektör Makinesi	1,0	0,0	172,0	759253,0	258,1	370,5

Çizelge 14 – Belirlilik değerinin algoritma bazında karşılaştırılması

Model	Doğruluk	Standart Sapma	Kazanım	Toplam Süre	Eğitim Süresi	Sonuç Süresi
Naif Bayes	1,0	0,0	172,0	574505,0	93,9	294,5
Lojistik Regresyon	1,0	0,0	172,0	737492,0	131,9	85,5

Karar Ağaçları	1,0	0,0	172,0	732061,0	82,5	109,3
Rastgele Orman	1,0	0,0	172,0	762874,0	76,9	237,5
Destek Vektör Makinesi	1,0	0,0	172,0	759253,0	258,1	370,5

Sonuç olarak, veri setinde bulunan ve korelasyonu düşük olup, daha çok text içeren, stabilitesi düşük olması sebebiyle çıkartılan değerler, sonucu çok fazla etkilememiştir. Buradan da anlaşılacağı üzere, kullandığımız veri setinde, otizm tahmini yapılırken etnik köken, cinsiyet gibi özelliklerin çok büyük bir etkisi bulunmadığı ortaya çıkmıştır.

Bu da aynı zamanda, veriler kullanırken kullanılan sınıfların ve özelliklerin seçiminin ne kadar önemli olduğunu ortaya çıkarmıştır. Bu şekilde gelecek çalışmalarda, farklı veri setleri ve farklı sınıflar ile özellikler denenerek otizmin tespiti için makine öğrenmesi kullanarak , asıl hedef olan otizmin doğru bir şekilde tespitine yakınlaşılabileceğini göstermektedir.

Günümüzde yapılan çalışmalar, otizmli çocuğa sahip ailelerin ikinci ve sonraki çocuklarının da otizmli olma riskinin yüzde 20 olduğunu göstermektedir. Bu da demek oluyor ki genetik faktörün otizm belirlenmesindeki önemi yadsınamaz ancak tek başına yeterli değildir. Otizm tanısı ile ilgili yapılmış olan birçok çalışma mevcuttur. Literatür taraması bölümünde bahsedilen, veri madenciliği yöntemleri kullanılarak (örneğin; karar ağaçları) yapılan çalışmalarda doğruluk (accuracy) değerinin yüzde yüzün altında olduğu görülmüştür. Bu tez çalışması sayesinde otizm ile ilgili yapılacak çalışmalarda hangi algoritmanın daha yüksek doğruluk sağladığı bilinip, çalışmalar bu yönde yapılabilir. Ve hangi algoritmanın en az doğruluk sağladığı bilinip bu tecrübe ile hareket edilebilir.

VI. KAYNAKÇA

MAKALELER

ABDULLAH A. A., vd. (2019). "Otizm Spektrum Bozukluğunun (ASD) Sınıflandırılmasına Yönelik Makine Öğrenimi Algoritmalarının Değerlendirilmesi", **Journal of Physics,** doi:10.1088/1742-6596/1372/1/012052

ALBERT J., vd. (2018). "Implementation of Random Forest Method for the Imaging Atmospheric Cherenkov Telescope Magic.", **Science Direct,** https://arxiv.org/pdf/0709.3719.pdf

ALLISON C., vd. (2008). "The Quantitative Checklist for Autism in Toddlers (Q-CHAT): Psychometric Properties." **Journal of Autism and Developmental Disorders**, 38,1414-1425.

ALWIDIAN J., vd. (2020). "Predicting Autism Spectrum Disorder using Machine Learning Technique.", **International Journal of Recent Technology and Engineering (IJRTE),** https://www.ijrte.org/wp-content/uploads/papers/v8i5/E6016018520.pdf

AYHAN S. ve ERDOĞMUŞ Ş. (2014). "Destek Vektör Makineleriyle Sınıflandırma Problemlerinin Çözümü İçin Çekirdek Fonksiyonu Seçimi", **Eskişehir Osman Gazi Üniversitesi İibf Dergisi,** 9 (1), 175-201.

BERMINGHAM M., vd. (2015). "Application of high-dimensional feature selection: evaluation for genomic prediction in man", **Scientific Reports.**

DAYTON M. (1992). "Logistic Regression Analysis." http://www.econ.upf.edu/~satorra/dades/M2012LogisticRegressionDayton.pdf

DEMİRHAN A. (2018). "Otizm Spektrum Bozukluk Vakalarını Belirlemede Makine Öğrenme Yöntemlerinin Performansı" DOI:10.22531/muglajsci.422546

DEMISSE G., vd. (2017). "Data Mining Attribute Selection Approach for Drought Modeling: A Case Study for Greater Horn of Africa.", **International Journal of Data Mining & Knowledge Management Process**. 7. 01-16. 10.5121/ijdkp.2017.7401

DIAZ-URIARTE R. ve ALVAREZ DE ANDRES S. (2006) “Gene selection and classification of microarray data using random forest.” BMC Bioinformatics 7, 3 doi:10.1186/1471-2105-7-3

DU W. ve ZHAN Z. (2002). "A Practical Approach to Solve Secure Multi-Party Computation Problems.", **The ACM Digital Library.**

GÖKER H. vd. (2015). “Erken Çocukluk Döneminde Otizm Teşhisine Yönelik Dinamik Uzman Sistem Tasarımı.”, **Bilişim Teknolojileri Dergisi**, doi: 10.17671/btd.65517

HYDE H., vd. (2019). “Otizm Spektrum Bozukluğu Araştırmasında Denetimli Makine Öğreniminin Uygulamaları: Bir Gözden Geçirme”

LIU B., vd. (2000). “Clustering Through Decision Tree Construction.” In Proceedings of the ACM International Conference on Information and Knowledge Management, **The ACM Digital Library**, doi:10.1145/354756.354775.

LOWD D. ve DOMINGOS P. “Naive Bayes Models for Probability Estimation” http://aiweb.cs.washington.edu/ai/nbe/nbe_icml.pdf

PARIKH M., vd. (2019). “Optimize Edilmiş Makine Öğrenimi Modelleri ve Kişisel Karakteristik Verilerle Otizm Teşhisini Geliştirmek”
doi: 10.3389/fncom.2019.00009

PARK H. (2013). “An Introduction to Logistic Regression: From Basic Concepts to Interpretation with Particular Attention to Nursing Domain” J Korean Acad Nurs Vol.43 No.2, 154-164 J Korean Acad Nurs Vol.43 No.2 doi:10.4040/jkan.2013.43.2.154

SEBALD D., vd. (2000). “Support Vector Machine Techniques for Nonlinear Equalization.”
IEEE Transactions On Signal Processing, vol. 48, no. 11

TONG S. ve KOLLER D. (2001). “Support Vector Machine Active Learning with Applications to Text Classification.”

VARSHINI G. ve CHINNAIYEN R**.** (2020). “Otizm Spektrum Bozukluğunun Tahmini için Optimize Edilmiş Makine Öğrenimi Sınıflandırma Yaklaşımları”

WEI H., vd. (2005). “Forecasting stock market movement direction with support vector machine” doi:10.1016/j.cor.2004.03.016

YAZICI B., vd. (1999). " Veri Madenciliğinde Özellik Seçim Tekniklerinin Bankacılık Verisine Uygulanması Üzerine Araştırma ve Karşılaştırmalı

Uygulama". **Journ. of Jap. Soc. For Artificial Intelligence**, vol. 14, no.5, sayfa 771-780, 1999

ZHANG L., vd. (2004). "Wavelet Support Vector Machine" IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, vol. 34, no.1

KİTAPLAR

HAN J., vd. (2011), **Data Mining: Concepts And Techniques,** Elsevier, 3rd Edition. g

BARANAUSKAS A., vd. (2012). "How Many Trees in a Random Forest?" In: Perner P. (eds) Machine Learning and Data Mining in Pattern Recognition. MLDM 2012. **Lecture Notes in Computer Science**, vol 7376. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-31537-4_13 doi:10.1007/978-3-642-31537-4_13

ELEKTRONİK KAYNAKLAR

URL-1: Anonim (2016) <https://otsimo.com/tr/otizm-spektrum-bozuklugu-nedir/>

URL-2: Durakcan, Y. (2016). "Otizmli Bireylerin Beyinleri, Alışılmadık Şekilde Simetrik Özellik Gösteriyor" <https://bilimfili.com/otizmli-bireylerin-beyinleri-alisilmadik-sekilde-simetrik-ozellik-gosteriyor>

URL-3: Tordjman S. (2011) "Évolution du concept d'autisme : nouvelles perspectives à partir des données génétiques", L'information psychiatrique <https://www.cairn.info/revue-l-information-psychiatrique-2011-5-page-393.htm>

URL-4: Volkmar F., Pauls D. (2003) "Otizm", Seminer <https://www.sciencedirect.com/science/article/abs/pii/S0140673603144716>

URL-5: Allison C., Auyeung B., ve Baron-Cohen S. (2012). Journal of the American Academy of Child and Adolescent Psychiatry 51(2):202-12. <https://www.autismalert.org/uploads/PDF/SCREENING--AUTISM--QCHAT-10%20Question%20Autism%20Survey%20for%20Toddlers.pdf>

URL-6: Hidayet Takcı 2011 https://verimadencisi.blogspot.com/2011/06/oznitelikleri-secsek-de-mi-veri.html

URL-7: Priyadarshiny U. (2019). "Introduction to Classification Algorithms" <https://dzone.com/articles/introduction-to-classification-algorithms>

URL-8: **“**Seaborn’a Hızlı Başlangıç”
< https://veribilimcisi.com/2017/09/06/seaborna-hizli-bir-baslangic/>
URL-9: (https://analyticsindiamag.com/a-beginners-guide-to-regression-techniques/)
URL-10: Shanghai Arch Psychiatry. 2015 Apr 25; 27(2): 130–135.
doi:10.11919/j.issn.1002-0829.215044)
<link:https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/>
URL-11: Rish I. “An empirical study of the naive Bayes classifier”
<https://www.cc.gatech.edu/~isbell/reading/papers/Rish.pdf>
URL-12: Noble W. <https://www.ifi.uzh.ch/dam/jcr:00000000-7f84-9c3b-ffff-ffffc550ec57/what_is_a_support_vector_machine.pdf/>
URL-13: Probst P., Wright M., Boulesteix A. (2019) “Hyperparameters and Tuning Strategies for Random Forest”, <https://arxiv.org/pdf/1804.03515.pdf>

DİĞER KAYNAKLAR

ACHENIE K., vd. (2019). “Yeni Yürüyen Çocuklarda Otizm Taraması için Makine Öğrenimi Stratejisi”
AMMAR S. (2010). “Modèles Graphiques Probabilistes pour l’Estimation de Densité en grande dimension: applications du principe Perturb & Combine pour les mélanges d’arbres.” fftel-00568136f
KING G. ve ZENG L. (2001). “Logistic regression in rare events” (link:http://www.econ.upf.edu/~satorra/dades/M2012LogisticRegressionDayton.pdf)

EKLER

EK-A

A.1: Kütüphanelerin Tanımlanması

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
# ignore update warnnings
warnings.filterwarnings("ignore")
```

EK-B

B.1: Sınıflandırma İçin Algoritma Modellerini İçe Aktarma

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import RandomForestClassifier
```

EK-C

C.1: Regresyon İçin Algoritma Modellerini İçe Aktarma

```
from sklearn.linear_model import LinearRegression,Ridge,Lasso,RidgeCV,
ElasticNet, LogisticRegression
from sklearn.ensemble import
GradientBoostingRegressor,RandomForestRegressor,AdaBoostRegressor,
BaggingRegressor
```

EK-D

D.1: Model Hazırlama

```
from sklearn.preprocessing import  Normalizer , scale
from sklearn.impute import SimpleImputer
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import RFECV
from sklearn.model_selection import GridSearchCV , KFold , cross_val_score,
ShuffleSplit, cross_validate

# Preprocessing :
```

```
from sklearn.preprocessing import MinMaxScaler , StandardScaler,
LabelEncoder
from sklearn.impute import SimpleImputer
# Regression
from sklearn.metrics import mean_squared_log_error,mean_squared_error,
r2_score,mean_absolute_error
# Classification
from sklearn.metrics import
accuracy_score,precision_score,recall_score,f1_score, classification_report

print("Setup Models complete...")
```

EK-E

E.1: Veri Setini Aktarma

```
data = pd.read_csv("datasetim.csv",sep=';')
data.head()
```

EK-F

F.1: Veri Setinde İstatistik

```
data.describe()
```

EK-G

G.1: Gereksiz Özelliğin Çıkartılması

```
data.drop(['Case_No'], axis = 1, inplace = True)
data.columns
Index(['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'A9', 'A10
', 'Age_Mons',
       'Qchat-10-Score', 'Sex', 'Ethnicity', 'Jaundice', 'Family_
mem_with_ASD',
       'Class'],
      dtype='object')
```

EK-H

H.1: Veri Tipini Belirleme

```
data.dtypes
```

EK-I

I.1: Bağıntı Kontrolü

```
corr = data.corr()
plt.figure(figsize = (15,15))
sns.heatmap(data = corr, annot = True, square = True, cbar = True)
<matplotlib.axes._subplots.AxesSubplot at 0x1187db250>
```

EK-J

J.1: Filtreleme

```
plt.figure(figsize = (23,8))
sns.countplot(x = 'Ethnicity', data = data)
<matplotlib.axes._subplots.AxesSubplot at 0x1192fd9d0>
```

EK-K

K.1: Verideki sütunları oluşturma

```
data.columns
Index(['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'A9', 'A10', 'Age_Mons',
       'Qchat-10-Score', 'Sex', 'Ethnicity', 'Jaundice', 'Family_mem_with_ASD',
       'Class'],
      dtype='object')
```

EK-L

L.1: Verinin %20sinin Eğitim İçin Ayrılması

```
data.drop('Qchat-10-Score', axis = 1, inplace = True)
X = data.drop(['Class'], axis = 1)
Y = data['Class']
# 20% for training
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.20, random_state
= 7)
```

EK-M

M.1: Modellerin Öğretilmesi

```
models = []
models.append(('Logistic Regression:', LogisticRegression()))
models.append(('Decision Tree      :', DecisionTreeClassifier()))
models.append(('Naive Bayes        :', GaussianNB()))
models.append(('SVM                :', SVC()))
models.append(('Random Forest      :', RandomForestRegressor()))

for name, model in models:
   model.fit(x_train, y_train)
   pred = model.predict(x_test).astype(int)
   print(name, accuracy_score(y_test, pred))
```

EK-N

N.1: Sonuç

```
Logistic Regression: 1.0
Decision Tree      : 0.933649289099526
Naive Bayes        : 0.95260663507109
SVM                : 0.8530805687203792
Random Forest      : 0.6398104265402843
```

EK-O

O.1: Jupiter Notebook Kod Görüntüsü 1

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# kütüphanelerin girişinin yapılması
```

In [2]:
```
# Sınıflandırma için algoritma modellerinin girilmesi
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import RandomForestClassifier
```

In [3]:
```
# Regresyon için algoritma modellerinin girilmesi
from sklearn.linear_model import LinearRegression,Ridge,Lasso,RidgeCV, ElasticNet, LogisticRegression
from sklearn.ensemble import GradientBoostingRegressor,RandomForestRegressor,AdaBoostRegressor ,BaggingRegressor
```

EK-P

P.1: Jupiter Notebook Kod Görüntüsü 2

In [4]:
```
# Modelleri hazırlama :


from sklearn.preprocessing import  Normalizer , scale
from sklearn.impute import SimpleImputer
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import RFECV
from sklearn.model_selection import GridSearchCV , KFold , cross_val_score, ShuffleSplit, cross_validate

# ön işleme :
from sklearn.preprocessing import MinMaxScaler , StandardScaler,  LabelEncoder
from sklearn.impute import SimpleImputer
# regresyon
from sklearn.metrics import mean_squared_log_error,mean_squared_error, r2_score,mean_absolute_error
# sınıflandırma
from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score, classification_report

print("Setup Models complete...")
```

```
Setup Models complete...
```

In [23]:
```
data = pd.read_csv("datasetim.csv",sep=';')
```

In [24]:
```
data.head()
```

EK-Q

Q.1: Jupiter Notebook Kod Görüntüsü 3

Out[24]:

	Case_No	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Age_Mons	Qchat-10-Score	Sex	Ethnicity	Jaundice	Family_mem_with_ASD	Who completed the test	Class
0	1	0	0	0	0	0	0	1	1	0	1	28	3	f	middle eastern	yes	0	family member	No
1	2	1	1	0	0	0	1	1	0	0	0	36	4	m	White European	yes	0	family member	Yes
2	3	1	0	0	0	0	0	1	1	0	1	36	4	m	middle eastern	yes	0	family member	Yes
3	4	1	1	1	1	1	1	1	1	1	1	24	10	m	Hispanic	no	0	family member	Yes
4	5	1	1	0	1	1	1	1	1	1	1	20	9	f	White European	no	1	family member	Yes

In [25]:
```
data.describe()
```

EK-R

R.1: Jupiter Notebook Kod Görüntüsü 4

In [25]:
```
data.describe()
```

Out[25]:

	Case_No	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Ag
count	1054.000000	1054.000000	1054.000000	1054.000000	1054.000000	1054.000000	1054.000000	1054.000000	1054.000000	1054.000000	1054.000000	1054
mean	527.500000	0.563567	0.448767	0.401328	0.512334	0.524668	0.576850	0.649905	0.459203	0.489564	0.586338	27
std	304.407895	0.496178	0.497604	0.490400	0.500085	0.499628	0.494293	0.477226	0.498569	0.500128	0.492723	7
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	12
25%	264.250000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	23
50%	527.500000	1.000000	0.000000	0.000000	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	1.000000	30
75%	790.750000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	36
max	1054.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	36

In [8]:
```
data.drop(['Case_No'], axis = 1, inplace = True)
data.columns
```

Out[8]:
```
Index(['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'A9', 'A10', 'Age_Mons',
       'Qchat-10-Score', 'Sex', 'Ethnicity', 'Jaundice', 'Family_mem_with_ASD',
       'Class'],
      dtype='object')
```

EK-S

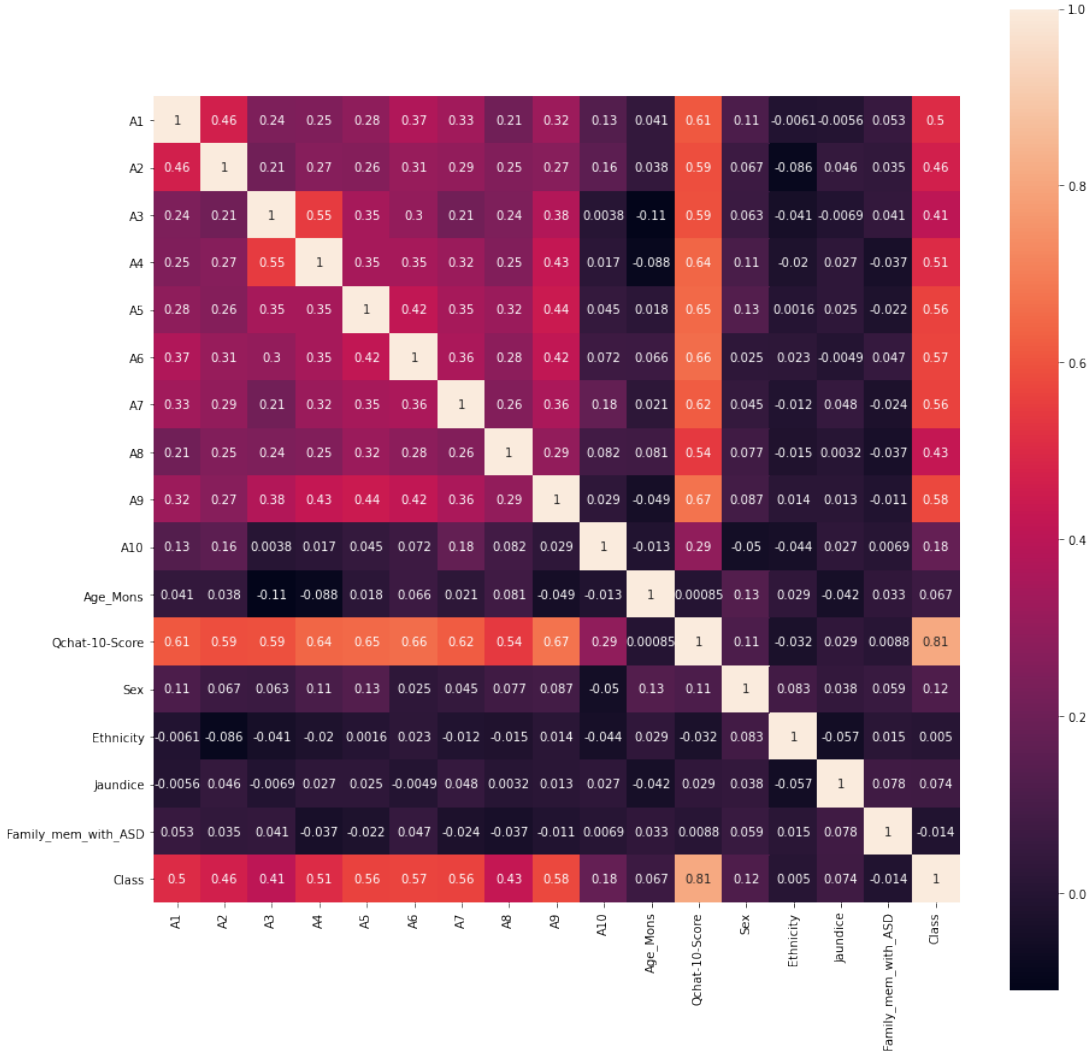
S.1: Jupiter Notebook Kod Görüntüsü 5

```
In [9]: data.dtypes

Out[9]: A1                    int64
        A2                    int64
        A3                    int64
        A4                    int64
        A5                    int64
        A6                    int64
        A7                    int64
        A8                    int64
        A9                    int64
        A10                   int64
        Age_Mons              int64
        Qchat-10-Score        int64
        Sex                   int64
        Ethnicity             int64
        Jaundice              int64
        Family_mem_with_ASD   int64
        Class                 int64
        dtype: object
```

```
In [10]: # korelasyon
         corr = data.corr()
         plt.figure(figsize = (15,15))
         sns.heatmap(data = corr, annot = True, square = True, cbar = True)

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x1187db250>
```

EK-T

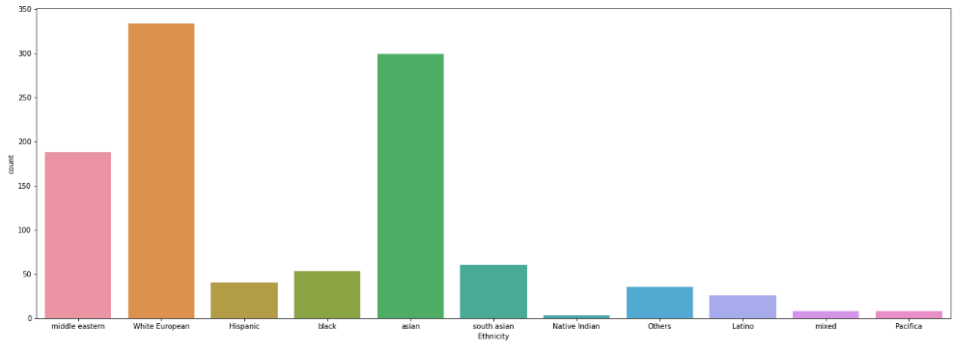
T.1: Jupiter Notebook Kod Görüntüsü 6



EK-U

U.1: Jupiter Notebook Kod Görüntüsü 7

```
In [26]: #Irka göre filtreleme
         plt.figure(figsize = (23,8))
         sns.countplot(x = 'Ethnicity', data = data)

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x1192fd9d0>
```



EK-V

V.1: Jupiter Notebook Kod Görüntüsü 8

```
In [12]: data.columns

Out[12]: Index(['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'A9', 'A10', 'Age_Mons',
                'Qchat-10-Score', 'Sex', 'Ethnicity', 'Jaundice', 'Family_mem_with_ASD',
                'Class'],
               dtype='object')

In [13]: data.drop('Qchat-10-Score', axis = 1, inplace = True)

In [14]: X = data.drop(['Class'], axis = 1)
         Y = data['Class']
         # 20% eğitim yani training için ayrılması
         x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.20, random_state = 7)
```

EK-Y

Y.1: Jupiter Notebook Kod Görüntüsü 9

```
In [15]: models = []
         models.append(('Logistic Regression:', LogisticRegression()))
         models.append(('Decision Tree      :', DecisionTreeClassifier()))
         models.append(('Naive Bayes        :', GaussianNB()))
         models.append(('SVM                :', SVC()))
         models.append(('Random Forest      :', RandomForestRegressor()))


         for name, model in models:
             model.fit(x_train, y_train)
             pred = model.predict(x_test).astype(int)
             print(name, accuracy_score(y_test, pred))

Logistic Regression: 1.0
Decision Tree      : 0.933649289099526
Naive Bayes        : 0.95260663507109
SVM                : 0.8530805687203792
Random Forest      : 0.6398104265402843
```

ÖZGEÇMİŞ

ÖGRENİM DURUMU **:** Yüksek Lisans (Devam Etmekte)

- **ÖN LİSANS** **:** Yıldız Teknik Üniversitesi – Bilgisayar Programcılığı
- **LİSANS** **:** İstanbul Aydın Üniversitesi – Yazılım Mühendisliği (İngilizce)
- **YÜKSEK LİSANS** **:** İstanbul Aydın Üniversitesi - Bilgisayar Mühendisliği

Asseco-SEE Teknoloji, 1.Seviye Destek Uzmanı (Şubat 2016– Ağustos 2013)
Proje Yöneticisi, Akinon (Ağustos 2019-*)