**Bebeğinizin Otizmli Olma Olasılığını Karar Ağacı (Decision Tree) Algoritması ile Tahmin Etme ve Python Dilinde Programlama**

**Beyza Beliz Coşkun 20231012131**

**Nişantaşı Üniversitesi YAPAY ZEKA PROJESİ**

**Bilgisayar Programcılığı (İÖ) 1. Sınıf**

# ÖZET

Karar Ağacı algoritması bir makine öğrenme algoritmasıdır. Sınıflandırma ve regresyon denetimli öğrenme algoritması olarak kabul edilir. Girdiğimiz veri setine göre hesaplama yapar ve tahmin oluşturur. Karar ağacı algoritmasını sağlık alanında kullandım. Python dilini kullandığım kodumda girdiğim belirtilere verdiğiniz cevaplar sonucunda bebeğinizin otizmli olma olasılığını karar ağacı algoritmasıyla hesaplama ve tahmin yapılmasını sağladım. Otizm ise henüz çaresi bulunmayan ve kişinin hasta olup olmadığı kolayca anlaşılamayan bir hastalık olmasına rağmen erken teşhis sonucunda ilaçlarla yavaşlatılabilir ve kişiyi daha iyi bir hale getirebilir.

**Anahtar Kelimeler: karar ağacı, tahmin, otizm**

# GİRİŞ

Makine Öğrenmesi (Machine Learning) Nedir?

Bilgisayar programlarının algoritmalar ve eğitim verileri aracılığıyla kalıpları öğrenebildiği bir yapay zeka uygulamasıdır. Bilgisayarları verilerden öğrenmeyi öğretmeye ve bunu yapmak için açıkça programlanmak yerine deneyimle geliştirmeye odaklanır. Makine öğrenmesinde algoritmalar kararları ve tahminleri yapmak için eğitilir. Makine öğrenmesi uygulamaları kullanımla gelişir ve daha fazla veri erişimi olduğunda daha doğru hale gelir.

Tüm sektörlerdeki işletmeler, makine öğrenmesinden farklı şekillerde faydalanır.

Bankacılık ve Finans

Risk yönetimi ve dolandırıcılığı önleme, makine öğrenmesinin finansal bağlamlarda değer sunduğu önemli alanlardandır.

Sağlık Hizmetleri

Tanılama araçları, hasta izleme ve salgınların tahmin edilmesi, makine öğrenmesinin hasta bakımının iyileştirilmesine yardımcı olduğu uygulamalara örnek verilebilir.

Ulaşım

Trafik anormalliği belirleme, teslimat yolu iyileştirmesi ve sürücüsüz araçlar, makine öğrenmesinin ulaşım üzerinde olumlu bir etki oluşturabileceğini gösteren örneklerdir.

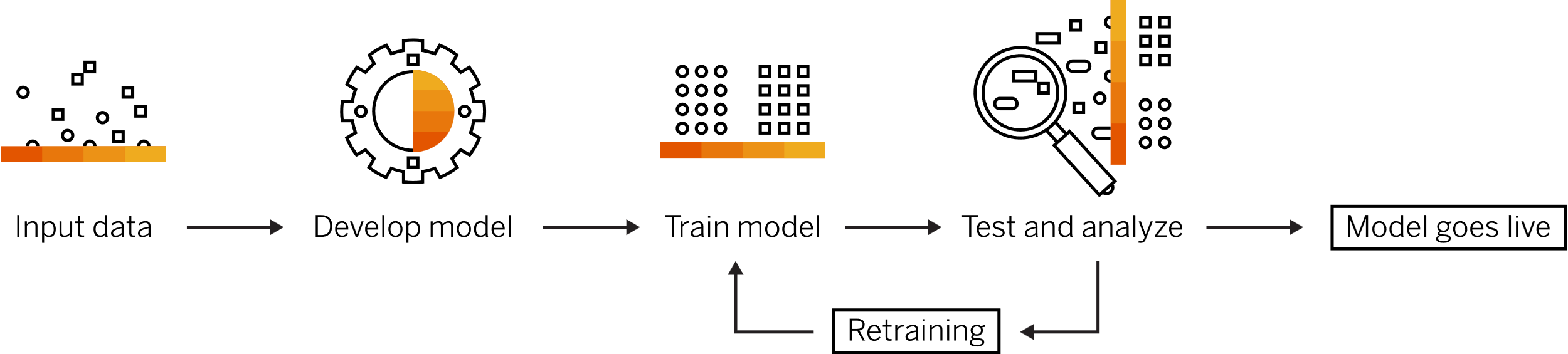
Müşteri hizmetleri

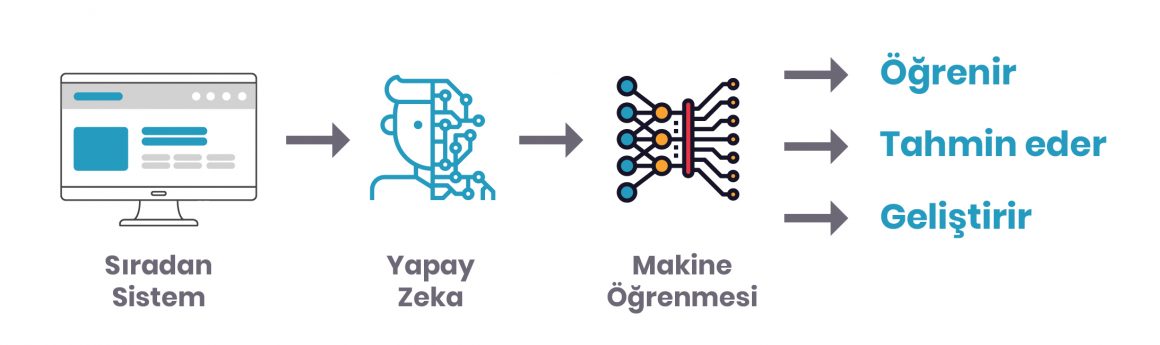
Sorulara yanıt verme, müşteri amaçlarını ölçme ve sanal yardım sunma, makine öğrenmesinin müşteri hizmetleri sektörüne sunduğu desteğe örnek verilebilir.

Tarım

İşgücü eksikliğini gidermek için robot geliştirme, bitki hastalıklarını tanılama ve toprağın durumunu izleme gibi konularda makine öğrenmesi, tarımın geliştirilmesine katkıda bulunabilir.

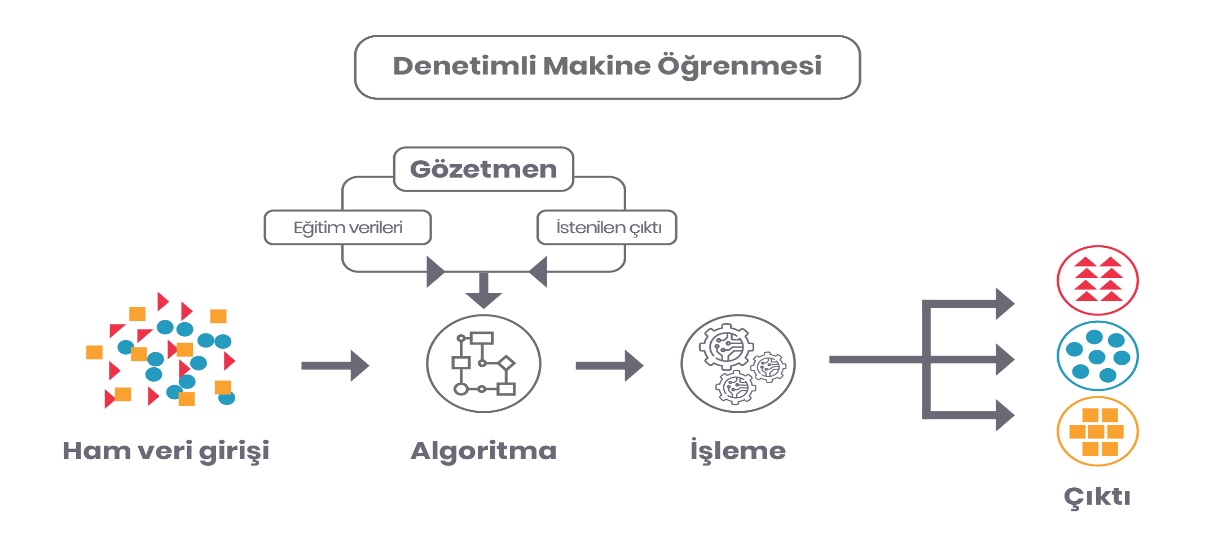
Bunun gibi birçok örnek mevcut.

Makine öğrenmesi, çeşitli algoritmik teknikler kullanılarak farklı makine öğrenmesi modelleri türlerinden oluşur. Verilerin niteliğine ve istenilen sonuca bağlı üç öğrenme modelinden biri kullanılabilir: denetimli, denetimsiz, yarı denetimli.



Denetimli Makine Öğrenmesi (Supervised Algorithms)

Denetimli makine öğreniminde amaç, bir dizi bağımsız değişken üzerinden tanımlanan bir işlevi kullanarak hedef değişkeni tahmin etmektir. Geliştirici eğitim verilerini etiketler, algoritmanın izleyeceği katı kuralları ve sınırları belirler. Böylece algoritmalar, gelecekteki olayları tahmin etmek için etiketli örnekleri kullanarak geçmişte öğrenilenleri yeni verilere uygulayabilir. Denetlenen algoritmalar bir dizi giriş verisini ve beklenen sonuçları tanımlayarak çalışır Algoritma, eşlemeleri ve tahminleri doğru bulunduğunda başarılı olarak kabul edilir.

Denetimli makine öğreniminin en ünlü örneklerinden biri, Boston konut fiyatları veri kümesidir. Satışı yapılmış evleri, özelliklerini ve satış fiyatlarını içeren bu veri kümesi herhangi bir evin satış fiyatını tahmin edebilen bir makine öğrenimi modeli oluşturmayı amaçlamıştır.

Makine öğrenmesi algoritmasının özel algoritmaları vardır. Bunlar:

Regresyon

Sınıflandırma

Topluluk

İlişkilendirme

Kümeleme

Sınıflandırma ve regresyon denetimli öğrenme algoritması olarak kabul edilir. Makine öğrenmesi algoritmaları temel olarak olayları sınıflandırmak, örnekler bulmak, sonuçları tahmin etmek ve bilinçli kararlar vermek için tasarlanmıştır. Karar ağacı algoritması bir sınıflandırma aynı zamanda da regresyon algoritmasıdır.

Regresyon

Regresyon algoritmaları; bağımsız değişkenlerin bağımlı olanı ne kadar etkilediğini anlamak için, farklı değişkenler arasındaki olası ilişkileri bulmak amacıyla kullanılan denetimli algoritmalardır. Regresyon analizini bir denklem olarak düşünebilirsiniz. Örneğin, y = 2x + z denkleminde y bağımlı değişken ve x ile z bağımsız değişkenlerdir. Regresyon analizi, x ve z’nin y’nin değerini ne kadar etkilediğini bulur. Aynı mantık, daha gelişmiş ve karmaşık problemler için de geçerlidir ve bu amaçla kullanılan birçok regresyon algoritması türü bulunur.

Doğrusal Regresyon (Linear Regression)

Doğrusal regresyon, özünde, iki değişken arasındaki ilişkiyi belirlemeye yönelik doğrusal bir yaklaşımdır; bu değerlerden biri bağımlı bir değerdir, diğeri bağımsızdır. Bir değişkendeki değişikliğin diğerini nasıl etkilediğini ve sonucun olumlu veya olumsuz olmasına neden olan ilişkiyi anlamaya yarar.

Doğrusal Regresyon, y = a + bx şeklinde bir çizgi olarak temsil edilir ve tahmin edilen çıktının sürekli olduğu ve sabit bir eğime sahip olduğu durumlarda uygulanır: Satışları tahmin etmek, risk değerlendirmesi, tahmine dayalı analitik, müşteri anketi sonuç analizi, hava durumu veri analizi, ürün fiyatlarını optimize etmek

Lojistik Regresyon

Lojistik Regresyon algoritması, genellikle iki değerden birinin; geçti veya kaldı, doğru veya yanlış gibi durumlarla sonuçlandığı ikili sınıflandırma problemlerinde kullanılır. Bağımlı değişkenin yanıtın iki kategorisinden birine girme olasılıklarını tahmin etme ihtiyacı için en uygun algoritmadır. Kategorik verileri analiz etmek için başvurulan bu algoritma; verilen el yazısının söz konusu kişiyle eşleşip eşleşmediğini bulma ve ilerleyen aylarda petrol fiyatlarının yükselip yükselmeyeceğini tahmin etme gibi durumlarda kullanılır.

Ridge Regresyon

Regresyon modeli çok karmaşık hale geldiğinde, ridge regresyonu; modelin katsayılarının boyutunu düzeltir.

Lasso Regresyon

Lasso regresyon, değişkenleri seçmek ve düzenlemek için kullanılan regresyon türüdür.

Polinom Regresyon (Polynomial Regression)

Bu tür algoritmalar, doğrusal olmayan verileri uydurmak için kullanılır. Burada en iyi tahmin düz bir çizgi değildir; tüm veri noktalarına uymaya çalışan bir eğridir.

Sınıflandırma (Classification)

Makine öğreniminde sınıflandırma, öğeleri önceden kategorize edilmiş bir eğitim veri kümesine göre kategorilere ayırma sürecidir. Sınıflandırma algoritmaları, yeni bir öğenin tanımlanan kategorilerden birine girme olasılığını hesaplamak için eğitim verilerinin kategorizasyonunu kullanır. Sınıflandırma algoritmalarının en iyi bilinen örneği, gelen e-postaların “spam” veya “spam değil” şeklinde filtrelenmesidir.

Farklı sınıflandırma algoritmalarından bazıları şunlardır;

KNN (K-Nearest Neighbors)

Karar Ağaçları (Decision Trees)

Naive Bayes

SVM (Support Vector Machine)

metin, ekran görüntüsü, dikdörtgen, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturulduKarar Ağaçları

Karar ağaçları hem sınıflandırma (kategorik veriler) hem de regresyon (numerik veriler) problemlerin çözümünde kullanılabilen veri madenciliği ve makine öğrenme algoritmasıdır. Veri madenciliğin temel amacı veriyi analize uygun hale getirmektir. Çok güçlü algoritmalar olan karar ağaçları kompleks ve büyük veri setlerinin çözümünde oldukça yararlanılan denetimli öğrenme algoritmaları ailesine aittir. Bir karar ağacı temel olarak 4 ana düğüm (node)’den oluşur. Bunlar sırasıyla kök (root) düğüm, iç düğümler ve yaprak (leaf) düğümlerden oluşur. Bu düğümler dallar aracılığıyla birbirine bağlanır. Terminal ve yaprak düğümleri karar düğümleridir. Karar ağacı öğrenmesinde, karar ağacının kök düğümünden başlayarak yukarıdan aşağıya doğru oluşturulur. Karar ağacının her düğümü, veri kümesini iki veya daha fazla alt kümeye bölen bir testtir. Karar ağacının her yaprağı ise, veri kümesinin bir sınıfını veya değerini temsil eder. Biçimsel olarak ağaçtan, işleyiş olarak ise insan karar verme sürecinden ilham almaktadır. Karar ağaçlarında hedef değişken (target variable) kategorik ise sınıflandırma algoritması, numerik (sürekli veya kesikli) değişken ise regresyon algoritması kullanılır.

Karar ağaçları, verilerdeki kalıpları belirlemeye yardımcı olan bir tür makine öğrenimi modelidir. Bir dizi girdi değeri alarak ve ardından ağacın mevcut veriler için en iyi kararın ne olduğuna inandığına bağlı olarak bunları farklı dallara ayırarak çalışır. Ağacın verdiği her karar, mümkün olan en iyi seçeneğin belirlenmesinde bir adım olarak düşünülebilir. Bir karar ağacının nihai sonucu genellikle her bir girdi değerinin (önceden belirlenmiş) çıktı değerlerinden birine karşılık gelme olasılığını yansıtan belirli bir çıktı değerleri kümesidir. Karar ağaçlarının diğer modellere göre sahip olduğu önemli bir avantaj, karmaşık veri setleriyle hızlı bir şekilde başa çıkabilmeleridir. Bunun nedeni, verileri daha küçük parçalara bölmek için basit kurallar kullanmaları ve daha sonra bu parçalar içindeki kalıpları aramalarıdır. Bu yöntem genellikle “böl ve yönet” olarak adlandırılır ve karar ağaçlarının sorunları diğer model türlerinden çok daha hızlı bir şekilde ele almasını sağlar. Karar ağaçlarının bir diğer büyük avantajı da çok yönlü olmalarıdır. Bu, örüntü tanıma, tahmin ve sınıflandırma dahil ancak bunlarla sınırlı olmamak üzere çeşitli farklı görevler için kullanılabilecekleri anlamına gelir. Karar ağaçları iş dünyasında finans, pazarlama, operasyon ve strateji gibi farklı alanlarda kullanılmaktadır.

Sağlık Hizmetleri: Karar ağaçları hastalıkları tahmin etmek, tedavileri değerlendirmek ve hasta bakımını iyileştirmek için kullanılır.

Eğitim: Karar ağaçları, eğitimsel veri madenciliği, öğrenci verilerinin sınıflandırılması ve öğrenci performansının tahmin edilmesi için kullanılır.

Veri Madenciliği: Karar ağaçları, kümeleme ve sınıflandırma gibi veri madenciliği görevleri için kullanılır.

Robotik: Karar ağaçları robot navigasyonu ve kontrolünde kullanılır.

Bilgisayarlı Görme: Karar ağaçları nesne tanıma ve görüntü sınıflandırmada kullanılır.

Doğal Dil İşleme: Karar ağaçları metinleri sınıflandırmak ve metinlerdeki örüntüleri tanımlamak için kullanılır.

Üretim: Karar ağaçları, üretim süreçlerinde tahmine dayalı modelleme ve öngörüde kullanılır.

Oyun: Karar ağaçları bilgisayar oyunlarında ve yapay zekada da kullanılır.

Makine Öğrenimi Karar Ağaçları Ne İçin Kullanılır?

Bu model çok yönlüdür, dolayısıyla karar ağacı algoritması makine öğreniminde birçok uygulama alanı bulmaktadır:

Veri madenciliği

Veri bilimi

İstatistikler

Karar Ağaçların Avantajları ve Dezavantajları

Avantajları:

Anlaması ve yorumlaması kolaydır.

Kullanılan ağaç yapılar görselleştirilebilir.

Az oranda bir veri hazırlığına ihtiyaç duyar. Fakat unutulmamalıdır ki bu model kayıp değerleri desteklememektedir.

Hem sayısal hem de kategorik verileri işleyebilir.

Çok çıktılı problemleri ele alabilmektedirler.

İstatistiksel testler kullanılarak bir modelin doğrulanması mümkündür.

Parametrik olmayan bir yöntem olarak düşünülebilir. Yani uzay dağılımı ve sınıflandırma yapısı hakkında bir yaklaşıma sahip değillerdir.

Dezavantajları:

Veriyi iyi bir şekilde açıklamayan aşırı karmaşık ağaçlar üretilebilir. Bu durumda ağaç dallanması takip edilemeyebilir.

Ezbere öğrenme yaşanabilir (“over-fitting”). Bu problemin çözümü için model parametrelere kısıtlamalar ve budama gibi yöntemler kullanılabilir.

Kararsızdırlar, yani verilerdeki küçük bir değişikliğin, en iyi durumdaki karar ağacının yapısında büyük bir değişikliğe yol açabileceği anlamına gelmektedir.

Karar ağacı algoritmaları, verileri sınıflandırmak veya regresyon yapmak için kullanılan ağaç yapılı yöntemlerdir. Karar ağacı algoritmaları arasında ID3, C4.5, C5.0, CART, CHAID, MDL, QUEST gibi farklı algoritma yaklaşımları bulunmaktadır. Bu algoritmalar, veri kümesindeki öznitelikleri farklı ölçütlerle seçerek dallanma ve budama işlemleri yaparlar. ID3 algoritması bilgi kazanımı ölçüsünü, CART algoritması Gini indeksini, CHAID algoritması ise Ki Kare Testini kullanır. Karar ağacı oluşturulurken verilerin homojen bir şekilde dağılması hedeflenmektedir. Homojenliği hesaplamak için entropi veya gini gibi kavramları kullanır. Entropi veya gini değeri ne kadar düşükse, veri kümesi o kadar düzenli ve saf (tek sınıfa yakın) demektir.

Entropi: Bir grup örnekte safsızlık, düzensizlik veya belirsizlik ölçüsüdür. Entropi, bir karar ağacının verileri bölmeye nasıl karar verdiğini kontrol eder.

Gini: Gini, bir karar ağacının verileri bölmeye nasıl karar verdiğini kontrol eder. Gini, bir veri kümesinin tek bir sınıfa ne kadar yakın olduğunu gösterir.

Otizm Nedir? Belirtileri Nelerdir?

Otizm, genellikle küçük çocuklarda görülen geniş bir yelpazedeki beyin rahatsızlıklarıdır. Otizm, Otizm Spektrum Bozukluğu olarak da adlandırılır. Otizm, bireyin iletişim kurma ve başkalarına duygusal olarak bağlanma kabiliyetini azaltır. Otizm spektrum bozukluğundaki “spektrum”, belirtilerin, becerilerin ve bozukluğun geniş bir yelpazede olduğunu ifade eder. Otizm spektrumundaki en yaygın üç rahatsızlık, otizm, asperger sendromu ve yaygın gelişimsel bozukluktur.

Otizm, çeşitli yaşlarda gelişebilir. Bazı bebekler erken otizm belirtileri gösterebilirken bazıları 15-30 ayda normal gelişebilir. Otizmin erken belirtileri fark edilebilir. Ebeveynler veya doktorlar otizmli bir bebeği teşhis edebilirse, tedavi bebeğin beynin gelişimini önemli ölçüde iyileştirebilir. Otizm belirtileri genellikle 12 ila 18 ay arasında görülür, ancak ebeveynler sonraki zamanlarda da otizm belirtilerine dikkat etmelidir. Otizmden etkilenen çocuklar, aşırı sinapslara veya beyin hücreleri arasındaki bağlantılara sahiptir. Bunun nedeni, beyin gelişiminde ortaya çıkan normal budamanın kapanmasıdır. Tipik budama işlemi, ergenlik çağının sonlarına doğru korteks sinapslarının yarısını ortadan kaldırmayı içerir. Kortikal sinapslar, duyuların düşünülmesi ve işlenmesi için merkezi olan kortekste ortaya çıkar.Otizmi olan bazı çocukların beyinleri normalden daha büyüktür, ancak bulgular tutarsızdır. Otizmi olan bazı çocukların MRG incelemelerinde anormal kortikal yanıtlar, bazılarında başka anormallikler görülür. Beyin çalışmalarındaki gelecekteki ilerlemeler, beyindeki otizmin rolünü daha iyi anlamamızı sağlayabilir. Otizmi teşhis etmek zor olabilir, çünkü kan testleri gibi çocukları teşhis edebilecek tıbbi testler yoktur. Bu nedenle tedaviler yıllarca ertelenebilir. Asperger sendromu, otizm spektrum bozukluğunun bir parçasıdır. Asperger sendromu “yüksek işleyen” bir otizm spektrum bozukluğu türüdür. Çocuklar normal veya üstün zekaya sahip olabilir ancak insanlarla ilişki kurmada ve arkadaş edinmede zorluk çekebilir. Ayrıca uzmanlaşmış görevlere odaklanma eğilimindedirler.

Klasik Otizm: Sadece otizm olarak da bilinen klasik otizm, sosyalleşme ve iletişim sorunlarının ağır biçimde görüldüğü, dil ve konuşma gecikmeleri, davranış bozuklukları gibi belirtilerle kendini gösterir.

Asperger Sendromu: Otizmin daha hafif seyrettiği bu tip bozukluklarda, diğer otizm türlerinde olduğu gibi sosyal davranışlarda anormallik gözlenir. Fakat Asperger sendromunda konuşma bozukluğu ve zihinsel engellilik gibi problemler görülmez.

Atipik Otizm: Klasik otizm ve Asperger sendromu ile ortak belirtileri bulunan, ancak tüm kriterleri taşımayan bireyler atipik otizm bozukluğu tanısı alır. Bu tip otizm bozukluğu taşıyan kişiler, sosyalleşmede ve normal yaşamlarını sürdürmede ileri derecede sorun yaşamazlar ve diğer otizm çeşitlerine göre topluma daha kolay uyum gösterirler. Hastalığın ilaçla kesin bir tedavi yöntemi henüz olmasa da erken tanı hastalığın yarattığı sonuçları hafifletmede etkilidir. Tedavi yöntemi olarak uygulanan rehabilitasyon ve ilaçlar, başta otizmli bireyin çevresindeki diğer kişilerin hayatını kolaylaştırmak ve son tahlilde otizmli bireyin zararlı davranışlarını azaltmayı amaçlamaktadır. Hastalığın erken teşhisi ileride yüksek maliyetlere neden olabilecek otizmin yavaşlatılması için son derece önemlidir. Bu nedenle, 12 aydan itibaren tüm bebeklere otizm testi yapılabilir.

AKIŞ DİYAGRAMI

Bebeğin yaşını girin (1,2,3)

Eğer 1 yaşındaysa 1 yaş koşuluna uygun 5 belirti sorusu cevabı girin

Eğer 2 yaşındaysa 2 yaş koşuluna uygun 5 belirti sorusu cevabı girin

Eğer 3 yaşındaysa 3 yaş koşuluna uygun 5 belirti sorusu cevabı girin

Otizmli olma olasılığını hesapla

## KOD

import pandas as pd

from sklearn.tree import DecisionTreeClassifier

data = pd.read\_csv("data.csv")

x = data.drop(columns=["Otizmli\_Olma\_Olasiligi"])

y = data["Otizmli\_Olma\_Olasiligi"]

model = DecisionTreeClassifier()

model.fit(x, y)

yas\_degerleri = [1, 2, 3]

values = [0, 1]

yas = int()

belirti1, belirti2, belirti3, belirti4, belirti5 = int(), int(), int(), int(), int()

yas\_secimi = False

while yas\_secimi is not True:

yas = int(input('Bebeğin yaşını girin(1,2,3):'))

if yas in yas\_degerleri:

yas\_secimi = True

if yas == 1:

choice = False

while choice is not True:

belirti1 = int(input('Hangisi (0-1) birini seç ? (Rol Yapma Becerisi Az (0), Rol Yapabiliyor (1):'))

if belirti1 in values:

choice = True

choice = False

while choice is not True:

belirti2 = int(input('Hangisi (0-1) birini seç ? (Icine Kapanik (0), Sosyal Etkilesimi Sever (1):'))

if belirti2 in values:

choice = True

choice = False

while choice is not True:

belirti3 = int(input(

'Hangisi (0-1) birini seç ? (Sozlu ve Sozsuz Iletisim Kurma Az (0), Sozlu ve Sozsuz Iletisim Kurmayi Sever (1):'))

if belirti3 in values:

choice = True

choice = False

while choice is not True:

belirti4 = int(input('Hangisi (0-1) birini seç ? (Goz temasi kurmaz (0), Goz temasi normal(1):'))

if belirti4 in values:

choice = True

choice = False

while choice is not True:

belirti5 = int(input(

'Hangisi (0-1) birini seç ? (Ebeveynin sesine karsilik vermez (0), Ebeveynin sesine karsilik verir (1): '))

if belirti5 in values:

choice = True

tahmin = model.predict([[yas, belirti1, belirti2, belirti3, belirti4, belirti5]])

print(tahmin)

elif yas == 2:

choice = False

while choice is not True:

belirti1 = int(input(

'Hangisi (0-1) birini seç ? (Gorme, Isitme, Koku, Dokunma,Tat Hassasligi (0), Gorme, Isitme, Koku, Dokunma,Tat Normal (1):'))

if belirti1 in values:

choice = True

choice = False

while choice is not True:

belirti2 = int(input(

'Hangisi (0-1) birini seç ? (Giysi Giymek Zorunda Kaldiklarinda Huysuzlanma (0), Giysi Giymek Zorunda Kaldiklarinda Problem Yok (1):'))

if belirti2 in values:

choice = True

choice = False

while choice is not True:

belirti3 = int(input(

'Hangisi (0-1) birini seç ? (Vucut Hareketlerini Defalarca Tekrarlama (0), Vucut Hareketleri Normal (1):'))

if belirti3 in values:

choice = True

choice = False

while choice is not True:

belirti4 = int(

input('Hangisi (0-1) birini seç ? (Alisilmadik Sekilde Objelere Bagli Olma (0), Objelere Baglilik Normal (1):'))

if belirti4 in values:

choice = True

choice = False

while choice is not True:

belirti5 = int(input(

'Hangisi (0-1) birini seç ? (Gunluk Yasam Rutinleri Degisince Cok Sikinti Duymalari (0), Gunluk Yasam Rutinleri Degisince Cabuk Adapte Olma (1):'))

if belirti5 in values:

choice = True

tahmin2 = model.predict([[yas, belirti1, belirti2, belirti3, belirti4, belirti5]])

print(tahmin2)

elif yas == 3:

choice = False

while choice is not True:

belirti1 = int(

input('Hangisi (0-1) birini seç ? (Kelimeler Yerine Jest Kullanma (0), Kelimeleri Duzgunce Kullanma (1):'))

if belirti1 in values:

choice = True

choice = False

while choice is not True:

belirti2 = int(

input('Hangisi (0-1) birini seç ? (Dilin Yavas Gelismesi Veya Hic Gelismemesi (0), Guzel Dil Gelisimi (1):'))

if belirti2 in values:

choice = True

choice = False

while choice is not True:

belirti3 = int(input(

'Hangisi (0-1) birini seç ? (Nesnelere Bakmak Icin Bakis Ayarlama Guclugu (0), Nesnelere Bakmak Icin Bakis Ayarlama Problemi Yok (1):'))

if belirti3 in values:

choice = True

choice = False

while choice is not True:

belirti4 = int(input(

'Hangisi (0-1) birini seç ? (Reklamlar Gibi Ezberlenmis Sozleri Durmadan Tekrarlar (0), Reklamlar Gibi Ezberlenmis Sozleri Durmadan Tekrarlamaz (1):'))

if belirti4 in values:

choice = True

choice = False

while choice is not True:

belirti5 = int(input(

'Hangisi (0-1) birini seç ? (Kendinden Dogru Sekilde Bahsetmez (0), Kendinden Dogru Sekilde Bahsedebilir (1):'))

if belirti5 in values:

choice = True

tahmin3 = model.predict([[yas, belirti1, belirti2, belirti3, belirti4, belirti5]])

print(tahmin3)



Veri Seti

Yas,Belirti1,Belirti2,Belirti3,Belirti4,Belirti5,Otizmli\_Olma\_Olasiligi

1,0,0,0,0,0,%100

1,1,1,1,1,1,%0

1,1,0,0,0,0,%80

1,0,1,0,0,0,%80

1,0,0,1,0,0,%80

1,0,0,0,1,0,%80

1,0,0,0,0,1,%80

1,1,1,0,0,0,%60

1,1,1,1,0,0,%40

1,1,1,1,1,0,%20

1,0,1,1,0,0,%60

1,0,0,1,1,0,%60

1,0,0,0,1,1,%60

1,0,1,0,1,0,%60

1,0,1,0,0,1,%60

1,1,0,1,0,0,%60

1,1,0,0,1,0,%60

1,1,0,0,0,1,%60

1,0,0,1,0,1,%60

1,1,0,0,1,1,%40

1,1,1,0,1,0,%40

1,1,0,1,1,0,%40

1,1,0,1,0,1,%40

1,1,1,0,0,1,%40

1,0,1,1,1,0,%40

1,0,1,0,1,1,%40

1,0,1,1,0,1,%40

1,0,0,1,1,1,%40

1,0,1,1,1,1,%20

2,0,0,0,0,0,%100

2,1,1,1,1,1,%0

2,1,0,0,0,0,%80

2,0,1,0,0,0,%80

2,0,0,1,0,0,%80

2,0,0,0,1,0,%80

2,0,0,0,0,1,%80

2,1,1,0,0,0,%60

2,1,1,1,0,0,%40

2,1,1,1,1,0,%20

2,0,1,1,0,0,%60

2,0,0,1,1,0,%60

2,0,0,0,1,1,%60

2,0,1,0,1,0,%60

2,0,1,0,0,1,%60

2,1,0,1,0,0,%60

2,1,0,0,1,0,%60

2,1,0,0,0,1,%60

2,0,0,1,0,1,%60

2,1,0,0,1,1,%40

2,1,1,0,1,0,%40

2,1,0,1,1,0,%40

2,1,0,1,0,1,%40

2,1,1,0,0,1,%40

2,0,1,1,1,0,%40

2,0,1,0,1,1,%40

2,0,1,1,0,1,%40

2,0,0,1,1,1,%40

2,0,1,1,1,1,%20

3,0,0,0,0,0,%100

3,1,1,1,1,1,%0

3,1,0,0,0,0,%80

3,0,1,0,0,0,%80

3,0,0,1,0,0,%80

3,0,0,0,1,0,%80

3,0,0,0,0,1,%80

3,1,1,0,0,0,%60

3,1,1,1,0,0,%40

3,1,1,1,1,0,%20

3,0,1,1,0,0,%60

3,0,0,1,1,0,%60

3,0,0,0,1,1,%60

3,0,1,0,1,0,%60

3,0,1,0,0,1,%60

3,1,0,1,0,0,%60

3,1,0,0,1,0,%60

3,1,0,0,0,1,%60

3,0,0,1,0,1,%60

3,1,0,0,1,1,%40

3,1,1,0,1,0,%40

3,1,0,1,1,0,%40

3,1,0,1,0,1,%40

3,1,1,0,0,1,%40

3,0,1,1,1,0,%40

3,0,1,0,1,1,%40

3,0,1,1,0,1,%40

3,0,0,1,1,1,%40

3,0,1,1,1,1,%20

KARAR AĞACI AKIŞ DİYAGRAMI

%40

*Belirti 1, Belirti 2 Evet*

*Belirti 3, Belirti 4 Hayır*

%0

*Belirti 5 Hayır*

%0

*Evet*

*Belirti 1, Belirti 2 Hayır*

Otizmli olma olasılığı

%0

*Evet Belirti 3, Belirti 4 Hayır*

%0

*Belirti 5 Evet*

*Hayır* %20

*Belirti 1, Belirti 2 Hayır*

%0

*Belirti 3, Belirti 4 Hayır*

*Belirti 5 Hayır* %0

%0

KAYNAKLAR

<https://www.turhost.com/blog/makine-ogrenmesi-machine-learning-nedir/>

<https://medium.com/@mehmetkarataslar/maki%CC%87ne-%C3%B6%C4%9Frenmesi%CC%87-73da4646abd8#_2.1_Karar_A%C4%9Fa%C3%A7lar%C4%B1>

<https://www.nkfu.com/1-2-yasindaki-cocuklarda-otizm-belirtileri-erken-teshis-ve-tedavi-olanaklari/>

<https://www.cocukvegenc.com/otizm-belirtileri-nelerdir/>

<https://www.mindonmap.com/tr/blog/what-is-decision-tree/>

<https://azure.microsoft.com/tr-tr/resources/cloud-computing-dictionary/what-is-machine-learning-platform/>

<https://www.sap.com/turkey/products/artificial-intelligence/what-is-machine-learning.html>

<https://www.medicana.com.tr/saglik-rehberi-detay/12100/otizm-nedir-belirtileri-nelerdir>

<https://kodmatik.net/algoritma-nedir-akis-diyagrami-sekilleri-cizimi/>