Magic Gamma Telescope Veri Kümesinin Makine Öğrenme Yöntemleri ile Sınıflandırılması

Gamze Artan
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
gamzeartn@gmail.com

Deniz seviyesinden yaklaşık 2000 metre yukarıda konumlandırılmış iki Cherenkov teleskobundan oluşan Magic Gamma Telescope sistemi gama ışınlarını tespit edebilmek üzere tasarlanmıştır. Üst atmosfer tabakalarında gözlem yapmaktadırlar. Gama ışınlarının hadron sinyallerinden belirlenmesi astronomi için önemli konulardan biridir. Veri kümesi UCl adlı çevrim içi makine öğrenim sitesinden alınmıştır. Veri kümesi gama ve hadron olmak üzere iki sınıfa sahiptir. Amaç hadron sinyallerinden gelen gama ışınlarını sınıflandırmaktır. Bu amaç doğrultusunda makine öğrenmesi tekniklerinden k-nearest neigbour ve multilayer percepton kullanılmıştır. Veri kümesinin analizi ve sınıflandırıcının verdiği doğruluk sonucu karşılaştırılmıştır.

Makine öğrenmesi, knn, mlp, sınıflandırma, veri kümesi, doğruluk sonucu

I. GİRİŞ

İnsan sınıflandırmasının yetersiz kaldığı büyük ölçekli verilerin sınıflandırılabilmesi için makine öğrenme teknikleri kullanılmaktadır. Girdi vektörlerinin sonlu sayıda ayrık kategoriye atamayı amaçlayan durumlar sınıflandırma problemi olarak tanımlanmaktadır. Sınıflandırma problemlerinde çıktı uzayındaki eleman sınıf (class), problemi çözen modele de sınıflandırıcı adı verilmektedir. Önemli olan veri kümemiz için en doğru modeli ve yöntemleri bulabilmektir. Sınıflandırma problemlerinde veri kümesi test ve eğitim olmak üzere ayrılmaktadır. Doğru modeli bulmak hem test hem de eğitim verilerinde doğru sonucu bulmak için önemlidir. Bu projede eğitim ve test olarak ayrılan veri üzerinde çeşitli yöntemler uygulanarak en yüksek doğruluk oranı elde edilmeye çalışılmıştır. Test boyutu olarak %20 belirlenmiştir. Grid Search kullanılarak en iyi parametre grubu seçilerek model eğitilmiştir. Seçilen parametreler doğrultusunda çağırılan sınıflandırıcın en yüksek doğruluk değeri accuracy(doğruluk) metriğiyle ölçülmüştür. Bu projenin amacı verilen Magic Gamma Telescope veri kümesi için k-nearest neigbour ve multilayer perceptron sınıflandırıcısı kullanılarak sınıflandırma teknikleri ile doğruluk değerinin belirlenmesidir. Her iki sınıflandırıcı modelinde de genel olarak aşağıdaki adımlar izlenmiştir:

- Problem Tanımı
- Veri Analizi
- Veri Hazırlama (Eğitim ve test gruplarına ayırma)
- Modeli Değerlendirme
- Sonuçların Geliştirilmesi (En iyi parametre seçimi)
- Sonuçların Sunumu

Elde edilen doğruluk sonuçları sonuçlar kısmında karşılaştırılmıştır.

Kullanılan veri kümesi 11 öznitelikten oluşmaktadır. Bunlardan biri de class özelliğidir. Proje içerisinde class özelliği 1(h), 0(g) olarak kullanılmaktadır.

II. VERİ KÜMESİ

Veri kümesine UCl sitesinden ulaşılmış olup(magic04.data) projede kullanılmak üzere .csv (virgüller ayrılmış değerler dosyası) formatına dönüşütürülmüştür.

Veri Seti Özellikleri: Çok Değişkenli
Öznitelik Özellikleri: Gerçek
Veri Sayısı : 19020
Öznitelik Sayısı: 11
Kayıp Değer: Yok

A. Detaylı Bilgi

Veriler yüksek enerjili gama parçacıklarının kaydını simüle etmek için Cherenkov teleskopları tarafından toplanmıştır. Mevcut bilgiler bir düzlemde, kamerada düzenlenmiş fotomultiplier tüplerinde gelen Cherenkov fotonları tarafından bırakılan darbelerden oluşur. Aktif galaktik çekirdekler, süpernova kalıntıları, gama ışını patlamaları ve pulsarlar gibi birçok yerleşik gama ışını kaynağı hakkında hayati bilgi sağlamak için tasarlanmıştır. Görüntüyü oluşturan pikseller, çeşitli görüntü işleme ve özellik çıkarma teknikleriyle hillas parametresi olarak da adlandırılan bazı görüntü parametreleri kümesine dönüştürülür. Bu, istatistiksel olarak olayların ayrılmasına izin verir. Bir gama ışını sinyali teleskopun kamera düzleminde bir elipsi tanımlar Deneyde kullanılan veri kümeleri 10 görüntü parametresi içerir. Atmosferik radyasyonlar nedeniyle, yer tabanlı teleskop, arka plan olarak da adlandırılan hadron ve müonların ezici olaylarını toplar. . Gama ve Hadron olayları arasında zayıf bir ayrım vardır ve bu da verileri sınıflandırma teknikleri için mükemmel bir kanıt haline getirir. . Enerji birikintileri tipik olarak ana eksen boyunca asimetriktir ve bu asimetri ayrımcılıkta da kullanılabilir. Ayrıca, daha fazla ayırt edici özellik vardır. Amaç, birincil gammalar (sinyal, g) tarafından oluşturulan görüntüleri, üst atmosferdeki kozmik ışınlar tarafından başlatılan hadronik duşların görüntülerinden (arka plan, h) istatistiksel olarak ayırmaktır.

B. Öznitelik(Attribute) Bilgileri

Veri kümesinde sınıflandırıcı da dahil olmak üzere 11 tane özellik vardır. Veri tipleri ve kısa açıklamaları aşağıdaki tabloda belirtilmiştir.

Öznitelik	Tip	Açıklama
fLength	Numerical	Elipsin ana ekseni. (mm)
fWidth	Numerical	Elipsin küçük ekseni. (mm)
fSize	Numerical	Tüm piksellerin toplamının 10 tabanında logaritması.
fConc	Numerical	fsize üzerindeki en yüksek iki pikselin toplamının oranı.
fConc1	Numerical	fsize üzerinde en yüksek piksel oranı.
fAsym	Numerical	Ana eksene yansıtılan en yüksek pikselden merkeze mesafe. (mm)
fM3Long	Numerical	Ana eksen boyunca üçüncü anın üçüncü kökü. (mm)
fM3Trans	Numerical	Küçük eksen boyunca üçüncü anın üçüncü kökü. (mm)
fAlpha	Numerical	Ana eksenin vektör ile başlangıç noktasına açısı. (degrees)
fDist	Numerical	Başlangıç noktasından elipsin merkezine olan mesafe. (mm)
class	Binary	Gamma Rays(g) veya Background Hardron Radiation(h) (b)

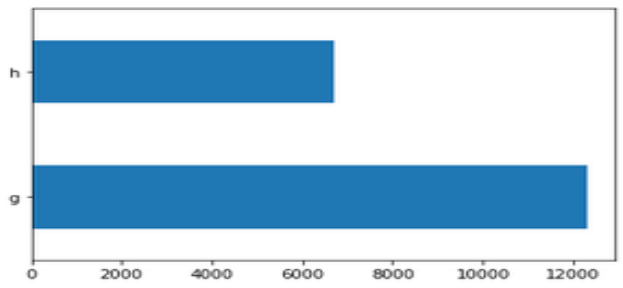
Şekil 1: Veri kümesi özellik tablosu.

C. Sınıf Bilgileri

19020 veri bulunan veri kümesi analiz edildiğinde g sınıfına ait 12332, h sınıfına ait 6688 veri tespit edilmiştir. Sınıflandırma g ve h değerlerine göre yapılmıştır. Sınıf bilgileri aşağıdaki tabloda görüldüğü gibidir.

Sınıf	Sayı
g (gama(sinyal), 0)	12332
h (hadron(arka plan), 1)	6688

Şekil 2: Sınıf bilgilerini içeren tablo.



Şekil 3: Sınıf bilgilerinin oransal olarak görünümü.

III. KULLANILAN TEKNOLOJİLER

Yapılan proje makine öğrenimi teknikleri kulanılarak Jupyter Notebook ortamında Python dili kullanılarak hazırlanmıştır. Kullanılan teknolojilerin detaylı açıklamaları aşağıda belirtilmiştir.

A. Jupyter Notebook

Jupyter Notebook, bir web tarayıcısı üzerinden notebook belgesi formatındaki kodları düzenlemeyi ve çalıştırmayı sağlayan bir sunucu-istemci uygulamasıdır. İlk çıktığında isim olarak IPython Notebook diye biliniyordu. Tümleşik bir python dağıtımı olan Anaconda üzerinden Jupyter Notebook aracına ulaşılabilir. Jupyter Notebook oldukça yararlı bir araçtır. Alınan notları ve hesaplamaları bir arada tutmak için kullanılabilecek bir veri defteridir. Not tutma, hesaplama yapmanın yanı sıra Python kodları yazmaya da olanak verir.

B. Python

Python esas olarak nesne tabanlı programlamayı, belli bir oranda da fonksiyonel programlamayı destekleyen genel amaçlı bir programlama dilidir. 1991 yılında ilk sürümü geliştirilen Python görece kolaylığı, geniş standart kütüphanesi ve dinamik yapısı nedeniyle günümüzde oldukça popülerleşmiş ve yaygın bir kullanıcı kitlesine ulaşmıştır. Özgür ve açık kaynak kod mantığına dayanan Python'ın standart kütüphanesi, geliştirme araçları ve diğer birçok kütüphanesi lisansa gerek duyulmaksızın açık kaynak kod olarak ücretsiz şekilde indirilebilmektedir. Hızlı işlem yapabilme sebebiyle birimsel hesaplamalarda veriler üzerinden işlem yapmada, veri manipülasyonunda, makine öğrenmesi gibi çeşitli yapay zeka uygulamalarında, oyun geliştirmede, web uygulamaları ve ağ programlama sistemlerinde ve nice çeşitli alanlarda kullanılır.

C. Makine Öğrenmesi

Temel olarak, otomatik öğrenme ve geliştirme ilkesine dayandırılır. Makine öğrenmesi, çeşitli algoritmalar ve yöntemler ile veride bazı kalıpları arar ve bu kalıplara karşılık gelen etiketlere bakarak önce öğrenir, daha sonra deneyimlerinden yararlanarak çıkarım yapabilen sistemler geliştirmeye imkan sağlar. Bu imkanı, çeşitli matematiksel ve istatistiksel yöntemlerin kullanıldığı birçok algoritma ile sağlamaktadır. Bu yöntem ve algoritmaların bir veya birkaçı bir arada kullanılarak model(ler) oluşturulur ve bu model(ler), tahmin edilmesi istenilen şeyi, en verimli, en kesin en hızlı biçimde tahminlemeyi amaçlamalıdır. Yazılım programlarının açık bir şekilde programlanmadan sonuçları tahmin etmede daha doğru olmasını sağlayan bir algoritma kategorisidir. Makine öğrenmesinin temel dayanağı, giriş verisini alabilen algoritmalar oluşturmak ve çıktıları yeni veriler ortaya çıktıkça güncellerken bir çıktıyı tahmin etmek için istatistiksel analiz kullanmaktır. Makine öğrenimi algoritmaları genellikle denetlenen veya denetlenmeyen olarak kategorize edilir.

IV. YÖNTEMLER

Veri kümesinin sınıflandırma probleminin çözümünde makine öğrenmesi tekniklerinden yararlanılmıştır. KNN ve MLP sınıflandırıcıları kullanılırken standardizasyon ve model tuning yöntemleri kullanulmıştır. En tutarlı ve yüksek doğruluk sonucuna ulaşmak hedeflenmiştir. Veri .csv formatında kullanıldıktan sonra çeşitli işlemlerden geçirilmiştir. Eğitim ve test olarak ayrılan veri kümesi ilk olarak standardizasyon ön işlemesine girmiştir. GridSearch ile en iyi parametrelere ulaşılmıştır. Bu parametreler ile eğitilen model üzerinden sınıflandırma sonucu elde edilmiştir.

A. KNN (k-yakın komşuluk)

k-en yakın komşu(KNN) algoritması makine öğreniminin ilk günlerinden itibaren temel bir sınıflandırma algoritmasıdır. 1967 yılında önerilmiştir. Hala görüntü analizi, öneri sistemleri, sınıflandırma gibi alanlarda kullanılmaktadır. Mesafeye dayalı bir sınıflandırıcıdır. K-NN algoritmasında, eğitim setinde yer alan örnekler n boyutlu sayısal nitelikler ile belirtilir. Her örnek n boyutlu uzayda bir noktayı temsil edecek biçimde tüm eğitim örnekleri n boyutlu bir örnek uzayında tutulur. Bilinmeyen bir örnek ile karşılaşıldığında, eğitim setinden ilgili örneğe en yakın k tane örnek belirlenerek yeni örneğin sınıf etiketi, k en yakın komşusunun sınıf etiketlerinin çoğunluk oylamasına göre atanır Algoritmaya ait pseudo-code aşağıda veilmiştir.

k-NN Algortiması Pseudo-Code:

Çıktı: test verileri için tahmini etiketler (y)
for Y dizisindeki her i eleman
  Yi'nin tüm eğitim noktalarına olan mesafeyi hesapla #A
  k en küçük mesafelerin indekslerini bul
  bu veri noktalarını sınıflarına göre ayır
  k veri noktalarının çoğunun ait olduğu sınıfı bul #B
  bu sınıfı Yi etiketi olarak ata
end for

A# Uzaklık fonksiyonu
B# Sınıflandırma fonksiyonu

Uzaklık hesapları için genel olarak aşağıda belirtilen fonksiyonlar kullanılmaktadır:

1. "Euclidean" Uzaklık
2. "Manhattan" Uzaklık
3. "Minkowski" Uzaklık

Yapılan projede uzaklık ölçütü olarak Öklid (Euclidean) Uzaklığı kullanılmıştır. Öklid uzaklığı, sınıflandırma ve kümeleme algoritmalarında en sık kullanılan uzaklık ölçütüdür. Öklid uzaklığı, iki nokta arasındaki doğrusal uzaklık olup herhangi iki nokta arasındaki Öklid uzaklığı aşağıda belirtilen formüle göre hesaplanır.

$$\left(\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}\right)$$

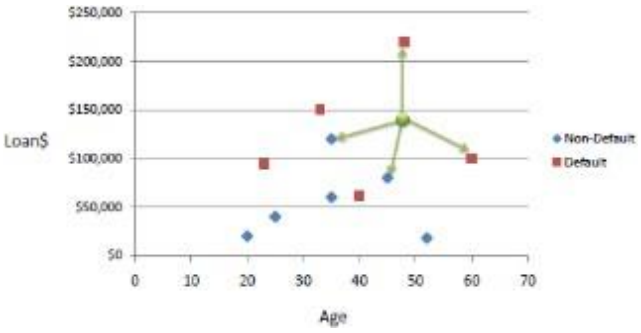
Şekil 4: Öklid uzaklık formülünü gösteren şekil.

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

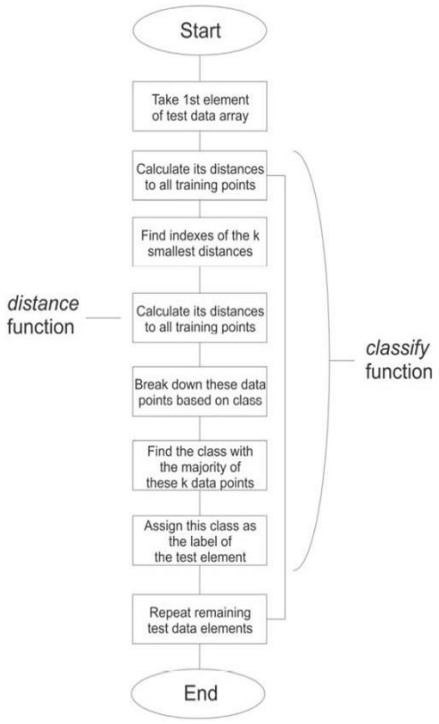
Şekil 5: Uzaklık hesabında kullanılan fonksiyonların uzaklık formüllerini gösteren şekil.



Şekil 6: k-NN algoritmasının görsel olarak örneklenmesi.

KNN yöntemi, sadece tek bir hiperparametrenin (k-komşu değeri) ayarlanmasını gerektiren basit bir yaklaşımdır. Bu çalışmada, K-NN algoritmasının komşu sayısı (k), uzaklık ve ağırlık fonksiyonlarına ilişkin parametrelerinin, sınıflandırma performansını ne ölçüde etkilediği incelenmiştir. k değerinin uygun değerde seçilmesi oldukça önem taşımaktadır. k değeri büyüdükçe daha düzgün karar sınırları oluşmasına karşın hesaplama yükü artacaktır. Uzaklık fonksiyonları da örneklerin dağılımına uygun olarak seçilmelidir.

Projede k için en uygun değeri bulmak için, eğitim setinde 10 set çapraz değerleme kullanıyoruz ve toplam verilerin %20'sini test seti olarak tutuyoruz. Çapraz değerleme(cross validation) öncesi 1-20 aralığında aranan hiper-parametre değeri olarak k=14 bulunmuştur. Bulunan hiper-parametre değeri ve öklid uzaklığı seçilerek yeniden düzenlenen algoritma en yüksek doğruluk sonucunu vermiştir.



Şekil 7: k-NN algoritması akış şeması.

B. MLP (Multilayer Perceptron)

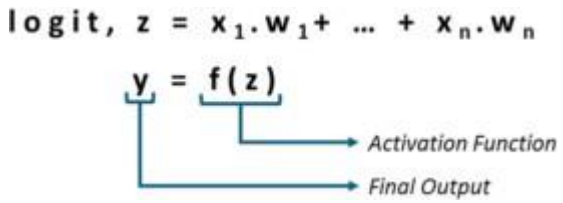
Son yıllarda klasik tekniklerle çözülemeyen sınıflandırma problemlerinin çözümünde sinir ağları uygulamasına ilgi artmıştır. Yapay Sinir Ağları (ANNs) veya bağlantıcı sistemler, hayvan beyinleri oluşturan biyolojik sinir ağlarından esinlenen bilgi işlem sistemleridir. Bu tür sistemler, genellikle göreve özgü programlama olmadan örnekleri göz önünde bulundurarak görevleri yapmayı öğrenir (kademeli olarak performansı iyileştirir). Sinir ağları kullanılarak çözülebilen her türlü sınıflandırma problemini iki geniş kategoriye ayırabiliriz:

- Doğrusal Olarak Ayrılabilir Problemler
- Doğrusal Olmayan Ayrılabilir Problemler

Temel olarak, veri setin tek bir satır kullanarak iki kategoride sınıflandırabiliyorsa, bu problemin doğrusal olarak ayrılabilir olduğu söylenir. Bir perceptron lineer bir sınıflandırıcı olarak çalışmaktadır. Perceptronlar, son derece sınırlı olmalarına karşın en eski sinir ağlarından biridir. Perceptron, bir sinir hücresinin birden fazla girdiyi alarak bir çıktı üretmesi prensibine dayanır.

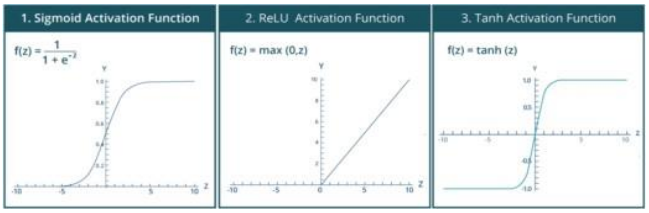
Aktivasyon Fonksiyonları

Yapay sinir ağlarına doğrusal olmayan gerçek dünya özelliklerini tanıtmak için aktivasyon fonksiyonuna ihtiyaç duyarız. Aktivasyon fonksiyonu aşağıdaki resimde gösterildiği gibi bir perceptronun çıkışına uygulanır.



Şekil 8: Perceptron çıkışına uygulanan aktivasyon fonksiyonu.

Lineer sınıflandırma yapılan problemlerde relu aktivasyon fonksiyonu kullanılabilir. Ancak gerçekleştirilmek istenen sınıflandırma doğası gereği doğrusal değilse, doğrusal olmayan aktivasyon fonksiyonlarından biri kullanılmalıdır. Belirgin doğrusal olmayan aktivasyon fonksiyonlarının bazıları aşağıda gösterilmiştir:

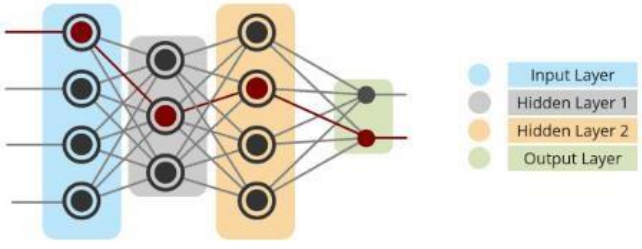


Şekil 9: Sigmoid, relu ve tanh aktivasyon fonksiyonlarının gösterimi.

Tek Katmanlı Perceptron

- Tek katmanlı perceptronlar, doğrusal olmayan ayrılabilir veri noktalarını sınıflandıramaz.
- Çok katmanlı parametreleri içeren karmaşık problemler, tek katmanlı perceptronlarla çözülemez.
- Tek kamanlı perceptronlar, lineer olmayan ayrılabilir veri noktalarını sınıflandıramaz.

Çok Katmanlı Perceptron (Multilayer Perceptron)
Beynimiz milyonlarca nörondan oluşmaktadır. Bir sinir ağı da tıpkı bunun gibi farklı yollarla bağlanan ve farklı aktivasyon fonksiyonlarında çalışan birçok perceptronun birleşimidir. Tek katmanlı algılayıcıların doğrusal olmayan problemlerin çözümünde başarısız olmasının üzerine geliştirilen çok katmanlı algılayıcılar (MLP), bilgi girişinin yapıldığı girdi katmanı, bir veya daha fazla gizli (ara) katman ve bir çıktı katmanından oluşmaktadır. ÇKA'da katmanlar arası ileri ve geri yayılım olarak adlandırılan geçişler bulunur. İleri yayılım safhasında, ağın çıktısı ve hata değeri hesaplanır. Geri yayılım safhasında ise hesaplanan hata değerinin minimize edilmesi için katmanlar arası bağlantı ağırlık değerleri güncellenir.



Şekil 10: MLP Yapısı

- Girdi Düğümleri: Giriş düğümleri dış dünyadan ağa bilgi sağlar ve birlikte "Giriş Katmanı" olarak adlandırılır. Hesaplama yapmaz gizli düğümlere bilgi aktarımı yaparlar.
- Gizli Düğümler: Gizli düğümlerin dış dünyayla doğrudan bağlantısı yoktur. Hesaplamalar yaparlar ve girdi düğümlerinden çıkış düğümlerine bilgi aktarırlar. Gizli düğümlerden oluşan bir koleksiyon "gizli katman" oluşturur. Bir ağ sadece tek bir giriş katmanına ve tek bir çıktı katmanına sahipken, sıfır veya çoklu gizli katmanlara sahip olabilir. Çok katmanlı bir perceptronun bir veya daha fazla gizli katmanı vardır.
- Çıktı Düğümleri: Çıktı düğümleri toplu olarak "Çıktı Katmanı" olarak adlandırılır. Ağdan dış dünyaya bilgi aktarımından sorumludur.

Çok katmanlı ağ öğretmenli öğrenme stratejisini kullanır. Ağa, hem örnekler hem de örneklerden elde edilmesi gereken çıktılar verilmektedir. Sistem, kendisine gösterilen örneklerden genellemeler yaparak problem uzayını temsil eden bir çözüm uzayı üretmektedir. Daha sonra gösterilen benzer örnekler için bu çözüm uzayı sonuçlar ve çözümler üretebilmektedir.

Çok Katmanlı Ağın Çalışma Şekli

- Örneklerin toplanması: Ağın çözmesi istenen olay için daha önce gerçekleşmiş örneklerin bulunması adımıdır.
- Ağın topolojik yapısının belirlenmesi: Öğrenilmesi istenen olay için oluşturulacak olan ağın yapısı belirlenir. Kaç tane girdi ünitesi, kaç tane ara katman, her ara katmanda kaç tane hücre elemanı ve kaç tane çıktı elemanı olması gerektiği belirlenmektedir
- Öğrenme parametrelerinin belirlenmesi: Ağın öğrenme katsayısı, proses elemanlarının toplama ve aktivasyon fonksiyonları, momentum katsayısı gibi parametreler bu adımda belirlenmektedir.
- Ağın başlangıç değerlerinin atanması: Hücre elemanlarını bir birine bağlayan ağırlık değerlerinin ve eşik değere başlangıç değerinin atanması
- Öğrenme setinden örneklerin seçilmesi ve ağa gösterilmesi: Ağın öğrenmeye başlaması.
- Öğrenme sırasında ileri hesaplamaların yapılması: Verilen girdi için ağın çıktı değerinin hesaplanması.
- Gerçekleşen çıktının beklenen çıktı ile karşılaştırılması: Ağın ürettiği hata değerlerinin hesaplanması.
- Ağırlıkların değiştirilmesi: Geri hesaplama yöntemi uygulanarak üretilen hatanın azalması için ağırlıkların değiştirilmesi.

En İyi Parametreleri Bulma

En iyi parametreleri bulmak için, parametrelere farklı değerler kullanarak çeşitli testler gerçekleştirmek gerekir. GridSearch yöntemi kullanılarak yapılan projede multilayer perceptron sınıflandırıcı için en iyi parametre bulma denemeleri yapılmıştır. Verilen aktivasyon fonksiyonları ve gizli katman sayıları sonucu en iyi parametre olarak algoritma 'alpha':0.03, 'hidden_layer_sizes': (10, 10) değerlerini seçmiştir. Aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılmıştır.

C. Standardizasyon

Veri okuma işleminden sonra veri kümesi istediğimiz gibi olmayabilir. Verinin biçimi, eksik veriler gibi etkenler veri kümesinin bozulmasına yol açar. Verilerin dağılımı göz önüne alındığında, veri kümesindeki her bir değerin çıkarılan ortalama değeri olacaktır ve daha sonra tüm veri kümesinin standart sapmasına bölünecektir. Bu da modeli eğitme aşamasında istemediğimiz sonuçlar doğurabilir. Bu durumu önlemek amacıyla uygulanan bu ön işleme standardizasyon denir. Verinin dağılımı dikkate alınarak düzenleme işlemi gerçekleştirilir. Özellikler ortak bir veri alanına çekilir. Ortalama değeri 0, standart sapma değeri ise 1 olur. Standardizasyon işlemi sonucu kullanılan veri kümesi daha normal ve baskınlığı azalmış bir hal almıştır. Standardizasyon hata payını azaltmaktadır.

D. Model Tuning

GridSearchCV model üzerinde kombinasyonları deneyerek en iyi parametreleri seçer. Bu parametreler hiper-parametre olarak adlandırılmaktadır. Bu hiper-prametreler probleme ve veri kümesine göre değişiklik göstermektedir. Modelin performansını artırmak hedeflenmektedir. Normalde kullanılan sınıflandırıcı teknikleri model için gerekli olan parametreleri kendi içinde tanımlı olarak getirmektedir.

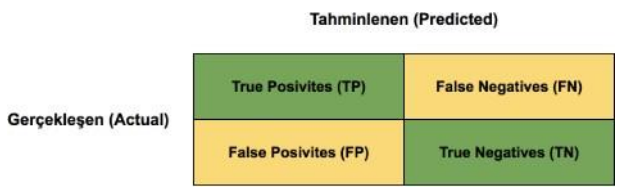
Model Tuning işleminde; Grid Seach yöntemiyle veriler taranarak bulunan en uygun parametreler kullanılarak modelin performansı artırılır.

Yapılan sınıflandırma problemlerinin çözümü projesinde kullanılan sınıflandırıcı modellerinde bu işlem uygulanmıştır. Bulunan hiper-parametreler ile eğitilen modelin doğruluk oranında artış olduğu gözlenmiştir.

V.DEĞERLENDİRME

Kullanılan sınıflandırıcı modellerinin performanslarının değerlendirilebilmesi için çeşitli değerlendirme ölçütleri bulunmaktadır. Yapılan projede accuracy (doğruluk) metriği kullanılarak modellerin tahmin performansı ölçülmüş ve karşılaştırılması yapılmıştır. Accuracy değeri ve diğer birkaç değer hesaplanırken karmaşıklık matrisi baz alınmaktadır.

Karmaşıklık matrisi tahminlerin doğruluğu hakkında bilgi veren bir ölçüm aracıdır. Ölçümün doğruluğu hakkında anlaşılması kolay bilgiler sağladığı için özellikle sınıflandırma algoritmalarında sıklıkla kullanılıyor.



Şekil 11: Karmaşıklık matrisi (confusion matrix) açıklama şekili.

Şekil 11'de gösterilen karmaşıklık matrisi açıklama şekline göre matrisin her bir elemanının özel bir anlamı bulunmaktadır.

- True Positive (TP) : Algoritma çıktısı ile asıl durum evet.
- False Negative (FN) : Algoritma çıktısı hayır fakat asıl durum evet.
- False Positives (FP) : Algoritma çıktısı evet fakat asıl durum hayır.
- True Negatives (TN) : Algoritma çıktısı ve asıl durum hayır.

Modele ait olan karmaşıklık matrisi ile accuracy (doğruluk) oranı hesaplanmaktadır. Model değerlendirilirken tüm bunlar göz önüne alınarak değerlendirilmiştir.

$$\frac{TP + TN}{TP + FP + TN + FN}$$

Şekil 12: Accuracy(Doğruluk) değeri formülü.

Yukarıda Şekil 12 ile verilen formül ile modelin accuracy değeri hesaplanmaktadır. Modelde doğru tahmin edilen alanların toplam veri kümesine oranıdır.

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

Şekil 13: Precision (Kesinlik) değerinin formülü.

Şekil 13'de gösterilen precision formülü ise positive olarak modelin tahmin ettiği değerlerin kaç tanesinin doğru olarak tahmin edildiğini göstermektedir. Model seçiminde önemli bir kriterdir.

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

Şekil 14: Recall değerinin formülü

Şekil 14 ile formülü gösterilen recall değeri de değerlendirme aşamasında kullanılan önemli ölçütlerden biridir. Recall; positive olarak tahmin edilmesi gereken işlemlerin ne kadarının positive olarak tahmin edildiğini gösteren bir metriktir.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Şekil 15: F1-score değerinin formülü

Şekil 15'de gösterilen F1 Score değeri de precision ve recall değerlerinin harmonik ortalamasını göstermektedir.

Yapılan projede yukarıda verilen ölçümler ve accuracy değeri kullanılmıştır. Modellerin karşılaştırılması accuracy (doğruluk) metriği üzerinden yapılmıştır.

VI. SONUÇLAR

A. KNN Sınıflandırıcısı Sonucu:

KNN sınıflandırıcısı k=5 değeri ile çalıştırıldığında elde edilen doğruluk sonucu şekil 16'da gösterilmiştir. Best Parametre seçiminden sonra elde edilen en iyi k değeri k=14 olarak seçilmiştir.

k-nearest neighbour varsayılan parametreler:
(algorithm='auto'
, metric='minkowski',
n_neighbors=5,
weights='uniform')

k-nearest neighbour seçilen en iyi parametreler:
{'metric': 'euclidean',
'n_neighbors': 14,
'weights': 'distance'}

K=14 karmaşıklık matrisi:
[[848 490]
[101 2365]]

k Değeri	Doğruluk (Accuracy) Değeri
k = 5 Test Size:0.20	0.832018927444795
k = 14 Test Size:0.20 (best parameters)	0.8446372239747634

Şekil 16: k değerlerine göre doğruluk sonucu tablosu.

Class	precision	recall	f1-score	support
0 (h)	0.89	0.63	0.74	1338
1 (g)	0.83	0.96	0.89	2466

Şekil 17: k= 14 , k-NN classification report.

B. MLP Sınıflandırıcısı Sonucu:

Seçilen en iyi parametreler ile MLP sınıflandırıcısının performansı iyileştirilmiş ve en yüksek accuracy değeri elde edilmiştir.

MLP varsayılan parametre değerleri:
MLPClassifier(activation='relu', alpha=0.0001,
batch_size='auto', beta_1=0.9, beta_2=0.999,
early_stopping=False, epsilon=1e-08,
hidden_layer_sizes=(100,), learning_rate='constant',
learning_rate_init=0.001, max_fun=15000, max_iter=200,
momentum=0.9, n_iter_no_change=10,
nesterovs_momentum=True, power_t=0.5,
random_state=None, shuffle=True, solver='adam',
tol=0.0001, validation_fraction=0.1, verbose=False,
warm_start=False)

MLP seçilen en iyi parametre değerleri:
{'alpha': 0.03, 'hidden_layer_sizes': (10, 10)}

```
#Multilayer perceptron classifier find the best parameters
mlp_params= {"alpha": [1,5,0.1,0.01,0.03,0.005,0.0001],
             "hidden_layer_sizes": [(10,10),(100,100,100),(100,100),(3,5)]}
```

Şekil 18: En iyi parametre seçiminde kullanılan değerler.

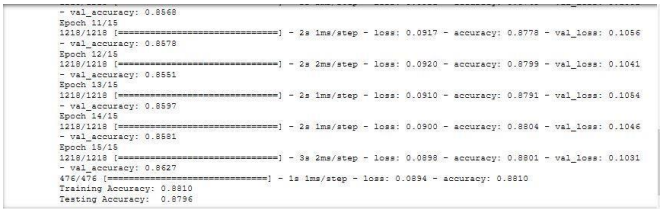
MLP Best Parameters Karmaşıklık Matrisi:
[[1030 308]
[144 2322]]

	Accuracy (Doğruluk) Sonucu
MLP Varsayılan Parametreler Test Size:0.20	0.8772344900105152
MLP Seçilen En İyi Parametreler Test Size:0.20	0.8811777076761304

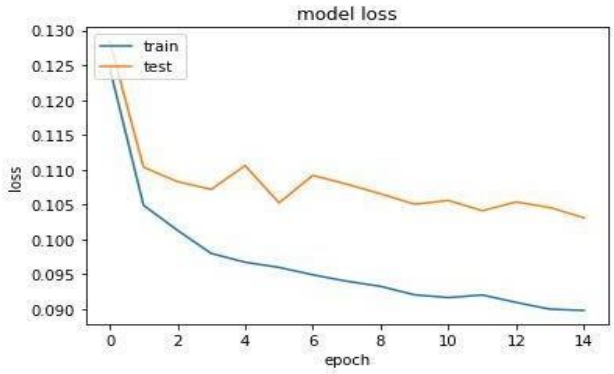
Şekil 19: MLP Accuracy Değerleri Tablosu

Class	precision	recall	f1-score	support
0 (h)	0.89	0.76	0.82	1338
1 (g)	0.88	0.95	0.91	2466

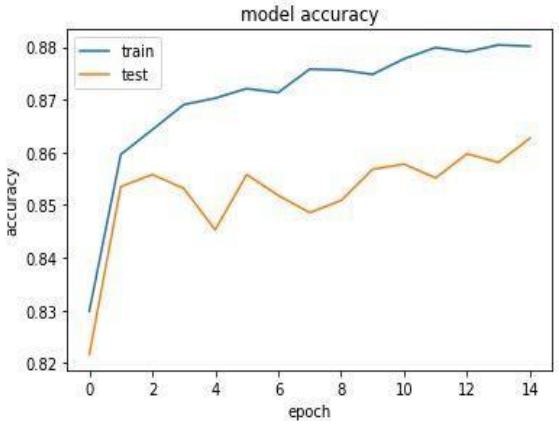
Şekil 20: Mlp best parameters classification report.



Şekil 21: Epoch sonuçlarını gösterek kod sonucu.



Şekil 22: Loss-Epoch Grafiği.(batch_size=10, epochs=15)



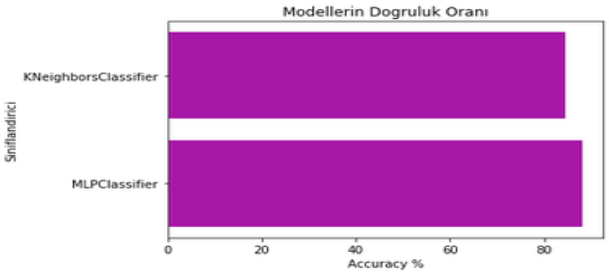
Şekil 23: Accuracy-Epoch Grafiği. .(batch_size=10, epochs=15)

C. Sonuç

Proje de Magic Gamma Telescope veri kümesi k-nearest neighbour ve multilayer perceptron algoritması kullanılarak sınıflandırma problem çözülmüştür. Kullanılan algoritmalar seçilen parametrelere ve tekniklere göre farklı doğruluk sonuçları vermektedir. Elde edilen accuracy (doğruluk) değerleri değerlendirme ölçütü olarak baz alındığında en yüksek başarı oranı ile en uygun sınıflandırıcı algoritmasının multilayer perceptron olduğu anlaşılmıştır.

Sınıflandırıcı	Accuracy (Doğruluk) Sonucu
K-NN	0.84
MLP	0.88

Şekil 24: k-NN ve MLP Accuracy(Doğruluk) metriği ile elde edilen değerlerinin karşılaştırılması



Şekil 25: KNN ve MLP doğruluk değerlerinin grafik üzerinden karşılaştırılması.

KAYNAKLAR

[1] https://archive.ics.uci.edu/ml/datasets/magic+gamma+telescope

[2] https://www.kaggle.com/vitorgamalemos/neural-network-02-multilayer-perceptron/

[3] http://rasbt.github.io/mlxtend/user_guide/classifier/MultiLayerPerceptron/#example-1-classifying-iris-flowers.

[4] http://rasbt.github.io/mlxtend/user_guide/classifier/MultiLayerPerceptron/

[5] https://medium.com/@ayyucekizrak/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-cee17fd1d9cd

[6] https://github.com/ayseilkay/Python_Data_Science_Udemy_Course_Notes/tree/614b7b20781c6aaeb7180f9aefe0803e30fc6262

[7] https://www.veribilimiokulu.com/veri-hazirliginin-vazgecilmezi-ozellik-olceklendirme/

[8] https://gelecegiyazanlar.turkcell.com.tr/konu/makine-ogrenmesi/egitim/makine-ogrenmesi-101/model-tuning

[9] https://scikit-learn.org/stable/modules/neural_networks_supervised.html#classification

[10] https://devhunteryz.wordpress.com/2018/06/30/derin-ogrenme-perceptron-ogrenme-algoritmasi/

[11] https://devhunteryz.wordpress.com/2018/07/05/yapay-sinir-agi-egitimi-cok-katmanli-perceptronmulti-layer-perceptron/

[12] https://www.harita.gov.tr/images/dergi/makaleler/130_3.pdf

[13] https://ekblc.files.wordpress.com/2013/09/mlp.pdf

[14] http://library.beykoz.edu.tr/wp-content/uploads/YAPAY-S%C4%B0N%C4%B0R-A%C4%9ELARI-Y%C3%96NTEM%C4%B0-%C4%B0LE-ARALIKLI.pdf

[15] https://qastack.info.tr/datascience/21877/how-to-use-the-output-of-gridsearch

[16] https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-model-dogrulama-ve-hiper-parametre-secim-yontemleri-823812d95f3

[17] http://sebastianraschka.com/Articles/2014_about_feature_scaling.html#standardization-and-min-max-scaling

[18] https://medium.com/@gulcanogundur/do%C4%9Fruluk-accuracy-kesinlik-precision-duyarl%C4%B1l%C4%B1k-recall-ya-da-f1-score-300c925feb38

[19] http://bilgisayarkavramlari.sadievrenseker.com/2008/11/17/knn-k-nearest-neighborhood-en-yakin-k-komsu/

[20] https://womaneng.com/cross-validation-nedir/