

Parkinson Hastalığının Tanınması İçin Veri Kümesi Kullanarak Çeşitli İkili Sınıflandırma Makine Öğrenimi Yöntemlerinin Uygulanması ve Performansı

Ayşenur Eskin

Özet

Parkinson hastalığı (PD), beynin sinirsel, davranışsal ve fizyolojik sistemlerini etkileyen nörodejeneratif bir hastalıktır. Hareketlerde yavaşlama, kas sertliği, titreme ve dengesizlik gibi motor semptomlara ek olarak Alzheimer hastalığı (AD), psikiyatrik sorunlar, uykusuzluk, anksiyete ve duyuşsal anormallikler gibi diğerk problemler de ortaya çıkarabilir. PD, dünyada en sık görülen ikinci nörodejeneratif hastalıktır. Dünyada 10 milyon, Türkiye'de yaklaşık 100-130 bin kişinin Parkinson hastası olduğı varsayılmaktadır. Kesin bir tanı yönteminin olmaması nedeniyle ve hastalarda belirtiler zamanla yavaş bir şekilde ortaya çıkar ve erken aşamalarda tespit edilmesi oldukça zordur. PD'nin erken ve doğru tanısı, zamanında müdahale ve kişiselleştirilmiş tedavi için çok önemlidir. Makine öğrenimi algoritmaları hastalıkları tahmin etmek ve teşhis etmek için bir araç olarak kullanılmıştır. Bu çalışmada amacımız, yüksek doğruluk, duyarlılık, özgüllük ve yorumlana bilirlikle çeşitli veri kaynaklarından Parkinson hastalığını tespit edebilen Makine Öğrenmesi (ML) ve Derin Öğrenme (DL) tabanlı bir sistem oluşturmaktır. Araştırmamızda, Parkinson hastalığı teşhisi konmuş 2105 hastanın kapsamlı sağlık bilgilerinin içeren veri kümesinde K-En Yakın Komşu (KNN), Destek Vektör Makinesi (SVM), Yapay Sinir Ağı, (ANN), Karar Ağaçları (DT), Topluluk Öğrenmesi makine öğrenimi algoritmaları sınıflandırmak için kullanılmıştır. Duyarlılık, özgüllük, doğruluk ve alıcı işletim karakteristik eğrisi altındaki alan (AUC-ROC) gibi performans ölçütleri, model performansını titizlikle değerlendirmek için kullandık. Özellikle, Destek Vektör Makinesi en iyi performansı gösteren makine olarak ortaya çıktı.

Motor veya motor olmayan belirtilerle karakterize nörodejeneratif bir karmaşa olan Parkinson hastalığı (PD), doğru tahmin ve erken tespit açısından benzersiz zorluklar sunar. Bu çalışmada, özellikle Parkinson hastalığı bağlamında hastalık tahminine odaklanarak, sağlık hizmetlerindeki ML algoritmalarının manzarasını daha derinlemesine incelemeyi amaçlıyoruz. Bu zorlukları ele almak için, doğruluk ve performans değerlendirmesine büyük önem vererek ikili sınıflandırma için ML yöntemlerinin kapsamlı bir analizini öneriyoruz. Çeşitli ML algoritmalarını sistematik olarak karşılaştırarak ve zıtlıştırarak, PD tespitine göre uyarlanmış bir tahmin sistemi geliştirmek için en uygun yaklaşımı belirlemeyi amaçlıyoruz. Kapsamlı hedefimiz, yanlış tahminlerin ve tanı yanlışlıklarının oluşumunu azaltırken PD riski taşıyan bireyleri doğru bir şekilde belirlemek için tutarlı bir yol oluşturmaktır.

I- Giriş

İnsan beyni, insan vücudunun ana hesaplama birimidir ve vücudun herhangi bir bölümünde küçük bir kazanın meydana gelmesi, doğrudan diğerk organları etkiler. Bu etkilerden biri de sessiz bir şekilde ortaya çıkan Parkinson hastalığıdır (PD) [1]. Parkinson hastalığı (PD), Nigrostriatal yolakta dopaminerjik nöronların dejenerasyonuna ve striatumda dopamin salınımının azalmasına neden olan yaygın bir nörodejeneratif bozukluktur [2]. Parkinson hastalığı, dopamin seviyelerindeki azalmaya yol açan nöronların ölümü nedeniyle ortaya çıkar. Düşük dopamin seviyeleri, sinapslar arasındaki iletişimi engeller ve motor işlevlerin etkisiz

hale gelmesine yol açar [3]. Parkinson, Alzheimer hastalığından sonra dünya çapında en kötü ikinci nörodejeneratif hastalıktır [4]. Parkinson Vakfı tarafından hazırlanan bir rapora göre, dünya genelinde yaklaşık 10 milyon insan Parkinson hastalığı (PD) ile yaşıyor ve bunlardan yaklaşık bir milyonu Amerika Birleşik Devletleri'ndedir. Bu rapor ayrıca erkeklerin, kadınlara kıyasla Parkinson hastalığından 1,5 kat daha fazla etkilendiğini belirtmektedir [5]. Bu hastalık çoğunlukla 60 yaş üstü kişileri etkiler ve vakaların yalnızca %4'ü 50 yaşın altındaki kişilerde görülür [6]. Bu hastalık için kesin bir tanı yoktur ve tanı büyük ölçüde hastanın tıbbi geçmişine bağlıdır [7]. Hastalığın, koku alma duyusunun kaybı veya azalması, uyku bozuklukları ve kabızlık, titreme ve hareketlerin yavaşlaması gibi motor semptomlardan yıllar önce başladığı düşünülmektedir [8]. Parkinson hastalığının bir tedavisi yoktur; bu nedenle Parkinson hastaları (PD), hastalığın ilerleyişini yavaşlatmak için erken teşhis ve kişiselleştirilmiş tedavilere güvenir [9]. Patolojik olarak doğrulanmış Parkinson hastalığının (PD) yalnızca %80'inde ilk ziyarette doğru klinik tanının doğru olduğu tahmin edilmektedir [10]. Bu nedenle, araştırmacılar hastalığın ilerlemesini durdurmak için hastalığın erken döneminde ortaya çıkan bu motor olmayan semptomları mümkün olduğunca erken tanımanın yollarını aramaktadır [8]. Makine öğrenimi (ML), uygulama kolaylığı ve yüksek doğruluğu nedeniyle son zamanlarda tıbbi hastalık teşhisi için sıklıkla kullanılmaktadır [11]. ML, literatürde PD tedavisi için de kullanılmıştır [12]. Parkinson Hastalığı'nın (PD) tespiti için farklı özellik çıkarma yöntemleri ve makine öğrenimi yöntemleri kullanarak denekleri erken PD ve sağlıklı normal olarak sınıflandıran çalışmalar mevcuttur. Araştırmalar arasında değerlendirilen sınıflandırıcılar arasında K-NN (K-En Yakın Komşu), Destek Vektör Makineleri (SVM), Karar Ağaçları (DT), Yapay Sinir Ağları (ANN), Topluluk Sınıflandırıcı yer almaktadır.

K-En Yakın Komşu (KNN) algoritması, basitliği, çok yönlülüğü ve parametrik olmayan yapısıyla değer verilen, makine öğreniminde temel ancak etkili bir araç olarak öne çıkar. Hem sınıflandırma hem de regresyon görevleri için uygun olsa da baskın uygulaması sınıflandırma tahmininde yatmaktadır. Yakınlık ilkesine göre çalışan KNN, yeni veri noktalarını daha önce eğitilmiş örneklerle hizalayarak kategorilere ayırır ve böylece bunları tutarlı kümeler veya segmentler halinde düzenler. Bu süreç, yeni gözlem ile mevcut veri kümesi arasındaki benzerlik varsayımına dayanır ve ilkini ikincisine en çok benzeyen sınıflandırmaya atar. Yakınlığa öncelik veren algoritma, giriş verilerini komşu örneklerle benzerliklerine göre düzenler ve böylece sınıflandırmalarını belirler. KNN'nin sınıflandırma doğruluğunu ve performansını korurken büyük veri kümelerini işlemedeki ustalığı dikkate değerdir [13]. Mounika ve Govinda Rao [14], Parkinson hastalığının erken teşhisi için çeşitli makine öğrenimi ve derin öğrenme yöntemlerini karşılaştırmış ve en yüksek doğruluk oranını %97,43 ile $k=5$ parametresiyle K-En Yakın Komşu (KNN) algoritmasında elde etmişlerdir.

Destek Vektör Makinesi (SVM) modeli, VC boyutunun ve ampirik risk minimizasyon kavramının araştırılmasına odaklanan, hesaplamalı ve istatistiksel ilkelere dayanan güçlü bir makine öğrenimi yaklaşımını temsil eder. Bu metodoloji, özellikle sınırlı örneklem büyüklükleri, veri heterojenliği ve hesaplamalı karmaşıklıklar içeren senaryolarda, desen tanıma görevleriyle ilişkili zorlukları ele almada belirgin avantajlar sunar. Özellikle, SVM, "boyutluluk laneti" gibi sorunları etkili bir şekilde ortadan kaldırır ve çeşitli karmaşıklıklara karşı sağlamlık sergileyerek "aşırı öğrenme" risklerini azaltır. Sağlam bir teorik temel ve basit bir matematiksel çerçeve tarafından desteklenen SVM, desen tanıma, regresyon analizi, fonksiyon tahmini, zaman serisi tahmini ve ilgili alanlarda önemli ilerlemeler sağlamıştır [15].

Zamanla, çeşitli araştırmacılar tarafından PD'yi teşhis etmek için çeşitli girişimlerde bulunuldu [16]. Konuşma analizi ve titremeler de PD'yi teşhis etmek için kullanılan önemli risk faktörleridir [17]. Bunlardan Benba A. ve çalışma arkadaşlarının [18] konuşma kaydı veri setinden PD'yi teşhis etmek için yapılan önemli çalışmalar incelendi. Bu çalışma arkadaşlarının [18] ses bozuklukları kullanarak PD hastalarını teşhis etmek için bir yöntem oluşturma çalışmalarında destek vektör makinesi (SVM) sınıflandırıcı algoritması kullanarak model %91,4'lük bir doğruluk oranına ulaştı. Ayrıca, Sakar ve arkadaşları [18] bir PD teşhis cihazı tasarladılar ve karşılıklı bilgileri SVM'de birleştirerek %92,75'lik bir sınıflandırma doğruluğu elde ettiler. [19]

Karar ağaçları, denetimli bir öğrenme yöntemi olup, hem sınıflandırma hem de regresyon problemlerini çözebilen çok yönlü bir araçtır; ancak, esas olarak sınıflandırma senaryolarında kullanılır. Bu ağaç benzeri yapıda, iç düğümler veri kümesinin öz niteliklerini, dallar karar sürecini, yaprak düğümler ise nihai sonuçları temsil eder. Karar Ağacı'nın temel bileşenleri, birden fazla dala ayrılan önemli kararlar için karar düğümleri ve daha fazla dallanmaya sahip olmadan belirli kararların sonuçlarını temsil eden yaprak düğümleridir. Veri kümesinin özelliklerinden yararlanarak, Karar Ağaçları karar alma süreçlerini, deneyleri ve testleri mümkün kılar. Grafiksel bir temsil olarak işlev gören Karar Ağaçları, belirli bir problem için çeşitli yolları sistematik bir şekilde keşfederek olası çözümlere ulaşmayı sağlar. [20]. Sumit ve arkadaşları [21], karar ağaçları algoritması eğitim veri kümesinde %100'lük dikkate değer bir doğruluk elde ettiler. Bu da verilerdeki karmaşık desenleri yakalamadaki ustalığını gösterir. Ancak, test veri kümesindeki performansı, %61,53'lük bir doğrulukla nispeten daha düşük elde etmişlerdir.

Derin öğrenme, insan beyninin yapısını ve işleyişini taklit eden yapay sinir ağları (ANN) [22] oluşturmayı içerir. Birbirine bağlı düğüm veya nöron katmanlarından oluşan bu sinir ağları, verilerdeki karmaşık desenleri ve ilişkileri tanımda mükemmeldir. Parkinson Hastalığı'nın vokal sinyalleri, Pahuja ve Nagabhushan [23] tarafından ANN (Yapay Sinir Ağları), K-NN (K-En Yakın Komşu) ve SVM (Destek Vektör Makineleri) kullanılarak değerlendirildi. Değerlendirme sonuçları, ANN'nin %95,89 genel doğruluk oranıyla en doğru performansı sergilediğini gösterdi.

Aditi Govindu ve Sushila Palwe [9], çalışmalarında Rastgele Orman sınıflandırıcı modeli %91,83 tespit doğruluğuna ve 0,95 duyarlılığa ulaşarak, Rastgele Orman sınıflandırıcısını PD'nin tespiti için ideal Makine Öğrenimi (ML) tekniği olarak ortaya koydu.

Bu makalede, parkinson hastalığının ayrıntılı sağlık verilerinden oluşan veri seti üzerinde farklı özellik çıkarma yöntemleri kullanarak, erken PD ve sağlıklı normal olarak denekleri sınıflandırmak için KNN, SVM, DT, ANN ve Ensemble sınıflandırıcılarını kullanarak tanı modelleri geliştirdik. Bu, PD'nin erken teşhisine yardımcı olabilir.

II- Malzemeler ve Yöntemler

Bu bölüm, çalışmamızın uygulanmasında kullanılan veri setini ve teknikleri açıklayacaktır.

A. PD Veri Seti Tanımı

Bu çalışmada kullanılan Parkinson veri seti, Kaggle'da [24] halka açık olarak erişilebilirdir. Bu veri seti, her biri 3058 ila 5162 arasında değişen kimliklerle benzersiz bir şekilde tanımlanan Parkinson Hastalığı teşhisi konmuş 2.105 hastaya ait kapsamlı sağlık bilgilerini içerir. Hiçbir

değişkende kayıp veri yoktur. Veri seti demografik ayrıntıları, yaşam tarzı faktörlerini, tıbbi geçmişi, klinik ölçümleri, bilişsel ve işlevsel değerlendirmeleri, semptomları ve bir tanı göstergesini içerir. Bu veri seti, Parkinson Hastalığı ile ilişkili faktörleri keşfetmeyi, öngörücü modeller geliştirmeyi ve istatistiksel analizler yürütmeyi amaçlayan araştırmacılar ve veri bilimcileri için değerlidir. Veri setindeki Parkinson Hastalığı için tanı durumu; 0 Hayır, 1 Evet anlamına gelir. Bu veri setinde “1” etiketine sahip 1304, “0” etiketine sahip 801 sütun bulunmaktadır. Veri setindeki deneklerin yaş ortalaması 69.90, standart sapması 11.59’dur. Cinsiyet ortalaması 0.49, standart sapması 0.50’dir. Bu veri seti 17 kategorik değişken ve 15 sürekli değişken bulunmaktadır.

Örneklem büyüklüğü satır sayısına (gözlem sayısına), değişken sayısı ise sütun sayısına (özellik + etiket sayısına) eşittir.

- Örneklem büyüklüğü (n): 2105
- Değişken sayısı (p): 35

B. ML Tabanlı Parkinson Hastalığı Tanı Algoritmaları

Bu çalışmada, sağlıklı bireyler ile Parkinson hastalarının belirlenmesi için PD veri seti KNN, SVM, DT, ANN ve Ensemble sınıflandırıcılarını kullanarak ayrı ayrı değerlendirilmiştir.

• Normal Dağılıma Uygunluk Testleri

Normal dağılım istatistiksel birtakım analizleri yapmadan önce uygulanan ve hangi analizi yapmamıza karar veren yardımcı bir analizdir. Verilerin normal dağılıma uygun olup olmadığını ortaya koymak amacıyla çeşitli normallik testlerinden yararlanmak mümkündür. Bu testler arasında Kolmogorov Smirnov normallik test istatistiği değeri, p değeri olarak adlandırılan bir değerle karşılaştırılır. P değeri, veri setinin normal dağılıma uygunluğunu ifade eder. P değeri 0 ile 1 arasında bir değer alır ve 0’a ne kadar yakınsa, veri setinin normal dağılıma o kadar uygun olduğunu gösterir. P değeri 0.05’ten küçükse, veri setinin normal dağılıma uymadığı kabul edilir.

Kolmogorov Smirnov normallik testi, özellikle örneklem sayısı (n) 50’nin üzerindeyse bakılır. PD hastalarının tanı durumu için bu çalışmada kullanılan veri setinizin büyüklüğü 2105 olduğu için Kolmogorov Smirnov testi kullanılmıştır.

• Özellik Çıkarımı

Özellik seçimi, sınıflandırma işleminden önce izlenmesi gereken önemli bir adımdır çünkü bu işlem, genel sınıflandırma performansını artırır ve işlem süresi ile karmaşıklığı azaltır.

Hastaları sağlıklı ve Parkinson hastalığı sınıflarına ayırmak için özellik kümesini seçmek amacıyla ilk olarak, bağımsız değişkenler (cinsiyet, kilo, boy, yürüme hızı gibi) arasında kategorik ve sürekli değişkenler iki gruba ayrıldı.

- **Sürekli Değişkenler:** Sürekli değişkenler, ölçülebilen ve iki değer arasında sonsuz sayıda ara değeri alabilen değişkenlerdir. Hangi sürekli özelliklerin hedef değişkeniniz ile daha fazla ilişkili olduğunu öğrenmek için ANOVA testi uygulandı.

○ Mann Whitney U Testi

Normal dağılım özelliği göstermeyen bir dağılımda iki bağımsız grup ortalamalarını karşılaştırmak amacıyla kullanılan non-parametrik bir yöntemdir. Mann-Whitney U testi genellikle bağımlı değişken sıralı olduğu durumda kullanılır. Bu test için bir kaç gerekli ön-şart (varsayım) aşağıdaki gibi sıralanabilir:

1. Bağımsız değişkenin kategorik olması
 2. Bağımlı değişkenin sürekli ya da sıralı olması
 3. Popülasyondan seçilen örneklemin random (rastgele) olması
 4. Gruplardan elde edilen skorların normal dağılım göstermemesi.
- **Kategorik Değişkenler:** Kategorik değişkenler, grupları veya kategorileri temsil eden ve genellikle sayısal bir anlam taşımayan değişkenlerdir. Hangi kategorik özelliklerin hedef değişkeniniz ile daha fazla ilişkili olduğunu öğrenmek için Ki-kare testi uygulandı.

○ Ki-Kare Testi

Ki-kare testi, bir veya daha fazla kategoride beklenen ile görülen arasındaki farkları belirlemenin istatistiksel bir yoludur. Araştırmacılar bu parametrik olmayan testi, aynı örneklem popülasyonu içindeki kategorik değişkenleri karşılaştırmak için kullanmaktadır. Testin arkasındaki temel fikir, boş hipotezin doğru olması durumunda ne bekleneceğini belirlemek için gerçek veri değerlerini incelemektir.

Ki-kare testi, gözlenen değer ile beklenen değer her hangi bir şekilde farklılık gösterip göstermediğini belirlemek için kullanılır. Ki-kare formülü aşağıdaki şekilde gösterilir;

$$X^2 = \sum (O - E)^2 / E$$

İşte,

- O gözlenen değerdir
- E beklenen değerdir

Chi-Square Skorları, her bir özelliğin hedef değişken (bu durumda Diagnosis) ile ne kadar ilişkili olduğunu gösterir. Yüksek bir skor, o özelliğin hedef değişkenle güçlü bir ilişki içinde olduğunu, düşük bir skor ise çok az bir ilişki olduğunu gösterir.

Ki-kare testi sonucunda p-değeri 0.05'ten küçükse bu özellik hedef değişkenle anlamlı bir ilişki gösteriyor demektir.

Ki-kare testi sonucunda p-değeri 0.05'ten büyükse bu özellik hedef değişkenle bağımsız demektir ve genellikle modelin doğruluğunu artırmak için dışarıda bırakılabilir.

• K-En Yakın Komşu Algoritması (KNN)

K-en yakın komşu (*k-nearest neighbors*, KNN) algoritması [25], gözlemlerin birbirlerine olan benzerlikleri üzerinden tahminlerin yapıldığı gözetimli makine öğrenmesi modellerinde regresyon ve sınıflandırma problemlerinde kullanılan bir algoritmadır.

Bu algoritma kapsamında tahminde bulunmak istediğimiz gözlem birimine en yakın K adet farklı gözlem birimi tespit edilir ve bu K adet gözlem biriminin bağımlı değişkenleri üzerinden ilgili gözlem için tahminde bulunulur.

- **Destek Vektör Makineleri (SVM)**

Destek vektör makinesi (SVM) [26], N boyutlu bir uzayda her sınıf arasındaki mesafeyi en üst düzeye çıkaran optimum bir çizgi veya hiper düzlem bularak verileri sınıflandıran bir gözetimli makine öğrenme algoritmasıdır.

SVM'ler sınıflandırma problemlerinde yaygın olarak kullanılır. Karşıt sınıfların en yakın veri noktaları arasındaki marjı en üst düzeye çıkaran optimum hiper düzlemi bularak iki sınıf arasında ayırım yaparlar.

Sınıfları ayırt etmek için birden fazla hiper düzlem bulunabileceğinden, noktalar arasındaki marjı en üst düzeye çıkarmak algoritmanın sınıflar arasındaki en iyi karar sınırını bulmasını sağlar. Bu da, yeni verilere iyi genelleme yapmasını ve doğru sınıflandırma tahminleri yapmasını sağlar. Optimum hiper düzleme bitişik olan çizgiler, bu vektörler maksimum marjı belirleyen veri noktalarından geçtiği için destek vektörleri olarak bilinir.

- **Karar Ağaçları (DT)**

Karar ağacı [27], her bir iç düğümün bir özelliği (veya niteliği), her bir dalın bir karar kuralını ve her bir yaprak düğümünün sonucu temsil ettiği akış şeması benzeri bir ağaç yapısıdır. Bir karar ağacındaki en üst düğüm kök düğüm olarak bilinir ve herhangi bir alt düğümü olmayan düğümler yaprak düğüm olarak adlandırılır. Karar ağacı algoritması kök düğümden başlar ve bir yaprak düğüme ulaşana kadar girdi özellik değerlerine göre bir karar vererek ağaç boyunca ilerler. Yaprak düğümdeki değer, tahmin edilen çıktı değerini temsil eder.

- **Kök Düğüm:** Kök düğüm tüm veri kümesini temsil eder ve ağacı başlatmak için kullanılır. Ağacın başlangıç noktasıdır ve verileri maksimum bilgi kazancı veya minimum Gini Impurity sağlayan özelliğe göre böler.
- **İç Düğüm:** Her bir iç düğüm, verileri iki veya daha fazla alt kümeye ayıran bir özelliği temsil eder. Bölme işlemi özelliğin değerine göre gerçekleştirilir ve her bir gözlemin izleyeceği yolu belirler. İç düğüm daha sonra birden fazla alt düğüme bölünür.
- **Yaprak Düğüm:** Yaprak düğüm, verilerin daha fazla bölünemeyen bir alt kümesini temsil eder. Kendisine ulaşan gözlemler için nihai tahmini içerir. Tahmin, alt kümedeki çoğunluk sınıfına veya hedef değişkenin ortalama değerine dayanır.

- **Yapay Sinir Ağları (ANN)**

Makine öğrenmesi sınıflandırma uygulamaları kapsamında incelenebilen yapay sinir ağları (YSA) [28], insan beyninin özelliklerinden esinlenerek öğrenilen bilgiler ışığında yeni bilgiler oluşturabilme ve keşfetme yetenekleri sağlayarak bunları kendi kendine yapabilmesi amacıyla geliştirilen bilgisayar sistemleridir.

Yapay sinir hücreleri bir araya gelerek yapay sinir ağlarını oluştururlar. Ağ yapısı; girdi katmanı, ara katmanlar ve çıktı katmanından oluşur.

YSA da bir girdi ve bir çıktı seti bulunur. Eğitilmek istenen YSA ya öğretilecek bilgi bir vektör haline çevrilerek verilir. Doğru çıktıyı üretecek şekilde sistem parametreleri ayarlanır.

YSA da öğrenme 2 aşamada yapılır. İlk aşamada sistem eğitilirken üretilen çıktı değerleri verilir. Çıktı değerinin doğruluğuna göre ağırlık bağlantılarının ağırlıkları değiştirilir. Ağırlık eğitimi bittikten sonra sistemin performansını ölçmek için test verisi sisteme sunulur. Test aşamasında ağırlık değerlerinde herhangi bir değişiklik yapılmaz eğitimde kullanılan ağırlıklar kullanılır ve bu karşılaşmadığı test verileri için çıktılar üretir. Elde edilen çıktılar doğruluk değerlerine göre sistemin performansı değerlendirilir.

- **Topluluk Öğrenmesi (Ensemble Yöntemler)**

Topluluk öğrenme yöntemi, tek bir algoritmadan elde edilenden daha iyi bir sınıflandırma veya tahmin sonucuna sahip olmak için birçok makine öğrenimi algoritmasını birleştirmekten oluşan bir makine öğrenimi tekniğidir.

Topluluk öğrenmesinin avantajlarından biri, yüksek boyutlu ve karmaşık veri yapılarıyla başa çıkma yeteneğidir, yani topluluk öğrenmesi, veri kümelerinin varyansını azaltmaya yardımcı olur ve daha doğru sonuçlar sağlar.

Torbalama (Bagging), Yükseltme (Boosting), Random Subsample (Rastgele Alt-uzaylar), İstifleme (Stacking) en popüler ve yaygın olarak kullanılan topluluk öğrenme yöntemleridir.

- **Stacking (İstifleme)**

Farklı türdeki modeller (örneğin Karar Ağacı, SVM, KNN) bir arada kullanılarak bir topluluk modeli oluşturulur. Bu modellerin ürettikleri tahminler, "meta-model" adı verilen bir model (genellikle lojistik regresyon veya başka bir güçlü model) tarafından birleştirilerek nihai tahmin yapılır.

Stacking, heterojen modelleri (farklı türde algoritmalar) birleştirir, Stacking'de birden fazla farklı model bir arada kullanılır.

C. Performans Değerlendirme Metrikleri

Farklı ML modellerinin performansları, Doğruluk (Accuracy), Duyarlılık (Recall), Özgüllük (Precision), F1-Skoru (F1_Score), AUC değeri ve ROC eğrisi kullanılarak değerlendirilebilir.

- **Doğruluk (Accuracy)**

Doğru olarak sınıflandırılan örneklerin yüzdesidir.

$$\text{Accuracy} = \frac{Tn+Tp}{Tn+Tp+Fn+Fp}$$

- **Duyarlılık (Recall)**

Pozitif olarak tahmin etmemiz gereken işlemlerin ne kadarını pozitif olarak tahmin ettiğimizi gösteren bir metriktir.

$$\text{Recall} = \frac{Tn}{Tp + Fn}$$

- **Özgüllük (Precision)**

Pozitif olarak tahminlediğimiz değerlerin gerçekten kaç adedinin pozitif olduğunu göstermektedir.

$$\text{Precision} = \text{Tp} / \text{Tn} + \text{Fp}$$

- F1-Skoru (F1 Score)

F1 skoru, bir testin doğruluğunun bir ölçüsüdür —kesinlik ve duyarlılığın harmonik ortalamasıdır. Maksimum 1 (mükemmel kesinlik ve duyarlılık) ve minimum 0'a sahip olabilir. Genel olarak, modelinizin kesinliğinin ve sağlamlığının bir ölçüsüdür.

$$\text{F1_Score} = 2 \times \text{Precision} \times \text{Sensitivity} / (\text{Precision} + \text{Sensitivity}) \times 100\%$$

- AUC (Area Under Curve)

AUC, “ROC Eğrisinin Altındaki Alan” anlamına gelir. Yani, AUC, (0,0) ‘dan (1,1)’ e kadar tüm ROC eğrisinin altındaki iki boyutlu alanı (integral hesabı düşünün) ölçer.

AUC, olası tüm sınıflandırma eşiklerinde toplam performans ölçümü sağlar.

- ROC

ROC eğrisi çeşitli eşik ayarlarında sınıflandırma problemi için bir performans ölçümüdür. ROC bir olasılık eğrisidir ve AUC ayrılabilirliğin derecesini veya ölçüsünü temsil eder.

- Cohen’s Kappa

Bir sınıflandırma modelinin performansını ölçmek için yararlı bir araç olabilen iki değerlendirici arasındaki mutabakat düzeyini ölçmek için kullanılan bir ölçüttür.

Kappa katsayısı <0 ile 1 arasında değişir, burada <0 zayıf uyum anlamına gelir, >0,8 neredeyse mükemmel uyum anlamına gelir.

$$k = (\text{Po} - \text{Pe}) / (1 - \text{Pe})$$

Po : Derecelendiriciler arasında gözlemlenen göreceli uyum

Pe : Şans anlaşmasının varsayımsal olasılığı

III- Sonuç ve Tartışma

Bu çalışmada, modelin değerlendirilmesi için holdout doğrulama yöntemi uygulanmıştır. Veri seti, %75 eğitim ve %25 test oranında rastgele ikiye bölünmüştür.

Bu yöntem, modelin eğitim ve test performansını bağımsız olarak değerlendirmek için kullanılmıştır. Veri setinin yalnızca bir defa bölünmesi hem işlem süresini azaltmış hem de modelin test veri setindeki genel performansını ölçmek için yeterli olmuştur.

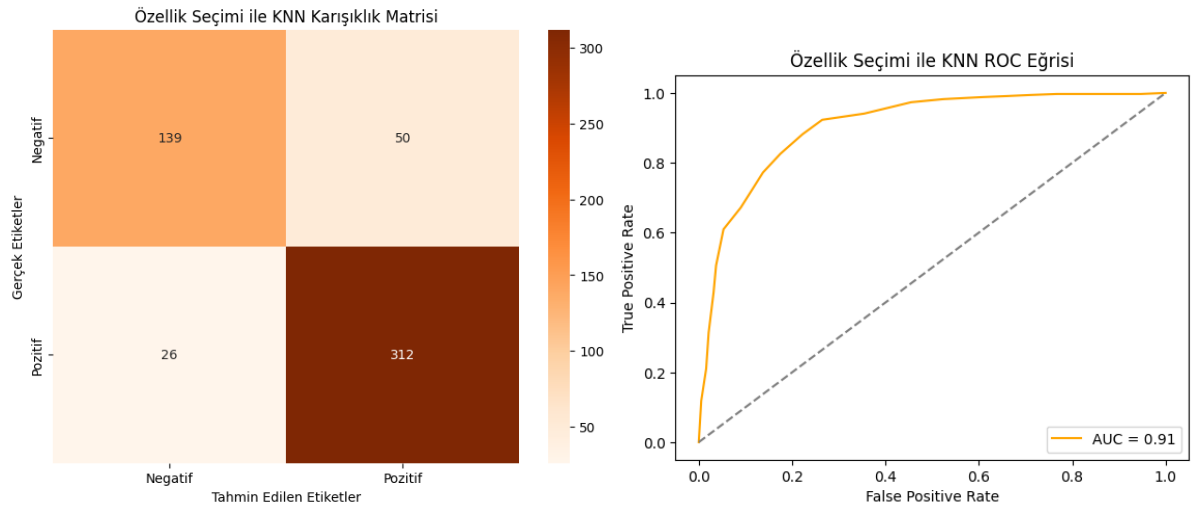
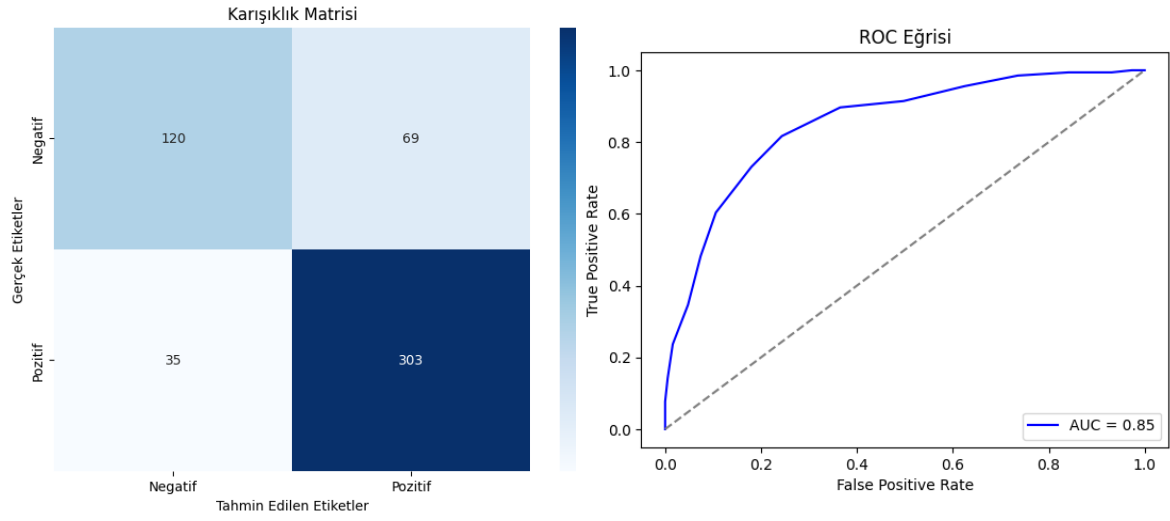
Veri normalizasyonu, farklı ölçeklerdeki özelliklerin aynı ölçekte değerlendirilmesini sağlayan önemli bir adımdır. Bu işlem, modelin performansını artırır ve farklı büyüklüklerdeki özelliklerin modele etkisini dengeleyerek daha iyi genelleme yapılmasını sağlar.

Bu çalışmada, veri setindeki özelliklerin her birinin ortalamasını 0 ve standart sapmasını 1 olacak şekilde dönüştürmek için StandardScaler yöntemi kullanılmıştır.

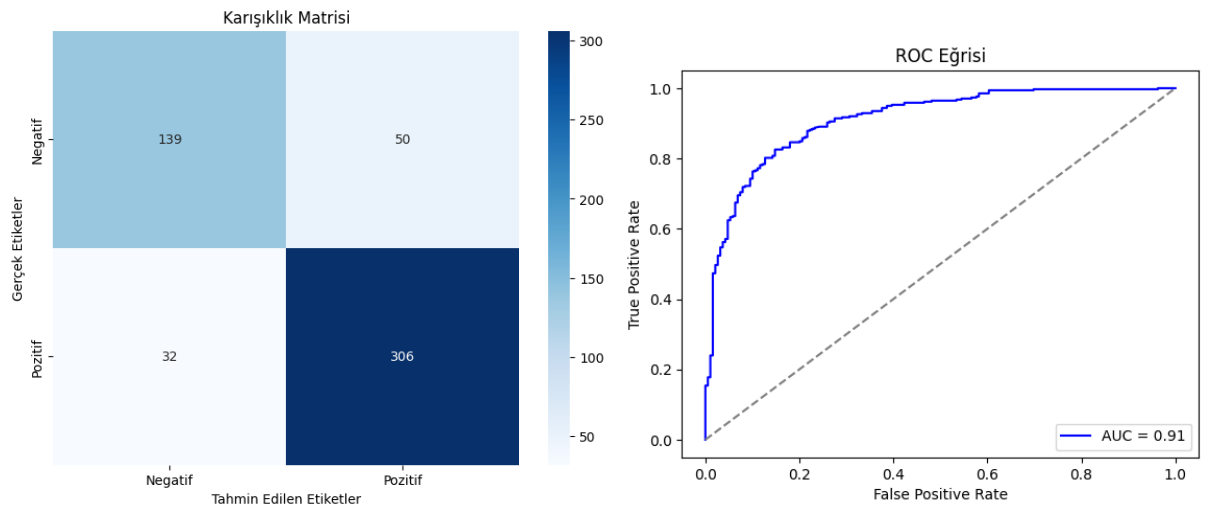
Bu yaklaşım, modelin daha dengeli bir şekilde öğrenmesini sağlar ve farklı büyüklüklerdeki özellikler arasındaki dengesizliği ortadan kaldırır.

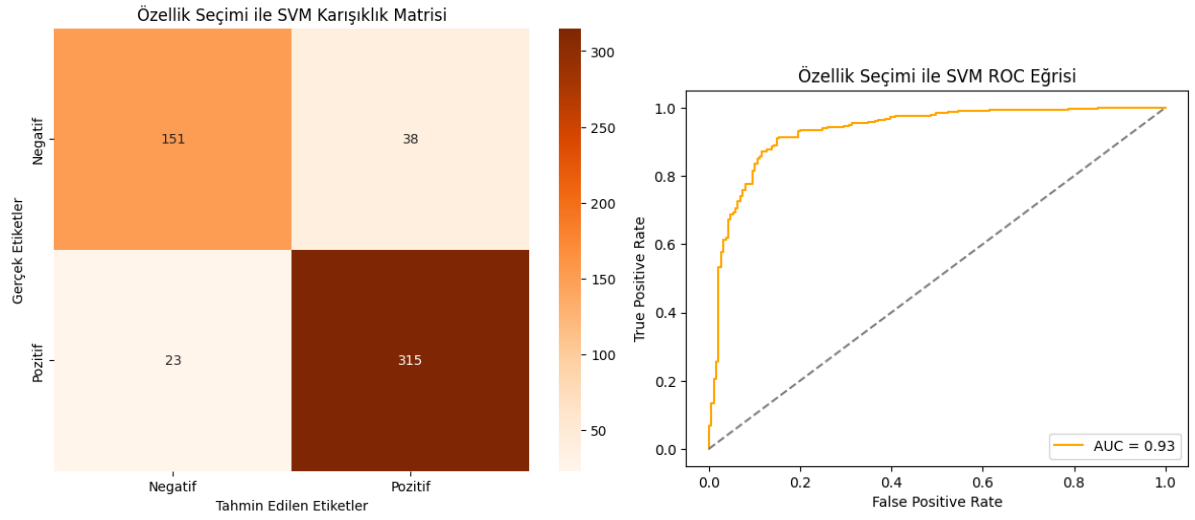
Sınıflandırma Algoritmalarına Göre Hata Matrisleri ve ROC Eğrileri

- **K-En Yakın Komşu Algoritması (KNN)**

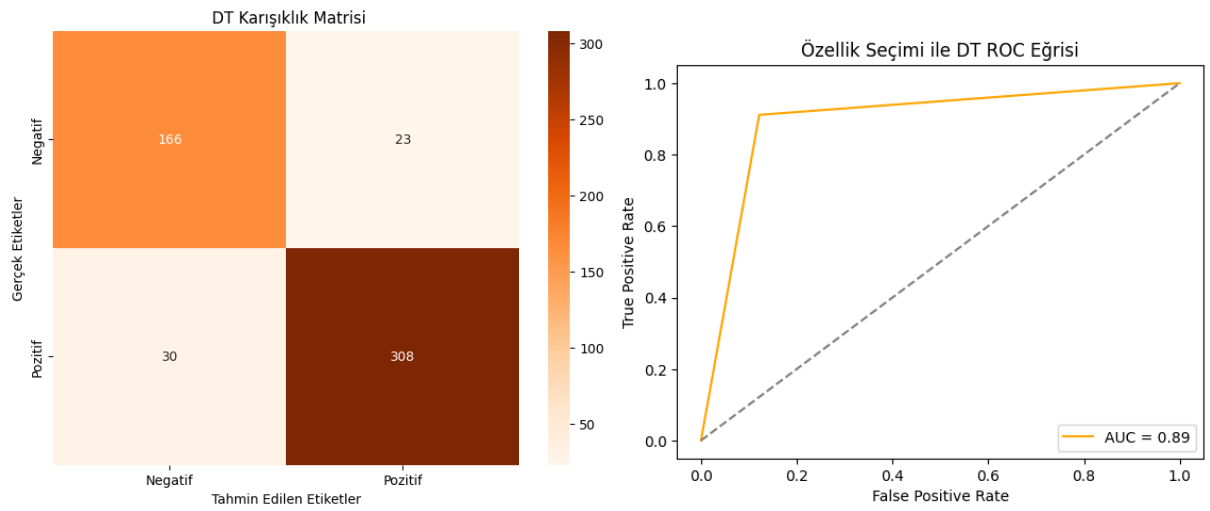
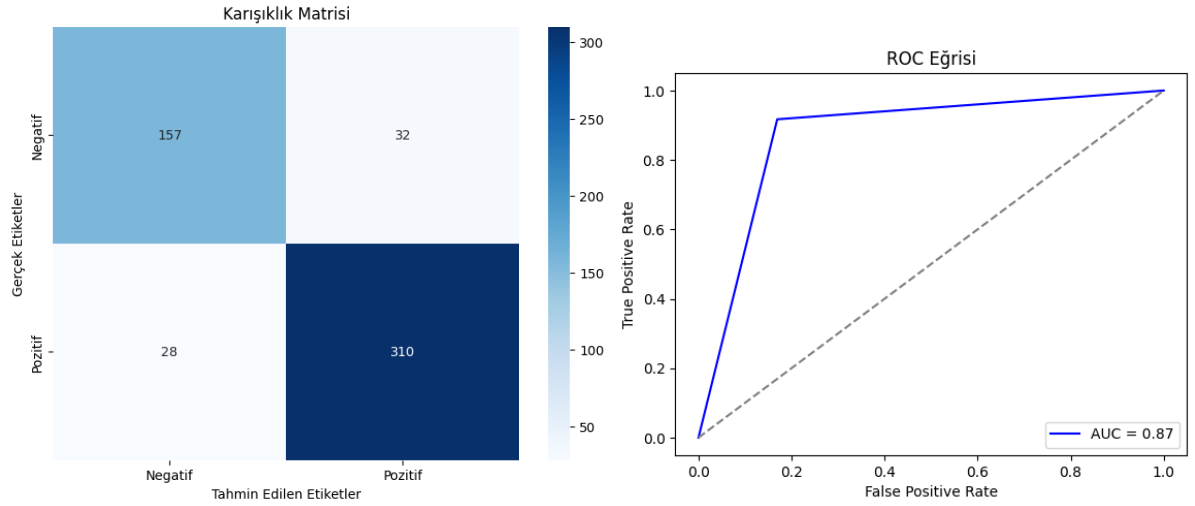


- **Destek Vektör Makineleri (SVM)**

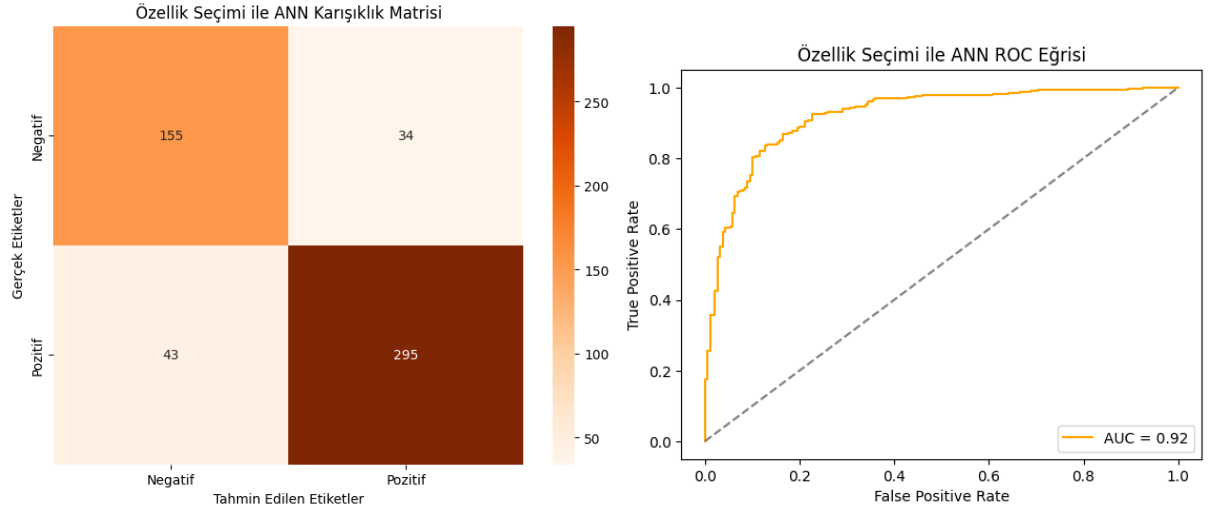
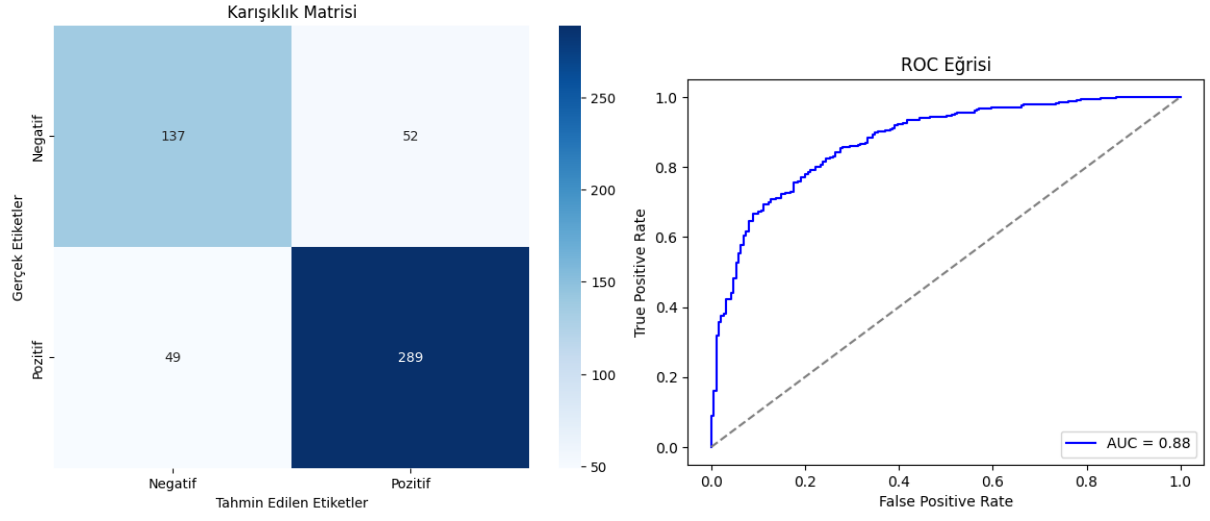




- Karar Ağaçları (DT)**

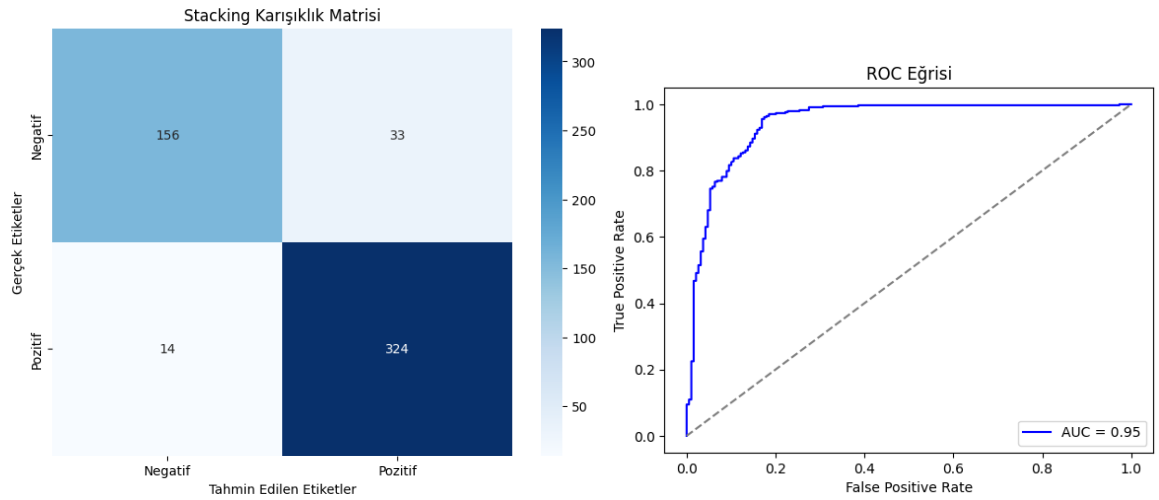


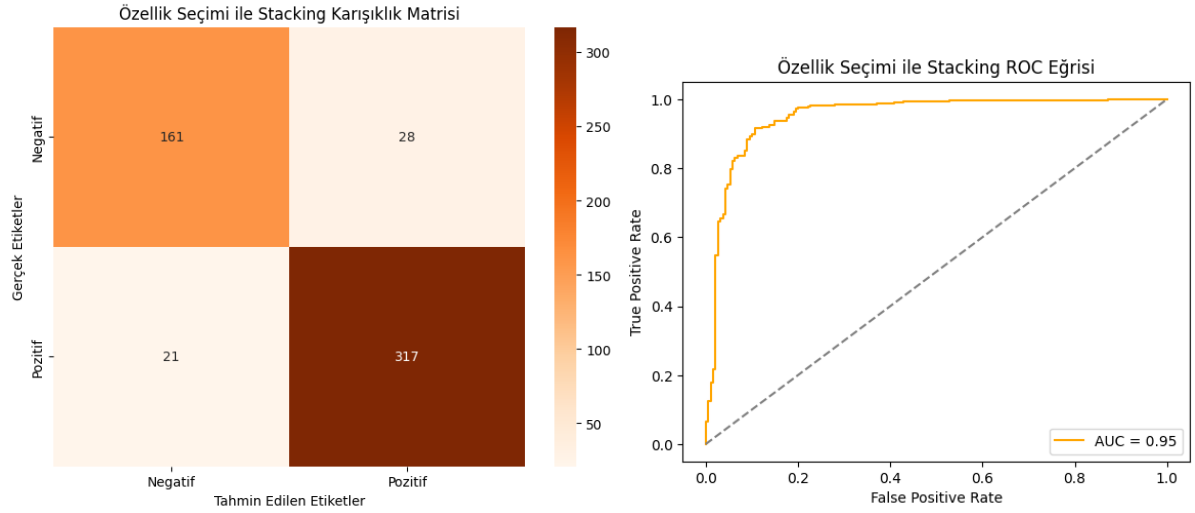
- Yapay Sinir Ağları (ANN)



- Ensemble Yöntemi (Topluluk Sınıflandırması)

Stacking





Sınıflandırma Algoritmalarına Göre Performans Parametreleri

| | Eğitim Doğruluk | Test Doğruluk | Duyarlılık | Özgüllük | F1-Skoru | Kappa | AUC |
|---------------------------|-----------------|---------------|------------|----------|----------|-------|-------|
| KNN | 0.806 | 0.787 | 0.896 | 0.634 | 0.853 | 0.553 | 0.852 |
| F-KNN (Özellik Seçimi) | 0.854 | 0.855 | 0.923 | 0.735 | 0.891 | 0.677 | 0.905 |
| SVM | 0.936 | 0.844 | 0.905 | 0.735 | 0.881 | 0.654 | 0.910 |
| F-SVM (Özellik Seçimi) | 0.913 | 0.884 | 0.932 | 0.798 | 0.911 | 0.743 | 0.932 |
| DT | 1.0 | 0.886 | 0.917 | 0.830 | 0.911 | 0.751 | 0.873 |
| F-DT (Özellik Seçimi) | 1.0 | 0.899 | 0.911 | 0.878 | 0.920 | 0.783 | 0.894 |
| ANN | 1.0 | 0.774 | 0.855 | 0.724 | 0.851 | 0.581 | 0.875 |
| F-ANN (Özellik Seçimi) | 1.0 | 0.853 | 0.872 | 0.820 | 0.884 | 0.685 | 0.921 |
| Stacking | 0.991 | 0.910 | 0.958 | 0.825 | 0.932 | 0.801 | 0.945 |
| Stacking (Özellik Seçimi) | 0.975 | 0.907 | 0.937 | 0.852 | 0.928 | 0.796 | 0.952 |

Performans Analizine Dayalı Sınıflandırma Algoritmaları Karşılaştırması

Bu çalışma, farklı sınıflandırma algoritmalarının performans parametreleri üzerindeki etkilerini inceleyerek özellik seçimi yönteminin genel performans üzerindeki rolünü ortaya koymayı amaçlamaktadır. Elde edilen sonuçlar, K-Nearest Neighbors (KNN), Destek Vektör Makineleri (SVM), Karar Ağaçları (DT), Yapay Sinir Ağları (ANN) ve Stacking gibi yaygın sınıflandırma algoritmalarının hem özellik seçimi yapılmadan hem de yapıldıktan sonraki performansını detaylı bir şekilde kıyaslamaktadır. Bu analizde, çeşitli metrikler üzerinden değerlendirmeler yapılmış olup, bu metrikler eğitim doğruluğu, test doğruluğu, duyarlılık, özgüllük, F1-Skoru, Kappa katsayısı ve Eğri Altında Kalan Alan (AUC) olarak belirlenmiştir.

KNN ve F-KNN Performansı

KNN algoritması, test doğruluğu açısından %78.7 başarı oranı göstermiştir. Ancak duyarlılık metriği %89.6, özgüllük metriği ise %63.4 seviyesinde olup, dengeli bir performans sergileyememiştir. Bununla birlikte, özellik seçimi yöntemi uygulandığında (F-KNN), test doğruluğu %85.5'e yükselmiş ve hem duyarlılıkta (%92.3) hem de özgüllükte (%73.5) belirgin bir artış kaydedilmiştir. Bu durum, özellik seçiminin KNN algoritması üzerindeki performans iyileştirici etkisini net bir şekilde göstermektedir.

SVM ve F-SVM Performansı

SVM algoritması test doğruluğu bakımından %84.4 oranında bir başarı göstermiştir. Özellikle duyarlılık (%90.5) ve özgüllük (%73.5) değerleri, algoritmanın dengeleyici bir performans sergilediğini ortaya koymuştur. Özellik seçimi ile zenginleştirilen F-SVM algoritması ise test doğruluğunu %88.4'e çıkarmış ve duyarlılıkta (%93.2) ile özgüllükte (%79.8) önemli iyileşmeler sağlamıştır. Bunun yanında F1-Skoru, Kappa ve AUC gibi daha karmaşık metriklerde de anlamlı artışlar kaydedilmiştir.

DT ve F-DT Performansı

Karar Ağaçları algoritması, eğitim veri seti üzerinde %100 doğruluk ile en iyi performansı sergilemiştir. Bununla birlikte, test doğruluğu %88.6 seviyesinde kalmış ve duyarlılık (%91.7) ile özgüllük (%83.0) oranlarında düşüş görülmüştür. F-DT algoritması ile özellik seçimi yapıldığında, test doğruluğu %89.9'a, duyarlılık %91.1'e ve özgüllük %87.8'e yükselmiştir. Bu iyileşmeler, özellik seçiminin DT algoritması üzerindeki etkinliğini göstermektedir.

ANN ve F-ANN Performansı

Yapay Sinir Ağları, eğitim seti üzerinde %100 doğruluk gösterirken test setinde %77.4 oranında daha düşük bir başarı kaydedilmiştir. Duyarlılık ve özgüllük metrikleri sırasıyla %85.5 ve %72.4 seviyesindedir. F-ANN algoritması, test doğruluğunu %85.3'e yükselterek duyarlılıkta (%87.2) ve özgüllükte (%82.0) belirgin bir iyileşme sağlamıştır. Bu durum, ANN algoritmasının özellik seçimi ile optimize edilebilir olduğunu göstermektedir.

Stacking Performansı

Ensemble yöntemlerinden biri olan Stacking algoritması, test doğruluğu %91.0 ile en iyi performansı sergileyen algoritmalar arasında yer almaktadır. Duyarlılık %95.8 ve özgüllük %82.5 seviyelerinde olup, çok yönlü ve dengeli bir performans sergilemiştir. Özellik seçimi uygulandığında test doğruluğu %90.7'ye, duyarlılık %93.7'ye ve özgüllük %85.2'ye yükselmiştir. Bunun yanında, AUC metriğinde %95.2 gibi çok yüksek bir başarı elde edilmiştir.

Genel Değerlendirme

Sonuçlar, özellik seçiminin sınıflandırma algoritmaları üzerindeki etkisini net bir şekilde ortaya koymuştur. Özellikle KNN, SVM, DT ve ANN gibi algoritmalarda özellik seçiminin test doğruluğu, duyarlılık, özgüllük ve AUC gibi kritik metriklerde anlamlı iyileşmeler sağladığı görülmektedir. Bununla birlikte, Stacking algoritması genel performansını koruyarak hem duyarlılık hem de AUC metriklerinde üstün başarı göstermiştir. Bu çalışma, özellik seçiminin sadece bireysel algoritmalarda değil, ensemble yöntemlerinde de önemli bir performans artışı sağlayabileceğini ortaya koymaktadır.

IV- Referanslar

- [1] Alzubaidi, M.S.; Shah, U.; DhiaZubaydi, H.; Dolaat, K.; Abd-Alrazaq, A.A.; Ahmed, A.; Househ, M. “The Role of Neural Network for the Detection of Parkinson’s disease: A Scoping Review,” in *Healthcare*, 9, 740, 16 June 2021, doi: <https://doi.org/10.3390/healthcare9060740>
- [2] Nader Salari, Mohsen Kazeminia, Hesam Sagha, Alireza Daneshkhah, Arash Ahmadi & Masoud Mohammadi, “The performance of various machine learning methods for Parkinson’s disease recognition: a systematic review,” in *Current Psychology* on vol. 42, pp. 16637-16660, 27 February 2022, doi: <https://doi.org/10.1007/s12144-022-02949-8>
- [3] Braak, H., Braak, E. (2000) “Pathoanatomy of Parkinson’s disease” in *J Neurol* 247, (Suppl 2) II3–II10, Nisan 2000. doi: <https://doi.org/10.1007/PL00007758>
- [4] Hirtz D, Thurman DJ, Gwinn-Hardy K, Mohamed M, Chaudhuri AR, Zalutsky R. “How common are the “common” neurologic disorders?” in *Neurology*, 68(5):326-37, 30 January 2007, doi: [10.1212/01.wnl.0000252807.38124.a3](https://doi.org/10.1212/01.wnl.0000252807.38124.a3)
- [5] “Statistics|Parkinson’s Foundation”, 2022
- [6] Van Den Eeden, S.K.; Tanner, C.M.; Bernstein, A.L.; Fross, R.D.; Leimpeter, A.; Bloch, D.A.; Nelson, L.M. “Incidence of Parkinson’s disease: Variation by age, gender, and race/ethnicity,” *Am J Epidemiol.* on vol. 157, pp. 1015–1022, 01 June 2003, doi: <https://doi.org/10.1093/aje/kwg068>
- [7] DeMaagd, G.; Philip, A. “Parkinson’s Disease and Its Management: Part 1: Disease Entity, Risk Factors, Pathophysiology, Clinical Presentation, and Diagnosis.” *Pharm. Ther.* 40, 504–532. Ağustos 2015,
- [8] Zehra Karapinar Senturk, “Early diagnosis of Parkinson’s disease using machine learning algorithms” in *Medical Hypotheses* on vol. 138, 109603, Mayıs 2020, doi: <https://doi.org/10.1016/j.mehy.2020.109603>
- [9] Aditi Govindu, Sushila Palwe, “Early detection of Parkinson’s disease using machine learning,” in *Procedia Computer Science* vol. 218, pp. 249-261, doi: <https://doi.org/10.1016/j.procs.2023.01.007>
- [10] Rizzo, G. et al. “Accuracy of clinical diagnosis of Parkinson disease: a systematic review and meta-analysis” *Neurology* 86, 566–576, 9 February 2016, doi: <https://doi.org/10.1212/WNL.0000000000002350>

- [11] Taigo Ítalo Pedrosa, Felipe F. Vasconcelos, Leonardo Medeiros, Leandro Dias Silva, "Machine Learning Application to Quantify the Tremor Level for Parkinson's Disease Patients," in *Procedia Computer Science* vol. 138, pp. 215-220, 2018, doi: <https://doi.org/10.1016/j.procs.2018.10.031>
- [12] R. Prashanth, Sumantra Dutta Roy, "Novel and improved stage estimation in Parkinson's disease using clinical scales and machine learning," in *Neurocomputing*, vol. 305, pp. 78-103, 30 August 2018, doi: <https://doi.org/10.1016/j.neucom.2018.04.049>
- [13] Jain V. "Introduction to KNN Algorithms. Analytics Vidhya." Published January 31, 2022. Accessed March 10, 2024. doi: <https://www.analyticsvidhya.com/blog/2022/01/introduction-to-knn-algorithms/>
- [14] P. Mounika, S. Govinda Rao, "Machine Learning and Deep Learning Models for Diagnosis of Parkinson's Disease: A Performance Analysis" in *IEEE*, 20 Aralık 2021, doi: [10.1109/I-SMAC52330.2021.9640632](https://doi.org/10.1109/I-SMAC52330.2021.9640632)
- [15] Support Vector Machine – an overview | ScienceDirect Topics. Published 2024. Accessed March 10, 2024. <https://www.sciencedirect.com/topics/computer-science/support-vector-machine>
- [16] Arti Rana, Ankur Dumka, Rajesh Singh, Manoj Kumar Panda, Neeraj Priyadashi, Bhakisipho Twala, "Imperative Role of Machine Learning Algorithm for Detection of Parkinson's Disease: Review, Challenges and Recommendations" in *Diagnostics* 2022, 12(8), 2003, doi: <https://doi.org/10.3390/diagnostics12082003>
- [17] Neharika, D.B.; Anusuya, S. "Machine Learning Algorithms for Detection of Parkinson's Disease using Motor Symptoms: Speech and Tremor." *IJRTE* **2020**, 8, 47–50.
- [18] Benba, A.; Jilbab, A.; Hammouch, A.; Sandabad, S. "Voiceprints analysis using MFCC and SVM for detecting patients with Parkinson's disease", in *Proceedings of the IEEE 2015 International Conference on Electrical and Information Technologies (ICEIT)*, Marrakech, Morocco, 25–27 March 2015; pp. 300–304.
- [19] Betül Erdoğan Sakar, M. Erdem İsenkul, C. Okan Sakar, Ahmet Sertbaş, Fikret Gürgen, Şakir Delil, "Collection and Analysis of a Parkinson Speech Dataset With Multiple Types of Sound Recordings," in *IEEE Journal of Biomedical and Health Informatics*, ol. 17, no. 4, pp. 828-834, July 2013, doi: 10.1109/JBHI.2013.2245674.
- [20] Decision tree learning. In: *Wikipedia*. ; 2024. Accessed March 10, 2024. doi: https://en.wikipedia.org/w/index.php?title=Decision_tree_learning&oldid=1211641048
- [21] Sumit Das, Tanusree Saha, Ira Nath, Dipansu Mondal, "Exploring Machine Learning Methods for Developing a Predictive System for Parkinson's Disease," in *Biosciences Biotechnology Research*, vol. 21, number 2, doi: [http://dx.doi.org/10.13005/bbra/3248](https://dx.doi.org/10.13005/bbra/3248)
- [22] Ziaul Hasan Majumder, Abu Khaer, Julkar Nayeem Mahi, Shaiful Islam Babu & Subrata Kumar Aditya, "Decision Support Technique for Prediction of Acute Lymphoblastic Leukemia Subtypes Based on Artificial Neural Network and Adaptive Neuro-Fuzzy Inference System," in *Inventive Systems and Control*, pp. 539-554, 8 Haziran 2021

- [23] Pahuja G. and Nagabhushan T., A comparative study of existing machine learning approaches for Parkinson's disease detection, *IETE Journal of Research*. (2018) 1–11, doi: <https://doi.org/10.1504/ijbet.2018.10019041>.
- [24] Rabie El Kharoua, “Parkinson's Disease Dataset Analysis,” Publisher *Kaggle*, Author Rabie El Kharoua, 2024, doi: 10.34740/kaggle/dsv/8668551, url: <https://www.kaggle.com/dsv/8668551>
- [25] Serdar Tafralı, “K-en Yakın Komşu Algoritması Nasıl Çalışır?”, 20 Temmuz 2022, url: https://miuul.com/blog/k-en-yakin-komsu-algoritmasi-nasil-calisir?gad_source=1&gclid=CjwKCAiAmfq6BhAsEiwAX1jsZ0fOnjiCqps4EL_uP2aSHl-XTFHz9ZtgZWbvtuFXEVwsghrA8LZiKR0CZiUQAvD_BwE
- [26] “Destek vektör makineleri (SVM'ler) nelerdir?”, url: <https://www.ibm.com/topics/support-vector-machine>
- [27] “Makine Öğrenmesi Karar Ağaçları (Decision Tree) Nedir?”, url: <https://bulutistan.com/blog/makine-ogrenmesi-karar-agaci-decision-tree-nedir/>
- [28] Betül Kara, “Yapay Sinir Ağları”, url: <https://miuul.com/blog/yapay-sinir-aglari>