KTO KARATAY ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK – ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRİK VE BİLGİSAYAR MÜHENDİSLİĞİ TEZLİ YÜKSEK LİSANS PROGRAMI

AĞ TRAFİĞİ ANALİZİNDE IP İTİBARI KULLANILARAK MAKİNE ÖĞRENMESİ YÖNTEMLERİNİN PERFORMANSLARININ ARTTIRILMASI

Furkan DANIŞ

Yüksek Lisans Tezi

KONYA
Ocak 2023

AĞ TRAFİĞİ ANALİZİNDE IP İTİBARI KULLANILARAK MAKİNE ÖĞRENMESİ YÖNTEMLERİNİN PERFORMANSLARININ ARTTIRILMASI

Furkan DANIŞ

KTO Karatay Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı
Elektrik ve Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı

Yüksek Lisans Tezi

Tez Danışmanı: Dr. Öğr. Üyesi Semih YUMUŞAK

Konya
Ocak 2023

BİLDİRİM

Enstitü tarafından onaylanan Yüksek Lisans tezimin tamamını veya herhangi bir kısmını basılı veya dijital biçimde arşivleme ve aşağıda belirtilen koşullar dahilinde erişime açma iznini KTO Karatay Üniversitesine verdiğimi bildiririm. Bu izinle, Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak ve gelecekteki çalışmalar (makale, kitap, lisans, patent vb.) için tezimin tamamının veya bir bölümünün kullanım hakları yalnızca bana ait olacaktır.

Tezimin bütünüyle kendi çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Telif hakkı bulunan ve sahiplerinden yazılı izinle kullanılması zorunlu olan kaynakları, yazılı izin alarak kullandığımı ve istenildiğinde izinlerin suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayımlanan "Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge" kapsamında, tezim, aşağıda belirtilen koşullar haricince, YÖK Ulusal Tez Merkezi ve KTO Karatay Üniversitesi Açık Erişim Sisteminde erişime açılır.

☐ Enstitü / Fakülte Yönetim Kurulu kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren 2 yıl ertelenmiştir.[1]

☐ Enstitü / Fakülte Yönetim Kurulunun gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren … ay en fazla 6 ay ertelenmiştir.[2]

☐ Tezimle ilgili gizlilik kararı verilmiştir.[3][4]

18 Ocak 2023

İmza

_____

Furkan DANIŞ

[1] MADDE 6 (1) Lisansüstü tezle ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir.

[2] MADDE 6 (2) Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış ve internetten paylaşılması durumunda 3. şahıslara veya kurumlara haksız kazanç imkanı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulunun gerekçeli kararı ile altı ayı aşmamak üzere tezin erişime açılması engellenebilir.

[3] MADDE 7 (1) Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik, sağlık vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir. Kurum ve kuruluşlarla yapılan işbirliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.

[4] MADDE 7 (2) Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir.

ETİK BEYAN

KTO Karatay Üniversitesi Lisansüstü Eğitim Enstitüsü Tez Hazırlama ve Yazım Kurallarına uygun olarak Dr. Öğr. Üyesi Semih YUMUŞAK danışmanlığında tarafımdan üretilen bu tez çalışmasında; sunduğum tüm veri, enformasyon, bilgi ve belgeleri bilimsel etik kuralları çerçevesinde elde ettiğimi, tüm değerlendirme, analiz, bulgu ve sonuçları bilimsel usullere uygun olarak sunduğumu, tez çalışmasında yararlandığım kaynakların tümüne bilimsel normlara uygun biçimde atıfta bulunarak kaynak gösterdiğimi, tezimin kaynak gösterilen durumlar dışında özgün olduğunu bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

18 Ocak 2023

İmza

_____

Furkan DANIŞ

TEŞEKKÜR

Tez çalışmalarım boyunca yardım ve desteklerini esirgemeyen, tüm çalışmalarım boyunca yol gösteren tez danışmanım Sayın Dr. Öğr. Üyesi Semih YUMUŞAK'a. Manevi desteklerini hiçbir zaman esirgemeyen, her zaman yanımda olan sevgili eşim ve kıymetli aileme teşekkür ederim.

18 Ocak 2023 Furkan DANIŞ

ÖZET

Furkan DANIŞ

Ağ Trafiği Analizinde IP İtibarı Kullanılarak Makine Öğrenmesi Yöntemlerinin Performanslarının Arttırılması

Yüksek Lisans Tezi

Konya, 2023

Günümüz bilgi ve iletişim sistemlerinde büyük miktarda ve heterojen ağ trafiği verisi üretilmektedir. Hücresel ağlar, web sunucular ve nesnelerin interneti dünyasındaki cihazların (IoT) ürettikleri ağ trafiği verisinin analiz edilmesi ve güvenlik metriklerine göre etkisinin ölçülmesi ihtiyacı doğmuştur. Bu ihtiyaca yönelik ise günümüz literatüründe ağ trafik sınıflandırma yöntemleri önerilmiştir. Ağ trafik verisinin içerisindeki kullanıcı hareketliliği ve heterojenliği gibi karmaşıklıklar verilerin sınıflandırılmasındaki zorluklar olarak kabul edilmektedir. Birçok ticari sistemin bu verilere ek olarak IP itibarını kullanarak kullanıcı trafiğini yönettiği bilinmektedir. Bu çalışmada ağ trafiğindeki verilerin zararlı olma durumuna göre ayrıştırılmasında, IP itibarının makine öğrenmesi algoritmaları ile oluşturulan modeller üzerindeki etkileri incelenmiştir. Bu amaçla kullanılan ve işlenen veri seti, veri ön işleme teknikleri ve IP itibarı ile veri kümesinin zenginleştirilmesi aşamaları açıklanmaktadır. Deneysel sonuçlar, IP itibarı bilgisinin zararlı trafiğin tespit edilmesinde makine öğrenmesi algoritmalarının performansını artırdığı göstermektedir.

Anahtar Kelimeler

Ağ güvenliği, ağ analizi, IP itibarı, makine öğrenmesi, sınıflandırma algoritmaları, zararlı internet trafiği

ABSTRACT

Furkan DANIŞ

Improving the Performance of Machine Learning Methods Using IP Reputation in Network Traffic Analysis

Master's Thesis

Konya, 2023

In today's information and communication systems, large amounts of heterogeneous network traffic data are produced. The need to analyze the network traffic data produced by cellular networks, web servers and devices in the world of the Internet of Things (IoT) and measure its effect according to security metrics has arisen. Network traffic classification methods have been proposed in today's literature for this need. Complexities such as user mobility and heterogeneity in network traffic data are considered as difficulties in classifying data. Many commercial systems are known to manage user traffic using IP reputation in addition to this data. In this study, the effects of IP reputation on the models created by machine learning algorithms in the separation of data in network traffic according to their maliciousness were examined. The data set used and processed for this purpose, data pre-processing techniques and the stages of enriching the data set with respect to IP are explained. Experimental results show that IP reputation information improves the performance of machine learning algorithms in detecting malicious traffic.

Keywords

Network security, network analysis, IP reputation, machine learning, classification algorithms, malicious internet traffic

İÇİNDEKİLER

TABLOLAR DİZİNİ

ŞEKİLLER DİZİNİ

KISALTMALAR DİZİNİ

Kısaltma	Açıklama
WAF	Web Application Firewall
SIEM	Security Information and Event Management
AUC	Area Under The Curve
WMI	Windows Management Instrumentation
ROC	Receiver Operating Characteristics
ML	Machine Learning
MLP	Multi Layer Perceptron
VPN	Virtual Private Network
DDOS	Distributed Denial of Service Attack
NAC	Network Access Control
HTTP	Hyper Text Transfer Protocol
IOT	Internet Of Things
IP	Internet Protocol
MAC	Media Access Control
IDS	Intrusion Detection System
IPS	Intrusion Prevent System

1. GİRİŞ

Günümüz bilgi ve iletişim çağında internet artık hayatımızın her yerinde aktif rol oynamaktadır. Profesyonel işlerde, sağlık sektöründe, eğitim sektöründe, otomobil ve iş makinelerinde, akıllı ev aletlerinde, akıllı telefonlarımızda ve tarım makinelerinde kısacası uzaktan yönetebildiğimiz, veri sorgulayabildiğimiz ve gerektiğinde bir takım işlerimizi otomatik hale getirebildiğimiz ağlar üzerinden etkişelim kurduğumuz her ortamda internetin varlığından ve bahsi geçen cihazların haberleşmesinden söz edebiliriz. İnternetin hayatımızdaki bu denli yoğun kullanımı tahmin edileceği üzere güvenlik kaygısını da beraberinde getirmiştir. Herhangi bir internet erişimini düşündüğümüzde üç ana aktörden söz edebiliriz. Bunlardan birincisi internet üzerinden hizmet veren uygulama (host), ikincisi bu hizmeti alan ve çeşitli işlemler gerçekleştiren son kullanıcı (client) ve üçüncüsü ve en önemlisi ise ilk iki ortamın arasına girmeye çalışan zararlı kullanıcı, (hacker) saldırganlardır. Güvenli bir internet erişiminin sağlanabilmesi için günümüzde yazılımdan, sunucu ayarlarına, donanım kurulumuna ve ağ operasyonlarına kadar geniş bir koruma metadolojisi uygulanır. Büyük ölçekli bilgi işlem alt yapısına sahip bir organizasyon düşünüldüğünde fiziksel sunucular üzerinde yapılan sanallaştırma operasyonları ile uygulamaların sanal sunucular üzerinden çalıştırılması ve bu sanal sunucuların da bir takım güvenlik kurallarıyla kurgulandığı bilinmektedir. Ağ çerçevesinde yapılan güvenlik önlemlerinde arındılmış bölge (DMZ), sanal ağlara bölme (VLAN) ve yüzlerce güvenlik duvarı (firewall) erişim kurallarından söz edebiliriz. Böylesine bir kurguya rağmen güvenlik konusunda günümüzde yetersiz kalındığı için web uygulamaları güvenlik duvarı (WAF), güvenlik bilgileri ve olay yönetimi (SIEM) sağlayan cihazlar, saldırı tespit ve önleme (IDS-IPS) sistemleri gibi güvenlik ürünleri de sisteme dahil edilir. Bu ürünlerin temel işlevi gelen trafik içerisindeki veriyi sınıflandırarak zararlı trafik verisini zararsız olandan ayırt etmektir.

Bahsi geçen ticari ve açık kaynak kod lisanslı ürünler barındırdıkları makine öğrenimi algoritmaları ve kendilerine özgü algoritmaları ile zararlı trafiği önceden veya saldırı anında tespit etmeye yararlar. Ancak donanım, yazılım ve ağ cihazlarının sürekli gelişmesi ile saldırı tekniklerinin gelişmesi paralel ilerlediğinden bu ürünlerin yeni çıkan saldırılara veya anomalilere karşı da direnç göstermesi gerekmektedir. Günümüz literatüründe üreticilerin de öğrenme ve tahmin yönünde algoritmalarını sürekli

geliştirdiklerini söyleyebiliriz. Böylesine güncel ve sektörde karşılığı bulunan bu problemi daha yakından incelemek için bu çalışmada Makine Öğrenmesi Algoritmaları ile IP İtibar Derecesine Dayalı Ağ Trafiği Sınıflandırma çalışması yapılmıştır.

Birinci bölümde ağ cihazlarının ve güvenlik ürünlerinin sahadaki kullanımı ve korelasyonu örnek bir topoloji üzerinden anlatılmıştır. Ardından veri madenciliği, makine öğrenmesi, sınıflandırma ve tahmin kavramlarına yer verilmiş ve kullanılan algoritmalar açıklanmıştır. Devamında veri ön işleme çalışmaları yapılarak sınıfladırma algoritmalarında işlemeye hazır hale getirilmiştir. Son olarak IP itibarına yönelik bir analiz çalışması yapılmış ve veri seti üzerinde test-eğitim işlemleri yapılmaya hazır hale getirilmiştir.

2. LİTERATÜR TARAMASI

Günümüz büyük ölçekli bilgi işlem operasyonlarında internet trafiğinin içerisinde yer alan zararlıların tespiti ağ yönetimi ve kaynak tahsisi aynı zamanda sistemin sürekliliği ve kalitesi (QoS) için önemli bir tekniktir. Ağ trafiğindeki zararlıların tespiti için artık geleneksel olarak kabul edilen yöntemlerden olan geliştirilmiş yapay sinir ağı tabanlı derin öğrenme modelleri ve optimizasyon algoritmaları ile tahmin iyileştirmesi yapılması gibi çalışmalara sıklıkla rastlanmaktadır. Yang ve arkadaşları yaptıkları çalışmada geleneksel doğrusal ve doğrusal olmayan ağ trafiği tahmin modellerinin, gelecekteki trafiğin tahmin edilmesi için yeterli doğruluğunu sağlayamayacağı fikrini savunmuşlardır. Bu fikirden hareketle SA (Simulated Annealing) yani Benzetilmiş Tavlama algoritması ile optimize edilmiş ARIMA (Otoregresif Entegre Hareketli Ortalama Modeli), BPNN (Geri Yayılımlı Sinir Ağı) tabanlı bir ağ trafiği tahmin yöntemi önermişlerdir. Önerdikleri bu yeni yöntem BPNN küresel optimizasyon yeteneğinin geliştirilmesiyle, geçmiş ağ trafiği verilerinin doğrusal ve doğrusal olmayan yasalarını çıkarma potansiyelini tam olarak gerçekleştirebildiğinden tahmin doğruluğunun iyileştiği rapor edilmiştir (Zhang vd., 2013).

Trafik sınıflandırması, ağ yönetiminde güvenlik izleme sistemlerinden hizmet kalitesi ölçümlerine kadar geniş uygulamalara sahiptir. Zhang ve arkadaşlarının yaptıkları çalışmada istatistiksel öznitelik tabanlı sınıflandırma yöntemleri uygulamak için makine öğrenmesi tekniklerinin uygulandığını gözlemlemişlerdir. Yaptıkları çalışmada eğitim sürecinde belli bir prosedür gerektirmemesi, çok sayıda sınıfın üstesinden gelebilmesi gibi avantajlarıyla üstün sınıflandırma performansı sergilediğinden en yakın komşu (NN) tabanlı yöntemi kullanmışlardır (Zhang vd., 2013).

Ağ trafiğinin sınıflandırmasında kullanılan makine öğrenimi modelleri, eğitim verilerinin ve test verilerinin bağımsız özdeş dağılımlara sahip olduğu varsayımını yapar. Ancak trafik özelliklerindeki değişiklikler nedeniyle pratik trafik sınıflandırmasında bu varsayım ihlal edilebilir. Mevcut verilerle eğitilen modeller yeni trafiği sınıflandırmada yetersiz kalabilmektedir. Sun ve arkadaşları yaptıkları çalışmada bahsi geçen varsayım yapılmadan bir transfer öğrenme modeli önermişlerdir. Transfer öğrenme modelinde temel sınıflandırıcı olarak maksimum entropi modeli (Maxent) benimsenmiştir. Önerilen yöntemin etkinliğini incelemek için ise Cambridge Üniversitesi'nde toplanan trafik veri

kümesini, eğitim ve test veri kümesinin aynı olmaması koşuluyla kullanmışlardır. Elde edilen sonuçlar incelendiğinde transfer öğrenme modeline dayalı olarak iyi bir sınıflandırma performansının elde edildiği görülmüştür (Sun vd., 2018).

Ağ trafiğinin sınıflandırılması Yamansavaçcılar ve arkadaşları tarafından yapılan çalışmada ise ağ trafiğindeki nitelikler yerine Facebook, Twitter ve daha birçok popüler son kullanıcı servisleri ele alınarak J48, Random Forest, K-NN ve Bayes algoritmaları ile sınıflandırma uygulaması yapılmıştır (Yamansavascilar vd., 2017).

Trafik mühendisliği, güvenlik izleme ve hizmet kalitesi (QoS) gibi kablolu ve kablosuz ağlardaki birçok uygulama ile zamanında ve doğru trafik sınıflandırması ve uygulama karakterizasyonu giderek daha önemli hale gelmektedir. Özellikle yazılım tanımlı ağ oluşturma (SDN), gelecekteki IP ağları ve 5G kablosuz ağlar üzerinde büyük etkisi olan yeni bir ağ oluşturma paradigmasıdır. Bağlantı noktası tabanlı ve yük tabanlı algoritmalar gibi geleneksel sınıflandırma yöntemleriyle karşılaştırıldığında, makine öğrenimi (ML) yaklaşımları, yükten bağımsız trafik istatistiklerini kullanarak internet trafiği karakterizasyonunda daha iyi bir seçim sunar. Fan ve arkadaşları çalışmasında trafik sınıflandırması için denetimli destek vektör makinesi (SVM) ve denetimsiz k-ortalama kümeleme olmak üzere iki ML algoritmasını incelemiştir (Fan ve Liu, 2017)

Web uygulamalarında en tehlikeli saldırılardan bir tanesi de SQL Injection saldırılarıdır. Bu saldırı, web uygulamaları için ciddi bir tehdit unsurudur. Bu saldırıyı erken bir aşamada engelleyerek veya meydana geldiğinde tespit ederek hafifletmek için ağ üzerindeki trafiğin iyi incelenmesi ve sınıflandırılması gerekmektedir. Jemal ve arkadaşlarının yaptıkları çalışmada SQL enjeksiyon saldırısına genel bir bakış ve önerilen algılama ve önleme çözümlerinin bir sınıflandırılması sunulmuştur (Jemal vd., 2020).

Kötü amaçlı bir IP adresini kara listeye alma kararı, davranış geçmişinin yanı sıra paket trafik verilerinin çeşitli yönlerinin dikkatli bir şekilde incelenmesini gerektirir. IP kara listeye alma için mevcut güvenlik izlemesi çoğu deneyimli uzmanların tecrübelerine dayalıdır. Bu soruna makine öğrenimi tekniklerini uygulanarak doğru kara listeye alma işlemi için modeller üretilebilecek olan BlackEye isimli sınıflandırma modeli tasarlanmış ve model içerisinde lojistik regresyon ve rastgele orman algoritmaları ile veri temizleme ve sınıflandırmayı birleştiren çok aşamalı yöntemin iyi sonuçlar verdiğini Jeon ve arkadaşları çalışmalarında belirtilmiştir. Bu çalışma ile kara listeye alma olayını yaklaşık

%90 oranında azaltarak kötü amaçlı IP adresinin faaliyette olduğu süreyi de ortalama 27 gün kadar kısalttığı sonucu paylaşılmıştır (Jeon ve Tak, 2022).

Trafik akışlarının doğru sınıflandırılması, güvenlik izleme, IP yönetimi, izinsiz giriş tespiti vb. konularda bilişim sistemi yöneticilerine yardımcı olmaktadır. Trafik sınıflandırma sorununu çözmek için literatürde makine öğrenimi (ML) yaklaşımları yaygın olarak kullanılmaktadır. Shafiq ve arkadaşlarının yaptıkları çalışmada geleneksel ML yaklaşımlarına ek olarak ağırlıklı karşılıklı bilgi (QMI) metriği ve ROC eğrisi altındaki alan (AUC) olmak üzere iki metriği kullanan WMI_AUC adlı ML tabanlı bir hibrit özellik seçim algoritması önerilmiştir. Önerilen yaklaşım, ML sınıflandırıcılarının doğruluğunu artırmakta ve kötü niyetli trafiğin tespit edilmesine yardımcı olmaktadır (Shafiq vd., 2018).

Siber tehditlerden korunmak için makine öğreniminin uygulanmasına yönelik çalışmalardaki zorluk, hızlı ve doğru siber tehdit tespitini sağlayan uygun özelliklerin belirlenmesi ve seçilmesidir. Beechey ve arkadaşlarının yaptıkları çalışmada her bir özelliğin belirsizliğinin genel sınıflandırma kararı üzerindeki etkisi incelenmiştir. Önerilen yaklaşım ile ağ güvenliği saldırılarının son zamanlardaki zorlu senaryosu üzerinde değerlendirme yapılır ve çoklu özellik seçim teknikleri ile karşılaştırılır. En iyi performansı % 0.99'luk bir F1 puan ve % 93.25'lik doğruluk ile karar ağacı algoritması göstermiştir (Beechey vd., 2021).

İnternet üzerinden geçen mevcut büyük miktarda trafikle birlikte, internet servis sağlayıcıları (ISS) ve ağ servis sağlayıcıları (NSS), internetten geçen uygulama akışının türünü doğru bir şekilde tahmin etmenin çeşitli yollarını aramaktadır. Bu tür bir tahmin, uygulama türünün önceden bilinmesini gerektirdiğinden, güvenlik ve ağ izleme uygulamaları için kritik öneme sahiptir. Bağlantı noktası tabanlı veya yük tabanlı analiz kullanan geleneksel yöntemler artık yeterli değildir, çünkü birçok uygulama algılanmaktan kaçınmak için dinamik bilinmeyen bağlantı noktası numaraları, maskeleme ve şifreleme teknikleri kullanmaktadır. Dong çalışmasında trafiğin uygulama türünü belirlemek için ağ akış düzeyi özellikleri kullanılmıştır. Ayrıca ağ trafiği tanımlamasındaki dengesizlik sorununu çözmek için, maliyete duyarlı SVM (CMSVM) adlı geliştirilmiş bir destek vektör makinesi algoritması kullanmıştır. CMSVM, uygulamalar için dinamik olarak bir ağırlık atayan aktif öğrenmeye sahip çok sınıflı bir

SVM algoritmasını benimser. Sonuçlar sınıflandırma doğruluğu iyileştirebileceğini ve dengesizlik sorununu çözebileceğini göstermektedir (Dong, 2021).

Ağ trafiğindeki artışla birlikte yeni uygulamaların günlük dağılımı, ağ analizi ve izleme karmaşıklığında bir büyümeyi beraberinde getirmiştir. Labayen ve arkadaşları hem denetimli hem de denetimsiz öğrenmeyi kullanarak ağ trafiğinden kullanıcı etkinliklerini sınıflandırmak için bir sistem önermişlerdir. Sistem, ağ üzerinden sergilenen davranışı kullanır ve belirli bir zaman aralığında kullanıcı tarafından oluşturulan tüm trafiği dikkate alarak temel kullanıcı etkinliğini sınıflandırır. Sınıflandırma görevini gerçekleştirmek için üç katmanlı bir model önerilmiştir. Modelin ilk iki katmanı K-Ortalamalar algoritması kullanılarak uygulanırken, son katman etkinlik etiketlerini elde etmek için Rastgele Orman algoritmasını kullanır. Yapılan çalışma ile hizmet kalitesi için ağ trafiğinin çevrimiçi olarak sınıflandırılmasına ve önceki tekliflerden daha iyi performans gösteren kullanıcı profiline izin veren hassasiyet ve geri çağırma değerleriyle ortalama % 97.37'lik doğruluk elde edilmiştir (Labayen vd., 2020).

Trafik analizi, ağ operasyonlarının ve yönetiminin performansını ve güvenliği değerlendirmek gibi birçok amaca sahiptir. Bu nedenle, ağ trafiği analizi, ağların işleyişini ve güvenliğini iyileştirmek için hayati olarak kabul edilir. Alqudah ve arkadaşlarının çalışmasında, trafik analizi için farklı makine öğrenimi yaklaşımları tartışılmıştır. Artan ağ trafiği ve yapay zekanın gelişimi, izinsiz girişleri tespit etmek, kötü amaçlı yazılım davranışlarını analiz etmek ve internet trafiğini ve diğer güvenlik unsurlarını sınıflandırmak için yeni yollar arayışına girmiştir. Makine öğrenimi, ağ sorunlarını çözmede etkili yetenekler gösterir. Söz konusu çalışmada da trafik analizinde kullanılan tekniklerin bir incelemesi sunulmaktadır (Alqudah ve Yaseen, 2020).

Günlük (log) dosyaları, bir bilgisayar sisteminin durumu hakkında bilgi verir ve siber güvenlikle ilgili anormal olayların tespit edilmesini sağlar. Ancak karmaşık kaynaklardan toplanan büyük miktarlarda ve yapılandırılmamış günlük verilerinin otomatik olarak analiz edilmesi zordur. Bu nedenle, ağ trafiğindeki hareketlerin tahmin edilmesi için yapılan makine öğrenimi sınıflandırma yöntemlerinin uygulaması yapılmadan önce veri okunurluğunu artırmak ve anlamlı hale getirebilmek için kümeleme tekniklerine başvurulabilir. Landauer ve arkadaşları kümeleme teknikleri aracılığıyla günlük verilerini yoğunlaştıran veya özetleyen birkaç yöntem önermiştir. Bununla birlikte, belirli bir

uygulama alanı için doğru yaklaşımı seçmek, algoritmalar belirli hedeflere ve gereksinimlere göre tasarlandığından önemsiz bir durum olmayacaktır. Mevcut yaklaşımları kümeleme tekniklere göre gruplandırır, uygulanabilirliklerini ve sınırlamalarını gözden geçirir, eğilimleri tartışır ve boşlukları belirler. Genel bakış, filtreleme, ayrıştırma, imza çıkarma, statik aykırı değer tespiti, diziler ve dinamik anormallik tespiti şeklinde olmuştur (Landauer vd., 2020).

Olay günlüklerinin (event logs) çok miktarda veri içermesi sebebiyle, olay günlüklerinden desen madenciliği yapmak önemli bir sistem yönetimi görevidir. Vaarandi çalışmasında, günlük dosyalarından sık görülen kalıpları algılamaya, günlük dosyası profilleri oluşturmaya ve anormal günlük dosyası satırlarını tanımlamaya yardımcı olan günlük dosyası veri kümeleri için yeni bir kümeleme algoritması önermiştir (Vaarandi, 2003).

Kümeleme analizi yöntemi ile Shirwaikar ve arkadaşları tarafından yapılmış olan analizde ise kümeleme algoritması yönteminin, kümeleme sonuçlarını doğrudan etkileyeceği üzerine durulmuştur. Bu çalışmada standart k-ortalamalar kümeleme algoritması incelenmiş ve standart k-ortalamalar algoritmasının eksikliklerini analiz etmektedir. Yapılan çalışmanın aynı zamanda örüntü tanıma için verileri analiz etmekte web madenciliği ile k-ortalamalar algoritması yardımıyla örüntünün belirlendiği de belirtilmiştir (Shirwaikar ve Bhandari, 2013).

Ağ iz kayıtlarında anormal trafiği tespit etmek için farklı veri sınıflandırma teknolojileri ve metodolojileri tercih edilmektedir. Bu problem genellikle, sinyal pencerelerinde çıkarılan özelliklerle sınıflandırarak incelenir. "Anomaly Detection in Web Traffic Data using Artificial Immune Algorithms" adlı çalışmada, ağ üzerinde anormal web trafiğini tespit etmek için artificial immune systems'in negatif seçim algoritmasına dayalı bir yöntem önerilmiştir. Bu yöntem için Yahoo Webscope S5 veri setinden gerçek veriler kullanılmış ve veriler pencere kaydırma yöntemiyle pencerelere bölünmüştür. Bu deneysel çalışmada, negatif seçim algoritmalarının yapısındaki etkin detektörlerin sayısındaki değişimler ile web trafik verilerinde anormal trafik verilerinin tespiti yapılmıştır (Dandıl ve İlhan, 2019).

Trafik ağlarının izlenmesi ve kontrolü için veriler dikkate alınarak karar vermek önemlidir. Anomali tespiti izlemede çok önemli bir yere sahiptir. Trafik ağlarında

anomali tespit yaklaşımlarının olayları erken tespit ederek müdahale imkanı sağladığından daha önce bahsedilmişti. Literatürde anomali tespiti için sınıflandırma, kümeleme ve istatistiksel sömürü gibi yaklaşımlar mevcuttur. Bu alanda anomali tespiti için destek vektör makineleri, bayes ağları, bulanık mantık, genetik algoritmalar, karar ağaçları gibi birçok yöntem kullanılmaktadır. Trafik ağlarında anormallik tespiti adlı bir çalışmada, karar ağacı algoritması ile trafik ağlarında anormallik tespiti için bir yöntem önerilmiş ve önerilen yöntem bayes ağları ile karşılaştırılmıştır (Örnek vd., 2018).

KDD Cup99 veri seti, ağ anomali ve saldırı tespit sistemlerinde sıkça kullanılan bir veri kümesi olması nedeniyle yapay sinir ağlarının çok aşamalı uygulaması test edilmiştir. Paralel optimizasyonu ve ağ içerisinde gerçekleşen saldırıların tespiti amacıyla yapay sinir ağlarının paralel progralama ile performansı karşılaştırılmıştır. Bu çalışmada, paralel programlama süresinin etkisi analiz edilmiştir (Yıldırım vd., 2014).

NSLKDD veri seti kullanılarak filtre tabanlı öznitelik seçim yöntemlerinin anomali tabanlı ağ saldırı tespit sistemleri üzerindeki etkisinin incelendiği bir çalışmada, eğitim için NSLKDD veri setindeki KDDTrain20Percent veri seti, test için KDDTest veri seti kullanılmış ve sistem test edilmiştir. farklı bir veri seti ile eğitilmiş, başka bir test seti ile test edilmiştir. Sistemin güvenilirliğini kanıtlamak için bir çalışma yapılmıştır (Emhan ve Akın, 2019). Güvenlik duvarlarının işlevselliği, filtre kurallarına ve bu kuralların sırasına bağlıdır. Bir çalışma, güvenlik duvarları için politika anormalliği algılama algoritmasının deneysel uygulamasını inceledi ve kuralların doğru sırasını belirlemek için kurallar arasındaki tüm matematiksel ilişkileri dikkate aldı. Tek ve dağıtılmış güvenlik duvarı ortamları için anormallik algılama algoritmaları, Policy Anomaly Identifier adlı bir yazılım aracında uygulanır (Çetin vd., 2023).

Yapay sinir ağlarının saldırı tespit sistemlerinde kullanımına yönelik çalışmada, saldırı tespit sistemlerinde (IDS) kullanılan yapay sinir ağlarının (YSA) ağ üzerinden gerçekleşen saldırıları tespit etme yetenekleri incelenmiştir. Öncelikle, saldırı tespit sistemlerinin nasıl yapılandırıldığı ve hangi yöntemleri kullandıkları araştırılmıştır. Daha sonra, saldırı türleri ve özellikleri genel olarak tartışılmıştır. Bu kapsamda, çok katmanlı perceptron (MLP) yapay sinir ağı modeli kullanılarak, ağda anormal saldırılar oluşturan "Neptün" ve "ölüm pingi" saldırılarının tespit edilmesi amaçlanmıştır. Bu amaçla,

MATLAB programı kullanılarak yapay sinir ağı oluşturulmuş ve DARPA veri kümelerine dayalı veri setleriyle test edilmiştir (Tanrıkulu ve Sazlı, 2009).

Makine öğreniminin saldırı tespit sistemleri üzerindeki etkisini inceleyen bir diğer çalışma ise bilinen saldırı türleri ve sunucu tabanlı saldırı yöntemlerinden toplanan veriler kullanılarak bir saldırı tespit sistemi oluşturulmuştur. Oluşturulan veri seti, CesarFTP, WebDAV, Icecast, Tomcat, OS SMB, OS Print Spool, PMWiki, Wireless Karma, PDF N, Backdoored Executable, Browser Attack, Infectious Media Attack verilerinin birleştirilmesiyle oluşturulmuştur. Bu veri kümesi, Destek Vektör Makinesi (SVM) ve Naive Bayes (NB) yöntemleri kullanılarak sınıflandırılmış ve eğitilmiştir. Eğitilen sistemin SVM ile test edilmesi sonucunda 0,7129 başarı oranı elde edilmiştir. Daha sonra, yeniden boyut küçültme ve ana bileşen analizi uygulanarak Naive Bayes ile 0,7914 başarı oranı elde edilmiştir. Bu verilere dayalı olarak, eğitilen saldırı tespit sisteminin çalışır durumda iken gelen saldırıları %79 doğrulukla tespit edebildiği gösterilmiştir (Takaoğlu ve Özer, 2019).

Şu anda mevcut saldırı tespit sistemleri, esas olarak, imza tabanlı yaklaşım kullanarak dağıtılmış ağlardaki karakteristik olmayan sistem olaylarını belirlemeye odaklanmaktadır. Yeni saldırılar bulma sınırlaması nedeniyle, hem bulanık hem de anomali saldırılarını tespit edebilen geliştirilmiş bulanık ve veri madenciliği tekniklerine dayanan hibrit bir model önerilmektedir. Yapılan bir çalışmada işleme için tutulan veri miktarını azaltmak, yani öznitelik seçim süreci ve aynı zamanda veri madenciliği tekniğini kullanarak mevcut IDS'nin tespit oranını iyileştirmek amaçlanmıştır. Ardından, güvenlik kurallarını tanımlamanın ortak yollarını yansıtan if-then kurallarını oluşturmayı sağlayan bulanık kuralları uygulamak için APRIORI algoritmasının değiştirilmiş bir sürümünü geliştiren Kuok bulanık veri madenciliği algoritmasını kullanılmıştır. Daha hızlı karar almak için üç değişken girişli mamdani çıkarım mekanizmasını kullanılmış bulanık çıkarım motoru uygulanmıştır. Onerilen model, etkinliği için DARPA 1999 veri setine göre ¨ test edilmiş ve karşılaştırılmış ve ayrıca kampus içindeki "canlı" ağ ortamına karşı test edilmiş ve sonuçlar tartışılmıştır (Shanmugam ve Idris, 2009).

İnternet kullanımının artması ve güvenlik hususunun önemli hale gelmesiyle web madenciliği, web günlükleri ve web güvenlik duvarı gibi ürünlere ait günlük dosyalarındaki verilerin okunması, gelebilecek saldırılara karşı tahmin edilerek

yorumlanması konularında literatürde pek çok çalışma yer almaktadır. Günlük dosyalarındaki verileri anlamlandırabilmek için kümeleme yöntemleri kullanılırken anlamlı hale getirilmiş veriler üzerinde tahmin yapabilmek için ise sınıflandırma yöntemleri kullanılmaktadır.

Bu çalışmada veri ön işleme teknikleri uygulanmış ve temiz veri haline getirilmiş veri seti üzerinde sınıflandırma algoritmaları kullanılmıştır. Sonuçlar tablo şeklinde verilmiş ve algoritmaların tahmin üzerindeki etkileri sonuç bölümünde tartışılmıştır.

3. MATERYAL VE YÖNTEM

3.1. Ağ Cihazları ve Siber Güvenlik Kavramları

Ağ cihazları, bir ağ üzerinde veri iletimini sağlayan cihazları ifade eder. Bu cihazlar arasında router, switch, HUB, modem gibi cihazlar yer alır. Router, ağ üzerinde veri paketlerinin yönlendirilmesini sağlar ve ağ üzerinde farklı ağlar arasında veri iletimini sağlar. Switch ise, ağ üzerinde veri paketlerini ağ cihazları arasında yönlendirir ve ağın hızını arttırır. HUB ise, ağ üzerinde veri paketlerini birbirine bağlayan cihazdır. Modem ise, ağ üzerinde veri iletimini sağlayan cihazdır ve genellikle internet erişiminde kullanılır.

Siber güvenlik, bir ağ üzerinde yer alan cihazların ve verilerin güvenliğini sağlamaya yönelik önlemleri ifade eder. Siber güvenlik, ağ cihazlarının güvenliği ve ağ üzerinde yer alan verilerin güvenliğini sağlamaya yönelik önlemleri içerir. Örneğin, bir ağ üzerinde yer alan cihazlar için güvenlik duvarı kurulması ve ağ üzerinde yer alan veriler için şifreleme gibi önlemler siber güvenlik önlemleri arasında sayılabilir. Siber güvenlik, ağ cihazlarının ve verilerin güvenliğini sağlamaya yönelik önlemler sayesinde, ağ üzerinde yer alan cihazlar ve verilerin güvenliğinin sağlanması hedeflenmektedir.

Günümüzde, siber güvenlik konusu oldukça önem taşımaktadır. Özellikle internet erişimi olan cihazların sayısı artış gösterdiği için, ağ üzerinde yer alan cihazlar ve veriler için siber güvenlik önlemleri alınması önem kazanmıştır. Örneğin, bir ağ üzerinde yer alan cihazlar için güvenlik duvarı kurulması ve ağ üzerinde yer alan veriler için şifreleme gibi önlemler alınarak, ağ üzerinde yer alan cihazlar ve verilerin güvenliği sağlanır. Ayrıca, bir ağ üzerinde yer alan cihazların ve verilerin güvenliğini sağlamak için periyodik olarak güncelleştirme yapılması ve güncel güvenlik önlemlerine uygun bir şekilde kullanılması da önemlidir.

Ağ cihazları ve siber güvenlik kavramları, bir ağ üzerinde yer alan cihazların ve verilerin güvenliğini sağlamaya yönelik önlemlerdir. Ağ cihazları, veri iletimini sağlayan cihazları ifade ederken, siber güvenlik ise ağ cihazlarının ve verilerin güvenliğini sağlamaya yönelik önlemleri içerir. Bu yöntemler sayesinde, ağ üzerinde yer alan cihazlar ve verilerin güvenliği sağlanır ve ağ üzerinde yer alan cihazlar ve verilerin güvenliği korunur.

Büyük bir altyapıda bulunması gereken siber güvenlik sistemleri arasında şu sistemler yer alabilir:

Firewall: Bir ağ üzerinde kurulan bir güvenlik duvarıdır. Firewall, ağ üzerinde yer alan cihazların ve verilerin güvenliğini sağlamak için kullanılır. Firewall, ağ üzerinde yer alan trafiği kontrol eder ve belirli kurallara göre izin verilen veya engellenen trafiği ayırarak ağa saldırıları engellemeye yarar.

VPN (Virtual Private Network): Bir ağ üzerinde yer alan cihazlar arasında güvenli bir bağlantı oluşturmak için kullanılan bir teknolojidir. VPN, ağ üzerinde yer alan cihazlar arasında şifrelenmiş bir bağlantı oluşturarak verilerin güvenliğini sağlar.

IDS (Intrusion Detection System) / IPS (Intrusion Prevention System): Bir ağ üzerinde yer alan cihazların ve verilerin güvenliğini sağlamak için kullanılan bir teknolojidir. IDS, ağ üzerinde yer alan trafiği analiz eder ve ağa saldırıları tespit ederken IPS ise bu saldırıları engellemeye yarar.

WAF (Web Application Firewall): Bir web uygulaması üzerinde yer alan verilerin ve cihazların güvenliğini sağlamak için kullanılan bir teknolojidir. WAF, web uygulaması üzerinde yer alan trafiği kontrol eder ve belirli kurallara göre izin verilen veya engellenen trafiği ayırarak web uygulamasına saldırıları engellemeye yarar.

DDoS Firewall: Bir ağ üzerinde yer alan cihazların ve verilerin güvenliğini sağlamak için kullanılan bir teknolojidir. DDoS Firewall, ağ üzerinde yer alan trafiği kontrol eder ve DDoS (Distributed Denial of Service) saldırılarını engellemeye yarar.

Network Access Control (NAC): Bir ağ üzerinde yer alan cihazların güvenliğini sağlamak için kullanılan bir teknolojidir. NAC, ağ üzerinde yer alan cihazların hangi erişimlere izin verilmesini ve hangi erişimlere engel olunmasını kontrol eder.

E-mail Security: Bir ağ üzerinde yer alan e-posta sistemlerinin ve verilerin güvenliğini sağlamak için kullanılan bir teknolojidir. E-posta güvenliği, e-posta sistemleri üzerinde yer alan e-posta trafiğini analiz eder ve spam, phishing, malware gibi tehlikeli e-postaları engellemeye yarar. Ayrıca e-posta sistemleri üzerinde yer alan verilerin şifrelenmesi, yedeklenmesi gibi güvenlik önlemleri de e-posta güvenliği kapsamında yer alır.

Bu sistemler sayesinde, bir ağ üzerinde yer alan cihazlar ve verilerin güvenliği sağlanır ve ağ üzerinde yer alan cihazlar ve verilerin güvenliği korunur.

3.2. Veri Madenciliği

Kullanıcılar tarafından herhangi bir internet sitesi ziyaret edildiğinde ağ trafik verisi oluşur ve tüm güvenlik ürünlerinde örneğin sunucularda, log sistemlerinde yalnızca bir tane trafik için bile pek çok iz bırakılır. Yalnızca internet trafiğinde bu denli bir veri üretiliyorken dünya üzerindeki çeşitli iş kollarını düşündüğümüzde her dakika milyonlarca verinin oluştuğunu söyleyebiliriz. Kayıt altında tutulan veya veri tabanlarında öylece atıl vaziyette bulunan bu büyük verilerin incelenmesi başta pazarlama sektörü olmak üzere pek çok iş alanına kazanç sağlamaktadır.

Veriler arasındaki ilişkilerin incelenmesi yeniden pazarlama ve yeni müşteriler bulma aksiyonlarına büyük katkı sağlamaktadır. Akademik dünyada ise araştırma yapabilmek için veri setlerinin oluşturulması işleminin çok zahmetli olması sebebiyle hazır veri setleri ile madencilik çalışmaları yapılmaktadır. Buradan hareketle veri madenciliğinin ulaşılmak istenen hedefe gidebilmek için kullanılacak veriler üzerinde yapılması gereken birtakım işlemler olduğunu söylenebilir. Veri madenciliği yaparak büyük veriden bilinmeyen, geçerli ve uygulanabilir veriler elde edilebilir. Bu işlemler veri seti üzerinde çeşitli algoritmalar çalıştırılarak gerçekleştirilebilir. Ancak bu işlemi yapabilmek için veri seti olduğu gibi kullanılmamaktadır. Veri setini bir algoritma üzerinde kullanabilmek için bir takım veri ön işleme aşamalarından geçirmek gerekmektedir. Veri önişleme, veri madenciliğinin en önemli aşamalarındandır.

Veri madenciliği operasyonlarına başlarken öncelikle veri önişleme yapılır. Ön işleme fazında veri üzerindeki aykırı veriler, gereksiz veriler, işleme dahil olmayacak veriler gibi veri üzerinde temizleme yapılır. Daha sonra elde edilen temizlenmiş veri üzerinde eksik veri tespiti yapılır ve eksik veriler uygun bir yöntemle doldurulur veya silinir.

3.3. Makine Öğrenmesi

Makine öğrenimi, bilgisayar biliminin bir alt dalıdır ve yapay zeka arayışından tarihsel olarak ortaya çıkmıştır. Öğrenme konusu üzerine yapılan bazı araştırmalar, makinelere verilerin öğrenilmesi gerektiğini göstermiştir. Bu nedenle araştırmacılar bu problemleri çözmek için farklı simgesel yöntemler geliştirmişlerdir. 1990'larda tekrar ayrı bir alana dönüşen makine öğrenimi, pratik hayatta çözülebilir problemlerin yapay zekayı kullanarak çözümü amacıyla ortaya çıkmıştır. Makine öğrenimi ve veri madenciliği

genellikle benzer yöntemleri kullanır ve bu yöntemler büyük ölçüde örtüşür (Alkan, 2019).

Makine öğrenimi, veriler üzerinde tahminler yapabilen öğrenebilir algoritmaların çalışmasını ve inşasını inceleyen bir sistemdir. Bu tip algoritmalar, statik program talimatlarına sıkı sıkıya uymak yerine, verilerden bir model oluşturarak veriye dayalı tahminler ve kararlar alır.

Makine öğrenme, veri madenciliğine kıyasla daha fazla bilinen özelliklere dayalı olarak öğrenilen verilerden yapılan tahminlere odaklanır. Veri madenciliği ise bilinmeyen özelliklerin keşfedilmesine odaklanır. Bu süreç, veri tabanlarında bilgi keşfi analizinin adımlarından biridir. Veri madenciliği birçok makine öğrenimi yöntemini kullanır, ancak genellikle mantıksal olarak farklı hedefleri vardır. Öte yandan, makine öğrenimi, öğrenme doğruluğunu artırmak için veri madenciliği yöntemleri gibi denetimsiz öğrenme veya ön işleme adımını da kullanır (Alkan, 2019).

Makine öğreniminde denetimli öğrenme, denetimsiz öğrenme, yarı denetimli öğrenme, güçlendirilmiş öğrenme, yoğun öğrenme gibi kavramlar vardır. Bu kavramlar aşağıda sırasıyla açıklanmıştır.

Denetimli Öğrenme (Supervised Learning): Makine öğrenimi yöntemlerinden biridir. Bu yöntemde, veri kümesine etiketlenmiş veriler (yani, verilere eşlik eden doğru cevaplar) eşlik etmektedir. Öğrenme sürecinde, model, etiketlenmiş veriler aracılığıyla sınıflandırma veya regresyon gibi bir hedef değişkeni tahmin etmeyi öğrenir.

Denetimli öğrenme yöntemleri, veri kümesinde bir giriş değişkenleri ve hedef değişkeni olarak ayrılmış veriler kullanır. Giriş değişkenleri, modelin öğrenme sürecinde kullandığı özelliklerdir ve hedef değişken ise modelin tahmin etmeyi çalıştığı değişkendir. Örneğin, ev fiyatlarını tahmin etme problemi için giriş değişkenleri evin büyüklüğü, yaşı ve bölgesi olabilirken hedef değişken ise evin fiyatıdır.

Denetimli öğrenme problemleri iki kategoriye ayrılır: "Regresyon" ve "Sınıflandırma". Regresyon problemi, sonuçları sürekli bir çıktıda tahmin etmeye çalışır. Diğer bir deyişle girdi değişkenlerini bazı sürekli fonksiyonlara uydurmaya çalışmaktır denilebilir.

Sınıflandırma problemi, sonuçları ayrı çıktılarda tahmin etmeye çalışır. Başka bir deyişle, girdi değişkenlerini ayrı kategorilere atama girişiminde bulunulur (Alkan, 2019).

Regresyon ve Sınıflandırma arasındaki farkı örneklerle ifade etmeye çalıştığımızda aşağıdaki üç örnek incelenebilir.

Örnek 1:

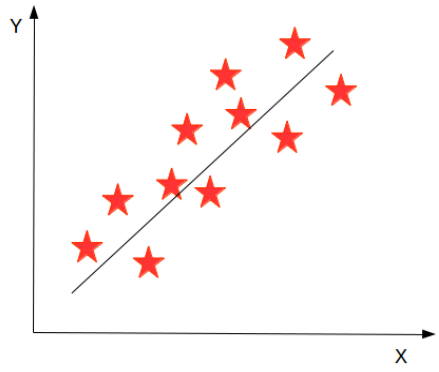
Otomotiv pazarındaki araçlara ilişkin veriler göz önüne alındığında, bu sorunu bir regresyon problemi olarak ele alabiliriz çünkü fiyat, fiyatları tahmin ederken büyüklüğün bir fonksiyonu olarak sürekli bir çıktıdır.

Ama arabaların tahmin edilen fiyatın altında mı yoksa altında mı satıldığını öğrenmek istiyorsak, problem artık bir sınıflandırma problemi olacaktır. Bu örnekte arabaları satış fiyatına göre iki ayrı kategoride inceleyebiliriz.

Örnek 2:

Bir insanın fotoğrafı verildiğinde, verilen fotoğrafı temel alarak yaş tahmini yapılması regresyon örneğidir. Şekil 1'de regresyon örneği verilmiştir.

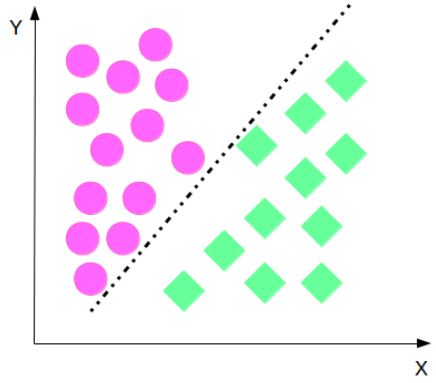
Örnek şekil:



Şekil 1. Regresyon örneği

Kaynak: Shawn (2022)

Tümörlü bir hasta söz konusu olduğunda ise tümörün iyi huylu olup olmadığını ön görmek sınıflandırma örneği olacaktır. Şekil 2'de sınıflandırma örneği verilmiştir.



Şekil 2. Sınıflandırma örneği

Kaynak: Shawn (2022)

Örnek 3:

A firmasının reklam harcamalarının satışları nasıl etkilediğini tahmin etmek yine bir fonksiyona bağlı olduğundan regresyon örneği olacaktır.

A firmasının posta kutusundaki spam ve spam olmayan e-postalarını ayırmak ise sınıflandırma örneğidir.

Özetle, denetimli öğrenme etkileşimli sistemlerden veri alır ve belirli bir düzende düzenler.

Denetimsiz Öğrenme (Unsupervised Learning): Makine öğrenimi yöntemlerinden biridir. Bu yöntemde, veri kümesine eşlik eden herhangi bir etiketlenmiş veri yoktur. Bu nedenle, modelin hedef değişkeni tahmin etmeyi öğrenmesi gerekmez. Bunun yerine, model veri kümesindeki özellikleri keşfetmeyi ve bu özellikler arasındaki benzerlikleri ve farklılıkları öğrenmeyi amaçlar.

Denetimsiz öğrenme yöntemleri, veri kümesinde sadece giriş değişkenleri bulunan veriler kullanır. Bu değişkenler, modelin öğrenme sürecinde kullandığı özelliklerdir ve model bu özellikler üzerinden veri kümesini keşfetmeyi ve anlamaya çalışır. Örneğin, bir veri

kümesinde insan vücut ölçüleri gibi özellikler bulunabilir ve model bu özellikler arasındaki benzerlikleri ve farklılıkları öğrenmeyi amaçlar.

Denetimsiz öğrenme yöntemleri, veri kümesinde bulunan veriler üzerinde keşif yapmayı amaçlar. Bu yöntemler, veri kümesindeki verileri gruplara ayırmayı (örneğin, kümeleme), veri kümesini birkaç parçaya bölmeyi (örneğin, parçalama) veya veri kümesinde bulunan gizli yapıları keşfetmeyi (örneğin, gizli Markov modelleri) amaçlar.

Denetimsiz öğrenme yöntemleri, veri kümesinde bulunan verilerin yapısını ve özelliklerini keşfetmeyi amaçlar, ancak bu yöntemlerin sonucu oluşan modeller genellikle daha az anlamlıdır ve kullanım alanları daha sınırlıdır. Bu yöntemler, veri keşfi, özellik seçimi ve veri ön-işleme gibi problemlerde kullanılabilir.

Denetimsiz öğrenme yöntemleri, veri kümesinde bulunan verilerin yapısını ve özelliklerini keşfetmeyi amaçlar, ancak bu yöntemlerin sonucu oluşan modeller genellikle daha az anlamlıdır ve kullanım alanları daha sınırlıdır. Bu yöntemler, veri keşfi, özellik seçimi ve veri ön-işleme gibi problemlerde kullanılabilir. Örneğin, bir veri kümesinde bulunan verilerin hangi özelliklerin benzer olduğunu keşfetmek için kümeleme yöntemi kullanılabilir.

Veri kümesinde bulunan özelliklerin hangilerinin modelin tahminlerinde daha önemli olduğunu keşfetmek için ise özellik seçimi yöntemi kullanılabilir. Veri kümesinde bulunan verilerin ön-işleme işlemine tabi tutulmadan önce hangi özelliklerin gereksiz olduğunu keşfetmek için ise parçalama yöntemi kullanılabilir.

Yarı Denetimli Öğrenme (Semi-supervised Learning): Makine öğrenimi yöntemlerinden biridir. Bu yöntem, veri kümesinde sadece bir kısmı etiketlenmiş veriler (yani, verilere eşlik eden doğru cevaplar) içerirken, bir kısmı da etiketlenmemiş verilerden oluşur. Bu yöntem, denetimli öğrenme yöntemlerine benzer şekilde, hedef değişkeni tahmin etmeyi öğrenmek amacıyla kullanılır. Ancak, veri kümesinde etiketlenmemiş verilerin de bulunması nedeniyle modelin öğrenme süreci biraz farklı şekilde işlemektedir.

Yarı denetimli öğrenme yöntemleri, veri kümesinde bulunan etiketlenmiş verileri kullanarak modelin öğrenme sürecine başlar. Ancak, etiketlenmemiş verilerin de bulunması nedeniyle, model bu veriler üzerinden de öğrenme sürecine devam eder.

Örneğin, bir sınıflandırma probleminde, model etiketlenmiş veriler üzerinden bir sınıf etiketini tahmin etmeyi öğrenir ve etiketlenmemiş veriler üzerinden de bu sınıf etiketini doğrulama veya düzeltmeyi çalışır. Böylece, model hem etiketlenmiş veriler hem de etiketlenmemiş veriler üzerinden öğrenme sürecine devam eder ve bu sayede daha başarılı tahminler yapabilir.

Yarı denetimli öğrenme yöntemleri, veri kümesinde etiketlenmiş verilerin az olduğu problemlerde kullanılabilir. Örneğin, sınıflandırma problemlerinde etiketlenmiş verilerin az olduğu durumlarda, modelin etiketlenmemiş veriler üzerinden de öğrenmesi sayesinde daha başarılı tahminler yapması beklenir. Aynı şekilde, regresyon problemlerinde de etiketlenmiş verilerin az olduğu durumlarda, modelin etiketlenmemiş veriler üzerinden de öğrenmesi sayesinde daha başarılı tahminler yapması beklenir.

Sonuç olarak, yarı denetimli öğrenme, veri kümesinde etiketlenmiş verilerin az olduğu durumlarda kullanılan bir makine öğrenimi yöntemidir. Bu yöntem, etiketlenmiş veriler üzerinden öğrenme sürecine başlar ve etiketlenmemiş veriler üzerinden de öğrenme sürecine devam eder. Bu sayede, model daha başarılı tahminler yapabilir. Yarı denetimli öğrenme yöntemleri, sınıflandırma ve regresyon gibi problemlerde kullanılabilir.

Takviyeli Öğrenme: Makine öğrenimi yöntemlerinden biridir. Bu yöntem, bir sistemin bir davranışını öğrenmesini ve bu davranışı uygun bir şekilde sergilemesini hedefler. Takviyeli öğrenme yöntemleri, bir agent (öğrenen sistem) ve bir çevre (sistemin bulunduğu ortam) arasında bir etkileşim süreci içerir. Agent, çevresine göre belirli davranışlarda bulunur ve bu davranışların sonucundan bir ödül veya bir ceza alır. Bu sayede, agent ödül ve ceza gibi geri bildirimler üzerinden davranışlarını öğrenir ve bu davranışları uygun bir şekilde sergilemeyi hedefler.

Takviyeli öğrenme yöntemleri, bir agent'ın bir amaç doğrultusunda bir davranışı öğrenmesini amaçlar. Örneğin, bir robotun bir sezgisel kontrol yöntemi öğrenmesi için takviyeli öğrenme yöntemi kullanılabilir. Bu sayede, robot ödül ve ceza gibi geri bildirimler üzerinden davranışlarını öğrenir ve bu davranışları uygun bir şekilde sergilemeyi hedefler.

Takviyeli öğrenme yöntemleri, genellikle Markov karar süreci (Markov decision process-MDP) adı verilen bir yapı üzerinden inşa edilir. MDP, bir agent'ın bir çevrede bulunmasını ve bu çevrede belirli eylemler yapmasını içeren bir süreçtir. Agent, çevresine göre belirli eylemler yapar ve bu eylemlerin sonucundan bir ödül veya bir ceza alır. Bu sayede, agent ödül ve ceza gibi geri bildirimler üzerinden eylemlerini öğrenir ve uygun bir şekilde sergilemeyi hedefler.

Takviyeli öğrenme yöntemleri, ayrıca bir agent'ın bir amaç doğrultusunda bir davranışı öğrenmesi sürecini modellemeyi amaçlar. Bu süreç, genellikle bir agent'ın bir çevrede bulunmasını ve bu çevrede belirli eylemler yapmasını içeren bir süreç olarak düşünülür. Bu yöntemler, bir agent'ın ödül ve ceza gibi geri bildirimler üzerinden davranışlarını öğrenmesini ve uygun bir şekilde sergilemeyi hedefler.

Takviyeli öğrenme yöntemleri, bir agent'ın bir çevrede bulunmasını ve bu çevrede belirli eylemler yapmasını içeren bir süreci modelleyerek, agent'ın ödül ve ceza gibi geri bildirimler üzerinden davranışlarını öğrenmesini ve uygun bir şekilde sergilemeyi hedefler. Bu yöntemler genellikle robotların sezgisel kontrol yöntemlerini öğrenmesi, bir sistemin hedefe doğru ilerlemeyi öğrenmesi gibi problemlerde kullanılabilir. Takviyeli öğrenme yöntemleri, genellikle Markov karar süreci adı verilen bir yapı üzerinden inşa edilir ve agent'ın çevresine göre belirli eylemler yapmasını ve bu eylemlerin sonucundan bir ödül veya ceza almasını içerir.

3.4. Sınıflandırma, Regresyon Tahmin

Sınıflandırma (classification) ve regresyon tahmini (regression prediction), makine öğrenimi yöntemlerinde kullanılan çeşitli tekniklerdir. Sınıflandırma, bir veri setinin belirli kategorilere göre sınıflandırılması sürecidir. Örneğin, bir veri setinde yer alan resimlerin insanlar veya hayvanlar olup olmadığı sınıflandırılabilir. Regresyon tahmini ise, veri setindeki değişkenler arasındaki ilişkiyi tahmin etme sürecidir. Örneğin, bir evin fiyatının evin büyüklüğü, yapım tarihi ve bulunduğu bölge gibi değişkenlere göre tahmin edilebilir.

Sınıflandırma ve regresyon tahmini, iki çeşit makine öğrenimi yöntemidir. Sınıflandırma yöntemleri, veri setlerini belirli kategorilere ayırarak sınıflandırır. Regresyon tahmini yöntemleri ise, veri setlerindeki değişkenler arasındaki ilişkiyi tahmin eder. Bu

yöntemler, veri setlerindeki verilerin özelliklerine göre kullanılır ve veri setlerinin özelliklerine göre farklı modeller kullanılır.

Sınıflandırma ve regresyon tahmini yöntemleri, veri setlerindeki verilerin özelliklerine göre farklı modeller kullanılır. Örneğin, veri setinde yer alan resimlerin sınıflandırılmasında farklı yöntemler kullanılır. Bunlar arasında destek vektör makine modelleri, k-en yakın komşu yöntemleri, Bayes teoremlerine dayalı yöntemler ve diğerleri yer alır. Regresyon tahmini yöntemleri ise, veri setindeki değişkenler arasındaki ilişkiyi tahmin etmek için farklı modeller kullanılır. Bunlar arasında lineer regresyon, polinom regresyon, logistik regresyon ve diğerleri yer alır.

Sınıflandırma ve regresyon tahmini yöntemleri, genellikle makine öğrenimi problemlerinin çözümünde kullanılır. Örneğin, bir sağlık veri setinde yer alan verilerin sınıflandırılmasında sınıflandırma yöntemleri kullanılabilir. Bu sayede, veri setinde yer alan veriler hastalık oluşma olasılığı gibi kategorilere göre sınıflandırılabilir. Regresyon tahmini yöntemleri ise, örneğin bir veri setinde yer alan evlerin fiyatlarının evlerin büyüklüğü, yapım tarihi ve bulunduğu bölge gibi değişkenlere göre tahmin edilebilir.

Sınıflandırma ve regresyon tahmini yöntemleri, veri setlerindeki verilerin özelliklerine göre farklı modeller kullanılarak veri setlerinin sınıflandırılması ve değişkenler arasındaki ilişkiyi tahmin etme süreçleridir. Bu yöntemler, makine öğrenimi problemlerinin çözümünde yaygın olarak kullanılır ve veri setlerinin özelliklerine göre farklı modeller kullanılır. Örneğin, sınıflandırma yöntemleri veri setindeki verilerin belirli kategorilere göre sınıflandırılmasını sağlar, regresyon tahmini yöntemleri ise veri setindeki değişkenler arasındaki ilişkiyi tahmin eder. Bu yöntemler sayesinde, veri setlerindeki veriler daha iyi anlaşılır ve veri setlerinin özelliklerine göre farklı problemlerin çözümüne yardımcı olunur.

3.5. Sınıflandırmada Hata Metrikleri

Makine öğrenmesi yöntemleri kullanıldığında modelin doğruluğunun ölçülmesi gerekmektedir. Kullanılan makine öğrenmesi modelinin doğruluğunun ölçülebilmesi için çeşitli metrikler bulunmaktadır.

Makine öğreniminde, bir veri setinden çıkartılan niteliklerden hangisinin hedef değişkene etki edeceği öngörülür. Örneğin, öğrenci notları için bir veri seti olsun. Bu veri seti

öğrenci adı, derslerden aldığı harf notları, derse harcanan zaman, ödev teslim oranı gibi nitelikleri içerir. Bu niteliklerden, derslerden aldığı harf notlarına etki edeceği düşünülen nitelik, lineer regresyon modeline ve geçme/kalma tahmini yaparken de lojistik regresyon modeline tabi tutulur. Bu niteliklerin etki ettiğine ilişkin öngörüler, hipotezlerdir ve bu hipotezlerin doğruluğunu ölçmek için sensitivity, specificity, F1 score ve p-value gibi değerler kullanılır. Doğruluk analizi yaparken bilinmesi gereken birtakım tanımlar bulunmaktadır. Covid19 test uygulamalarına dair bir veri seti olduğunu varsaydığımızda bu tanımlar sırasıyla aşağıdaki gibi örneklenerek açıklanabilir.

True Positive: Hastalığınızın olduğunu düşünüyorsunuz ve yapılacak testin pozitif çıkacağını tahmin ettiniz ve test sonucu gerçekten de pozitif çıktı böyle bir durumda gerçekte olan ve yapılan tahmin aynı olduğu için bu durum True Positive (TP) olarak adlandırılacaktır.

False Positive: Hastalığınızın olduğunu düşünüyorsunuz bu durumla ilgili yaptığınız tahmininiz pozitif ancak yapılan testin sonucu negatif çıktı. Böyle bir durumda yapılan tahmin tutmadığı için bu durum False Positive (FP) olarak adlandırılacaktır.

False Negative: Hastalığınızın olmadığını düşünüyorsunuz ve bununla ilgili tahmininiz negatif ancak yaptırdığınız testin sonucu pozitif çıktı tahmininiz tutmadığı için bu durum false negative (FN) olarak adlandırılacaktır.

True Negative: Hastalığınızın olmadığı düşünüyorsunuz ve bununla ilgili tahmininiz negatif ve yaptırdığınız testin sonucu da negatif yani tahmininizi tutturdunuz bu durum true negative (TN) olarak adlandırılacaktır.

Doğruluk: Modelin veri kümesinde doğru sınıflandırma oranını ifade eder. Bu, eğitim kümesinden oluşturulan modelin test kümesindeki verileri doğru sınıflandırma oranıdır. Denklem 1' de yer alan formülde gösterildiği gibi hesaplanmaktadır.

$$Doğruluk = \frac{DP + DN}{DP + DN + YP + YN} \quad (1)$$

Duyarlılık ya da Doğru Pozitif Oran (DPO): Sınıflandırıcının gerçekten pozitif olarak etiketlenen verileri doğru bir şekilde tahmin etme oranını verir. Denklem 2' de yer alan formülde gösterildiği gibi hesaplanmaktadır.

$$Duyarlılık = DPO = \frac{DP}{DP + YN} \tag{2}$$

Özgüllük ya da Doğru Negatif Oran (DNO): Gerçekten negatif sınıfa ait olan verileri doğru bir şekilde tahmin eden bir sınıflandırıcının oranını verir. Denklem 3' de yer alan formülde gösterildiği gibi hesaplanmaktadır.

$$\text{Özgüllük} = DNO = \frac{DN}{DN + YP} \tag{3}$$

Kesinlik ya da Pozitif Tahmin Değeri (PTD): Sınıflandırma sonucunda pozitif olarak tahmin edilenlerin ne oranda doğru olarak tahmin edildiğini gösterir. Örneğin, bir hastalık testinin sonucunun pozitif olarak tahmin edilmesi durumunda, kesinlik oranı testin gerçekten hasta olan kişileri doğru olarak tespit etme oranını verir. Bu değer Denklem 4'te gösterildiği gibi hesaplanır.

$$Kesinlik = PTD = \frac{DP}{DP + YP} \tag{4}$$

Negatif Tahmin Değeri (NTD): Bir sınıflandırma sonucunda negatif olarak tahmin edilenlerin gerçekte ne kadarının gerçekten negatif sınıfa ait olduğunu gösterir. Bu değer, Denklem 5'te belirtildiği gibi, veri kümesinden hesaplanır.

$$NTD = \frac{DN}{DN + YN} \tag{5}$$

F Ölçütü: Kesinlik ve anma ölçütlerinin bir arada değerlendirilmesiyle elde edilen bir değerdir. Bu ölçütün değeri, kesinlik ve anma değerlerinin harmonik ortalamasını alarak hesaplanır. F ölçütü genellikle F1, F0.5 ve F2 olarak üç farklı şekilde kullanılır ve 0 ile 1 arasında değerler alır. Bir sınıflandırıcının doğru tahminlerde bulunması durumunda, F

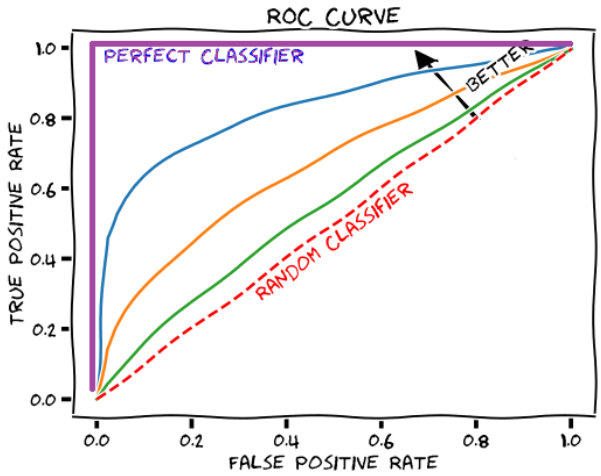
ölçütü değerinin 1'e yakın olması beklenir. Denklem 6’ da yer alan formülde gösterildiği gibi hesaplanmaktadır.

$$f_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{6}$$

Matthews Korelasyon Katsayısı (MCC): İki ve çok sınıflı sınıflandırma çalışmalarında performansı ölçmek için kullanılan bir yöntemdir. Bu yöntem diğer metriklerden farklı olarak, karışıklık matrisinde yer alan tüm değerleri kullanır. MCC, -1 ile +1 arasında değerler alır. Eğer bir sınıflandırıcı +1'e yakın değerler aldıysa, doğru tahminlerde bulunduğu anlamına gelir. 0 değerini aldığında ise rastgele tahminlerde bulunduğu ve -1'e yakın değerler aldığında yanlış tahminlerde bulunduğu anlaşılır. Denklem 7’ de yer alan formülde gösterildiği gibi hesaplanmaktadır.

$$MCC = \frac{DP \times DN - YP \times YN}{\sqrt{(DP + YP) \times (DP + YN) \times (DN + YP) \times (DN + YN)}} \tag{7}$$

Alıcı İşlem Karakteristikleri Eğrileri Altında Kalan Alan (AUC): Sınıflandırıcıların performansını ölçmek için yaygın olarak kullanılır. Bu metrik, 0 ile 1 arasında değerler üretebilir. AUC değerinin 1'e yakın olması, sınıflandırıcının doğru tahminler yaptığını gösterirken, 0.5 değerini alması sınıflandırıcının rastgele tahminler yaptığı anlamına gelir. Bu değerin altında kalması ise sınıflandırıcının doğru çalışmadığını gösterir. Şekil 3’ de ROC grafiği verilmiştir.

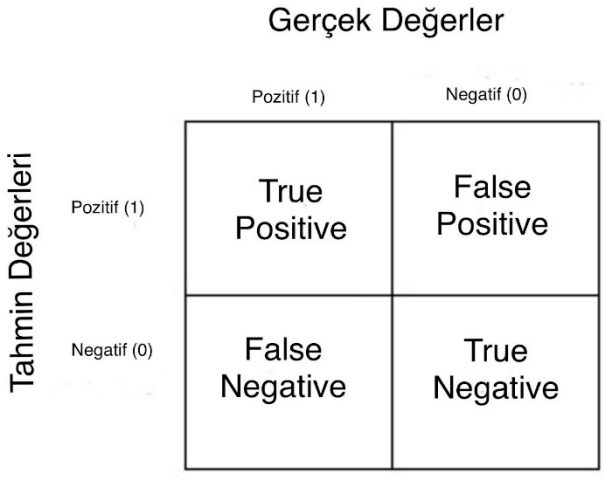


Şekil 3. ROC grafiği

Kaynak: Shawn (2022)

ROC (Receiver Operating Characteristic): modelin true positive oranıyla false positive oranı arasındaki ilişkiyi gösteren bir model olarak tanımlanabilir. AUC ise ROC eğrisinin altında kalan alanı verir. Bu değerler 0 ile 1 arasında değer alır ve 0 değerini aldığında tüm tahminlerin yanlış olduğu anlamına gelir. True positive rate, gerçekte durum pozitifse tahminlerin doğru pozitif olma oranını gösterirken, false positive rate, gerçekte durum negatifken tahminlerin yanlış pozitif olma oranını verir. ROC grafiğinde, 0.5, 0.5 noktasında, yani "random classifier" olarak bilinen yerde, sınıflandırma becerisi olmayan modeller yer alır. ROC grafiği üzerinde, sol üste doğru inildiğinde tahmin sayısı artar. Eğer negatif örnekler pozitif örnekleri baskılıyorsa, AUROC çok optimist olabilir. Bu nedenle, doğruluk analizi çalışmalarında öncelikle Confusion Matrix (karmaşıklık matrisi) incelenmelidir. Bunun sonucunda, gerektiğinde Precision ve Recall gibi metrikler daha kullanışlı olacağından incelemeye devam edilebilir.

Karmaşıklık Matrisi (Confusion Matrix): sınıflandırıcının doğru ve yanlış tahminlerini gösterir. Bu matris, sınıflandırıcı tarafından tahmin edilen sınıf ve gerçek sınıfın karşılaştırılmasını temsil eder. Bu, sınıflandırıcının performansını değerlendirmek için kullanılır. Karmaşıklık Matrisi, performansı değerlendirmede yararlı olan birçok metrik için taban verir. Örneğin, doğruluk (accuracy), pozitif doğruluk (precision) ve duyarlılık (recall) gibi metrikler, matris içindeki değerlerin birleştirilmesiyle hesaplanır. Örnek karmaşıklık matrisi Şekil 4'te gösterilmiştir.



Şekil 4. Karmaşıklık matrisi

Kaynak: Shawn (2022)

Karmaşıklık Matrisi (Confusion Matrix), veri sınıflandırma çalışmalarında modelin performansını değerlendirmeye yardımcı olan bir araçtır. Bu matris, gerçek değerlerin sınıflandırıcı tarafından tahmin edilen değerlerle karşılaştırılarak oluşturulur. Örneğin, bir hastalık testi için kullanılan bir sınıflandırıcıyı ele alalım. Karmaşıklık matrisi, gerçekte hasta olan kişilerin sınıflandırıcı tarafından tahmin edilen hasta sayısını (True Positive), gerçekte hasta olmayan kişilerin sınıflandırıcı tarafından tahmin edilen hasta sayısını (False Positive), gerçekte hasta olan kişilerin sınıflandırıcı tarafından tahmin edilen hasta olmayan sayısını (False Negative), ve gerçekte hasta olmayan kişilerin sınıflandırıcı tarafından tahmin edilen hasta olmayan sayısını (True Negative) gösterir. Karmaşıklık matrisi sadece dikotom (iki sınıflı) veriler için değil, çok sınıflı veriler için de kullanılabilir.

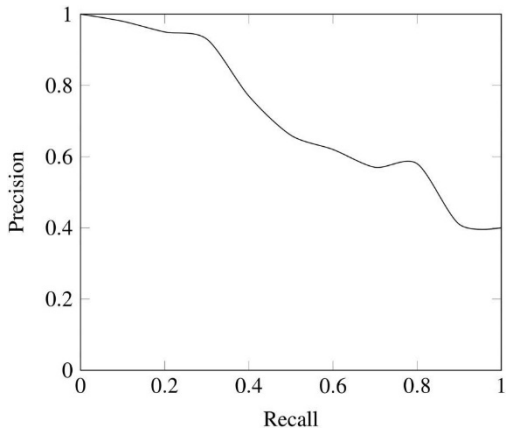
Recall: Bir sınıflandırıcının tahminlerinin doğruluğunu ölçmek için kullanılan bir metriktir. Özellikle false negatives (hasta olmadığı tahmin edilen ama gerçekten hasta olan insanlar) üzerinde odaklanır. Bu, hasta olmadığı tahmin edilen ancak gerçekten hasta olan insanlar olarak tanımlanır. Recall, ağır sonuçlar doğurabilecek olası hataların önüne geçmek için özellikle doktorlar tarafından kullanılır. Denklem 8'de yer alan formülde gösterildiği gibi hesaplanmaktadır.

$$Recall = \frac{Tru\ Positive}{True\ Positive + False\ Negative} \quad (8)$$

Precision: Bir sınıflandırıcının gerçekte pozitif olan örnekleri doğru tahmin etme oranını ölçer. Örneğin, bir hastalık testinin çok yüksek bir doğruluk oranı olmasına rağmen, sıklıkla hasta olmadığı halde pozitif olarak tahmin edilenler olabilir. Bu durumda, precision değeri düşük olacaktır ve sınıflandırıcının performansının düşük olduğu anlaşılacaktır. Denklem 9'da yer alan formülde gösterildiği gibi hesaplanmaktadır.

$$Precision = \frac{Tru\ Positive}{True\ Positive + False\ Positive} \quad (9)$$

Precision ile Recall ters orantılı metriklerdir. İki metrik ilişkisinde bir denge ayarlaması yapmak gerekmektedir. Bu ilişki Şekil 5'e gösterilmektedir.



Şekil 5. Precision ile recall ilişkisi

Kaynak: Shawn (2022)

Sınıflandırma performansını değerlendirmede yaygın olarak kullanılan bir metrik olan Karmaşıklık Matrisi (Confusion Matrix) paydada olan verilerin gerçek sınıfı ile tahmin sınıfının karşılaştırılmasını içerir. Bu matriste, satırlarda tahmin edilen pozitif ve negatiflerin sayıları yer alırken, sütunlarda gerçekte olan pozitif ve negatif çıktı sayıları bulunur. Sınıflandırma sonuçları sadece pozitif/negatif gibi dikotomik veriler dışında da kullanılabilir. Örneğin, hastalık testlerinde, testin %99,999 doğruluk oranı vaat etmesine rağmen, testin yine de güvenilir olmadığı görülebilir. Bu, sınıflandırma sonucunun "hasta" veya "hasta değil" olarak döndürülmesi ve verilerin çoğunluğunun "hasta değil" sınıfına ait olması nedeniyle olur. Bu durumda, Accuracy gibi hata metriklerinin yerine, Recall veya Precision gibi metrikler kullanılmalıdır. Recall, paydada olan false negative'leri inceler ve true positive'lerin gerçekten hasta olduğunu, false negative'lerin ise hasta olmadığı tahmin edilen ancak hasta olan insanları gösterir. Precision ise tahmin edilen pozitiflerin kaçının gerçekten pozitif olduğunu inceler. F1-Score ise Precision ve Recall'u bir arada değerlendirir ve bu iki metriğin harmonik ortalamasını verir. Bu, aritmetik ortalama alınmamasının sebebidir, çünkü Recall'un 1 ve Precision'ın 0 olduğu bir durumda (ya da tam tersi) ortalama 0.5 çıkar, bu da iyi bir görünüm vermez. Harmonik ortalama ise Recall'un 0.001 ve Precision'ın 0.999 olduğu bir durumda F1-Score'un 0.001 çıkmasını sağlar (0 olsaydı doğrudan 0 olurdu). Eğer Precision ve Recall arasında bir denge arıyorsak, F1-Score'un maksimum olduğu durumları incelemek gerekir.

F1-weighted: Sınıflandırma probleminde farklı sınıfların ağırlıkları farklı olabileceği durumlarda kullanılır. Örneğin, bir sağlık verisi setinde belirli bir hastalığın görülme sıklığı diğerlerine göre daha az olabilir. Bu durumda, o hastalık için tahminlerin doğruluğu diğer sınıflar için tahminlerin doğruluğuna göre daha önemli olabilir. Bu durumda, F1-weighted skoru kullanarak, o hastalık için tahminlerin doğruluğunun diğer sınıflar için tahminlerin doğruluğuna göre daha ağırlıklı olarak hesaplanması mümkündür.

F1-makro: F1-ölçütünün bir modifiye edilmiş halidir. F1-makro, bir etiketin hassasiyetini ve duyarlılığını tüm etiketler için ayrı ayrı hesaplar ve sonra bu değerleri ortalamak suretiyle hesaplar. Bu yöntemde her etiket için elde edilen F1-ölçütü değerleri hesaplandıktan sonra ortalama alınır. Bu sayede bir veri setinde az sayıda olan etiketlerin performansının daha belirgin bir şekilde gösterilmesi sağlanır.

F1-makro, performansın etiket bazlı ortalamasını hesaplamak için kullanılır ve her etiket için ayrı ayrı hesaplama yapar. Bu yöntemde her etiket için elde edilen F1-ölçütü değerleri hesaplandıktan sonra ortalama alınır. Bu sayede bir veri setinde az sayıda olan etiketlerin performansının daha belirgin bir şekilde gösterilmesi sağlanır.

F1-micro: Sınıflandırma modelinin tahminlerinin doğruluğunu ölçmek için kullanılan bir metriktir. F1-micro, precision ve recall değerlerinin harmonic ortalamasını hesaplar.

F1-micro, tüm sınıflar için precision ve recall değerlerini aynı ağırlıkta değerlendirir. Bu nedenle, F1-micro metriği, her bir sınıf için tahminlerin doğruluğunun eşit öneme sahip olduğu durumlarda tercih edilebilir. Ayrıca, F1-micro metriği, çok sınıflı sınıflandırma problemlerinde kullanılabilir ve tüm sınıflar için tahminlerin toplam doğruluğunu ölçer.

Top-k accuracy: Sınıflandırma modelinin tahminlerinin doğruluğunu ölçmek için kullanılan bir metriktir. Bu metrik, modelin verdiği tahminler arasından en yüksek olasılık değerine sahip k tane tahminin doğru olup olmadığını ölçer. Örneğin, top-3 accuracy değeri, modelin verdiği tahminler arasından en yüksek 3 tane tahminin doğru olup olmadığını gösterir. Top-k accuracy metriği, özellikle çok sınıflı sınıflandırma problemlerinde kullanılabilir. Bu metrik, modelin tahminlerinin doğruluğunu ölçmek için yalnızca en yüksek olasılık değerine sahip tahminleri değerlendirir, bu nedenle modelin diğer tahminlerinin doğruluğu ölçülmez.

Accuracy: Sınıflandırma modelinin tahminlerinin doğruluğunu ölçmek için kullanılan bir metriktir. Bu metrik, modelin verdiği tüm tahminlerin kaç tanesinin doğru olduğunu gösterir. Örneğin, bir sınıflandırma modelinin verdiği 1000 tahminin 900'ü doğru ise, accuracy değeri 0.9 olur. Accuracy metriği, modelin tüm tahminlerinin doğruluğunu aynı önemde değerlendirir ve herhangi bir sınıf için daha düşük performans göstermesi durumunda bu etki ortaya çıkar. Bu nedenle, accuracy metriği, her bir sınıf için tahminlerin doğruluğunun eşit öneme sahip olduğu durumlarda tercih edilebilir.

Balanced accuracy: Sınıflandırma modelinin tahminlerinin doğruluğunu ölçmek için kullanılan bir metriktir. Bu metrik, modelin tahminlerinin her bir sınıf için doğruluğunu ayrı ayrı hesaplar ve ortalama alır. Bu sayede, modelin bazı sınıflar için daha iyi performans göstermesi durumunda, diğer sınıflar için daha düşük performans göstermesi durumunda bu etki ortadan kaldırılır. Balanced accuracy metriği, çok sınıflı sınıflandırma

problemlerinde kullanılabilir ve her bir sınıf için tahminlerin doğruluğunun eşit öneme sahip olduğu durumlarda tercih edilebilir.

Jaccard score: İki kümenin benzerlik oranını ölçmek için kullanılan bir metriktir. Jaccard score, iki kümenin ortak elemanlarının sayısını, iki kümenin toplam eleman sayısına böler ve sonuç değerini verir. Bu değer, 0 ile 1 arasında değişir ve iki kümenin benzerliği arttıkça değer de artar.

Jaccard score, sınıflandırma problemlerinde de kullanılabilir. Örneğin, bir sınıflandırma modelinin verdiği tahminler ile gerçek sınıf etiketleri kümeler olarak düşünülebilir. Bu durumda, Jaccard score değeri, modelin tahminlerinin gerçek sınıf etiketleriyle ne kadar benzer olduğunu gösterir. Jaccard score değeri, modelin tahminlerinin doğruluğunu ölçmek için kullanılabilir ve herhangi bir sınıfın diğerlerine göre daha fazla önemli olduğu durumlarda tercih edilebilir.

Average precision: Sınıflandırma modelinin tahminlerinin doğruluğunu ölçmek için kullanılan bir metriktir. Average precision, modelin verdiği tahminlerin doğruluğunu true positive rate (TPR) ve false positive rate (FPR) değerlerine göre ölçer. TPR, modelin doğru tahminlerinin oranını gösterirken, FPR ise yanlış tahminlerin oranını gösterir. Average precision, TPR ve FPR değerleri arasındaki ilişkiyi ölçer ve modelin tahminlerinin doğruluğunu gösterir. Average precision, herhangi bir sınıfın diğerlerine göre daha fazla önemli olduğu durumlarda tercih edilebilir.

3.6. Kullanılan Sınıflandırma Algoritmaları

Makine öğrenmesinde kullanılan ve geleneksel yöntemler olarak literatürde yerini almış olan çeşitli sınıflandırma algoritmaları vardır. Bu algoritmalar güçlü ve zayıf yönlerine göre uygun modellerde tercih edilmelidirler.

Denetimli öğrenme yöntemi olan sınıflandırma, verinin önceden etiketlenmiş sınıflara ayrılmasını hedefleyen bir makine öğrenme tekniğidir. Bu işlem, veri setini eğitim ve test verisi olarak iki parçaya bölerek başlar. Eğitim verisi kullanılarak bir model oluşturulur ve test verisi kullanılarak bu model test edilir. Amaç, tasarlandığı gibi yeni bir örnek geldiğinde hangi sınıfa ait olduğunu doğru bir şekilde tahmin edebilmektir. Önemli sınıflandırma algoritmaları arasında k-en yakın komşu, karar ağaçları, naive bayes, destek

vektör makineleri, gradyan artırma, yapay sinir ağları, adaBoost, lojistik regresyon sınıflandırıcı ve rastgele orman algoritması bulunmaktadır (Alan vd., 2020).

3.6.1. K-NN Algoritması

K-en yakın komşu (K-NN) algoritması, bir veri kümesinde mevcut verilere dayanarak, veri kümesinde bulunmayan yeni bir verinin hangi sınıfa ait olduğunu tahmin etmeye yarayan bir denetimli öğrenme yöntemidir. Algoritma, verilerin birbirlerine olan uzaklıklarına ve benzerliklerine dayanarak sınıflandırma işlemi gerçekleştirir. K-NN algoritması, veri bir örüntü uzayında saklanır. Karara ait sınıf bilgisi bilinmeyen bir veri geldiğinde, verinin hangi sınıfa ait olduğunun belirlenmesi için, veri kendisine yakın olan en yakın k adet veriden hangisine daha çok benziyorsa, o veri sınıfına (Ayık vd., 2010).

K-en Yakın Komşu (K-NN) algoritması, denetimli öğrenme yöntemleri arasında sınıflandırma ve regresyon işlemlerinde kullanılan bir algoritmadır. Bu algoritma, eğitim aşamasına sahip olmadığı için büyük veri setlerinde tercih edilmesi pek uygun olmayabilir. Algoritma, yeni bir noktaya en yakın noktaları bulmak üzerine kuruludur ve K, bilinmeyen noktanın en yakın komşularının sayısını temsil eder. K-NN algoritması, girdi olarak verilen özellikler alanında eğitim örneklerinin en yakınını arar ve bu örnekler doğrultusunda sınıflandırma veya regresyon işlemi yapar.

K-NN (K-nearest neighbor), örüntü tabanlı öğrenme veya tembel öğrenme türü bir yöntemdir. Bu yöntemde, nesnelerin sınıflandırılması veya regresyon özelliklerinin tahmin edilmesi için, nesnenin yakın çevresindeki komşularının özellikleri kullanılır. Örneğin, bir nesnenin sınıflandırılmasında, o nesnenin en yakın komşularının çoğunluğunun hangi sınıfa ait olduğu dikkate alınır ve nesne o sınıfa atanır. Aynı şekilde, bir nesnenin özellik değerinin tahmin edilmesinde, o nesnenin yakın komşularının özellik değerlerinin ortalaması kullanılır. K-NN yönteminde, işlev yerel olarak yaklaştırılır ve tüm hesaplama sınıflandırmaya kadar ertelenir. Bu nedenle, bu yöntem "tembel" olarak adlandırılır. Ayrıca, komşuların katkılarına ağırlık verilerek, yakın komşuların ortalamaya daha uzak olanlardan daha fazla katkıda bulunmaları sağlanabilir. Örnek olarak, ortak bir ağırlıklandırma şeması, her komşuya 1/d ağırlığı verilmesini içerebilir; burada d, o komşuya olan uzaklıktır.

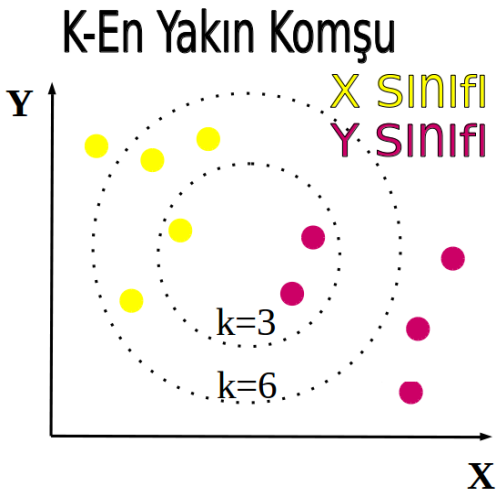
Komşular, sınıfın (kNN sınıflandırması için) veya nesne mülk değerinin (kNN regresyonu için) bilindiği birtakım nesnelerden alınır. Bu algoritma, verilerin yerel yapısına duyarlıdır ve eğitim verisi olarak adlandırılabilecek bir dizi nesne kullanır. Bu nesnelerin sınıfı veya mülk değeri bilinir. KNN algoritması, mevcut verilere benzerlik ölçüsü (örneğin, mesafe fonksiyonları) kullanarak yeni nesneleri sınıflandırır veya tahmin etmek için kullanılır. Örneğin, bir yeni nesne, en yakın komşularının çoğunluk oyuyla sınıflandırılabilir. Bu algoritma, 1970'lerin başında parametrik olmayan bir teknik olarak istatistiksel tahmin ve örüntü tanımada kullanılmıştır.

KNN yönteminde, mesafe ölçüsü olarak üç farklı mesafe kullanılabilir: Euclidean, Manhattan ve Minkowski mesafesi. Ancak, bu mesafeler yalnızca sürekli değişkenler için geçerlidir. Kategorik değişkenler (yani, sınıflandırılmış değişkenler) söz konusu olduğunda, Hamming mesafesi kullanılmalıdır. Ayrıca, veri kümesinde sayısal ve kategorik değişkenlerin bir karışımı varsa, 0 ile 1 arasındaki sayısal değişkenlerin standardizasyonu (ortalamalarının sıfır olması ve varyanslarının 1 olması) gerekebilir. Bu, KNN yönteminde mesafe ölçümünün daha doğru bir şekilde yapılmasını sağlar.

Verilerin incelenmesi öncelikli olarak, K değerinin en uygun seviyesini belirlemek için yapılmalıdır. Genellikle, yüksek bir K değeri daha hassas sonuçlar verebilir ancak bu, gürültünün azaltılmasını garanti etmez.

Çapraz doğrulama, K değerinin doğruluğunu test etmek için, tarihsel olarak, K değerinin optimal değerinin 3-10 arasında olduğu tespit edilen bağımsız bir veri kümesi kullanılarak yapılmıştır. Bu, 1NN'den daha iyi sonuçlar üretebilir. K değeri, bir algoritmanın çalışması sırasında belirlenir ve bu değer, bakılacak olan eleman sayısını belirtir. Gelen bir değer için, en yakın K eleman seçilerek, gelen değerle aralarındaki uzaklık hesaplanır. Uzaklık hesaplamalarında genellikle Öklid fonksiyonu kullanılır.

Uzaklık hesaplaması yapıldıktan sonra, elemanlar birbirlerine göre sıralanır ve gelen değer, en uygun olan sınıfa atanır. Bu hesaplamalar için, Öklid fonksiyonu gibi, alternatif olarak, Manhattan, Minkowski ve Hamming fonksiyonları da kullanılabilir.



Şekil 6. K-NN algoritması

Kaynak: Shawn (2022)

3.6.2. Decision Tree Algoritması

Karar ağaçları, veri setlerini sınıflandırma problemlerinde kolay yorumlanması ve bütünleştirilmesi nedeniyle sıklıkla tercih edilen bir yöntemdir. Bu algoritma, anlaşılır bir ağaç yapısına sahip düğümler ve dallardan oluşur ve her bir dal, bir olasılık durumunu temsil eder. Karar ağaçları, veriyi recursive olarak alt gruplara ayırarak böler ve bu ayırma aşamasındaki her bir dal bir kuralı temsil eder (Bozkır vd., 2009).

Karar ağaçlarında, ilk hücrelere 'kök' adı verilir. Bu kök hücresi, her bir gözlem için "Evet" veya "Hayır" olarak sınıflandırılmasını sağlar. Kök hücresinin altında, düğümler (ya da interval nodes) bulunur ve her bir gözlem bu düğümler aracılığıyla sınıflandırılır. Karar ağacının en altında yapraklar (ya da leaf nodes) bulunur ve bu yapraklar sonuçları verir. Karar ağacının düğüm sayısı arttıkça, modelin karmaşıklığı da artar.

Karar ağaçları, denetimli öğrenme algoritmaları içinde yaygın olarak kullanılan ve en tepeden aşağıya inen bir strateji sunan sınıflandırma algoritmalarıdır. Bu algoritmalar, bir dizi karar kuralı uygulayarak, çok sayıda kaydı içeren veri kümesini, daha küçük kümelere bölmek için kullanılır.

Karar ağaçları, basit karar verme adımlarının uygulanmasıyla, büyük miktardaki verileri çok küçük veri gruplarına ayıran bir yapıdır. Bu algoritmaların avantajları şunlardır:

• Anlaşılır ve yorumlanması kolaydır, çünkü kullanılan ağaç görselleştirilebilir.

• Veri hazırlığı gerektirmez, ancak unutulmaması gereken nokta, bu modelin kayıp değerleri desteklemediğidir.

• Kullanılan ağacın maliyeti, ağacı eğitmek için kullanılan veri noktalarının sayısına göre logaritmiktir.

• Hem sayısal hem de kategorik verileri işleyebilir.

• Çok çıktılı problemleri çözebilir.

• Bir modelin doğrulanması için istatistiksel testler kullanılabilir.

• Karar ağaçları, parametrik olmayan bir yöntem olarak düşünülebilir.

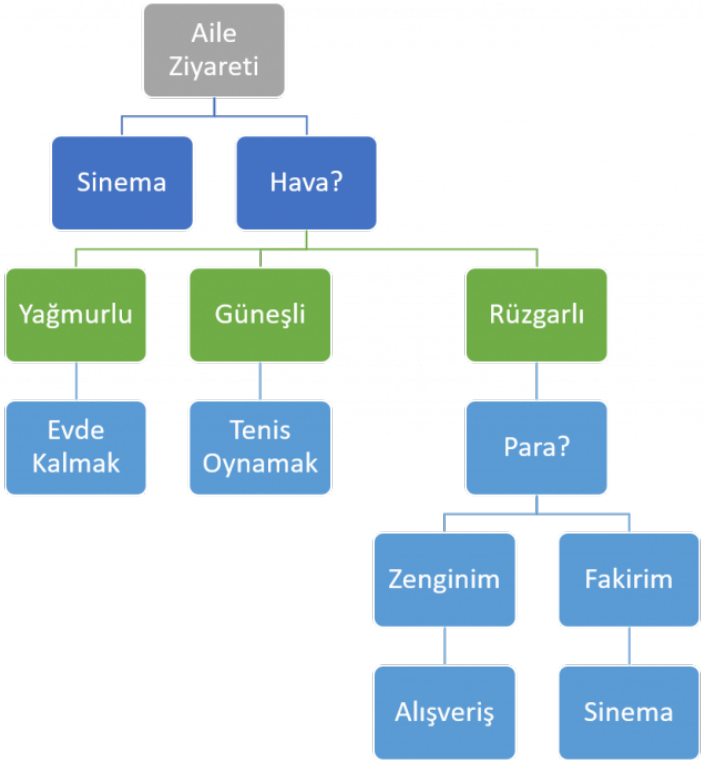
• Bu algoritmalar, veri dağılımı ve sınıflandırma yapısı hakkında bir yaklaşım sağlamaz.

Karar ağaçlarının dezavantajları şunlardır:

• Veriyi iyi açıklamayan, aşırı karmaşık ağaçlar üretebilir, bu durumda ağaç dallanması takip edilemeyebilir.

• Ezbere öğrenme yaşanabilir (over-fitting). Bu problemin çözümü için, model parametrelerine kısıtlamalar ve budama gibi yöntemler kullanılabilir. Budama, az sayıda nesneyi barındıran yaprak düğümlerin karar ağacı grafiğinden atılmasını ifade eder.

Karar ağacı algoritmaları, bir veri setini belirli kriterlere göre bölmeye yarayan yöntemlerdir. Bölünmenin nasıl gerçekleşeceği, ağacın doğruluğunu etkileyen önemli bir faktördür. Karar ağaçları, sınıflandırma ve regresyon problemleri için farklı bölünme kriterleri kullanır. Şekil 7'de örnek bir karar ağacı gösterilmiştir. Örneğin, Gini hesaplama yöntemi kullanılarak düğümler iki veya daha fazla alt düğümde bölünebilir. Alt düğümlerin oluşturulması, hedef değişkenlerin homojenliğini arttırır ve düğümlerin saflığını arttırır. Bu nedenle, algoritma seçimi hedef değişkenin tipine göre yapılır. Sıklıkla kullanılan algoritmalar arasında, kategorik değişkenler için Gini, Entropi ve Sınıflandırma Hatası; sürekli değişkenler için ise En Küçük Kareler Yöntemi sayılabilir.



Şekil 7. Karar ağacı algoritması

Kaynak: : Koyun (2020)

Entropi, veri setinde bulunan belirsizliğin bir ölçüsüdür. Sezgisel olarak, veri kümesinin tek bir etiketi (örneğin, tüm veriler aşı konusunda aynı durumda) olduğunda entropi düşük olur. Bu nedenle, verilerin entropiyi en aza indirecek şekilde bölünmesi gerekir. Bu bölünmeler ne kadar başarılı olursa, tahminler de o kadar doğru olacaktır.

Burada, p (x) bir grubun belirli bir sınırda yüzdesini ve H de entropiyi gösterir. Karar ağacı algoritmalarında, entropiyi en aza indirgemek için en uygun bölünmeler yapılması hedeflenir. Bu amaçla, bilgi kazancı kullanılır.

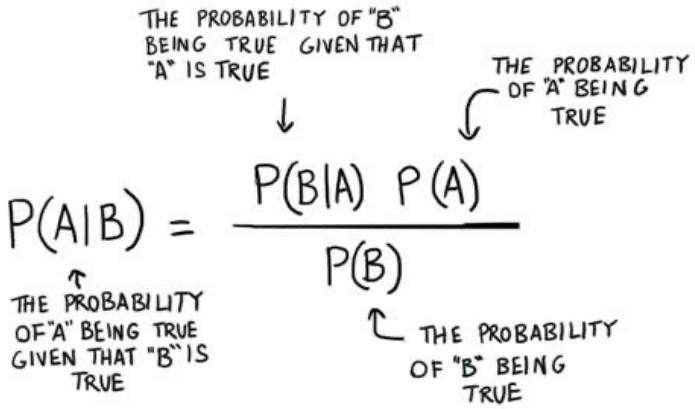
Veri seti, orijinal veri kümesi olan S'nin bölünmüş parçalarından oluşan D kümesine ayrılır. Bu parçalar, S'nin tüm alt kümeleri olan V'lerdir ve S'yi oluştururlar. Bu durumda, bilgi kazancı, bölünmeden önceki orijinal veri setinin entropisi ile her bir özniteliğin entropi değeri arasındaki fark olarak tanımlanır. Sınıflandırma problemlerinde, veri seti eğitim (train) ve test verilerine ayrılır. Algoritma, eğitim verilerini kullanarak model oluşturur ve bu model test verisi üzerinde uygulanarak modelin başarısı hesaplanır. Kökün ne olacağına belirlemek için kullanılan değerlerden bir tanesi Gini: Alt kümenin saflık değeridir.

Gini değeri, bir veri kümesinin sınıfları arasındaki benzerliğin ölçüsüdür. Bu değer 0 ile 1 arasında bir sonuç alır ve sonuç 0'a ne kadar yakınsa, o kadar iyi ayrım yapılmış olur. Gini değeri, her sınıf için hesaplanan Pj,j değerlerinin karelerinin toplamından çıkarılarak hesaplanır.

3.6.3. Naive-Bayes

Bayes teoremi, koşullu olasılık hesaplama formülüdür. Bu formül 1812 yılında Thomas Bayes tarafından keşfedilmiştir ve olasılık kuramı içinde önemli bir konudur. Bayes Teoremi, rassal değişken için olasılık dağılımı içinde koşullu olasılıklar ve marjinal olasılıklar arasındaki ilişkiyi gösterir. Bu algoritma, veri setinde bulunan sınıflandırılmış verileri kullanarak yeni bir verinin hangi sınıfa ait olduğu tahmin etmeyi amaçlar. Bayes teoremi, kolay anlaşılır ve uygulanabilir olması nedeniyle, diğer sınıflandırma algoritmalarına göre sıklıkla tercih edilir (Bozkır vd., 2009).

Bir sınıflandırma problemi, birçok özelliği ve bir sonuç (hedef) değişkeni olan bir veri setinden oluşur. Özelliklerin her birinin sonuç değişkenine etkisi incelenir ve hangisinin daha etkili olduğu tahmin edilir. Bu sınıflandırma problemlerinde, koşullu olasılık kavramı kullanılır. Koşullu olasılık, bir olayın gerçekleşme olasılığının, başka bir olayın gerçekleştiği durumda hesaplanmasıdır. Koşullu olasılık hesaplanması Şekil 8'de gösterilmiştir.



Şekil 8. Naive bayes algoritması

Kaynak: : Koyun (2020)

P (A | B) = B olayı gerçekleştiğinde A olayının gerçekleşme olasılığı,

P (A) = A olayının gerçekleşme olasılığı

P (B | A) = A olayı gerçekleştiğinde B olayının gerçekleşme olasılığı

P (B) = B olayının gerçekleşme olasılığı

Bu formül yardımıyla, A olayının gerçekleşme olasılığı B olayının gerçekleşmesi durumunda hesaplanır.

Bir sınıflandırma algoritması, veri seti içindeki her bir eleman için her bir sınıfın gerçekleşme olasılığını hesaplar ve en yüksek olasılık değerine sahip sınıfa göre elemanı sınıflandırır. Bu algoritma, eğitim verisi olarak çok az veri kullanıldığında bile çok başarılı sonuçlar verebilir. Ancak, test kümesinde bulunan bir elemanın eğitim kümesinde görülmemiş bir değere sahip olması durumunda, olasılık değeri olarak 0 verilerek tahmin yapılmaz. Bu durum, genellikle "Zero Frequency" olarak adlandırılır ve çözümü için düzeltme teknikleri kullanılabilir. Örneğin, Laplace tahmini gibi bir düzeltme tekniği kullanılabilir.

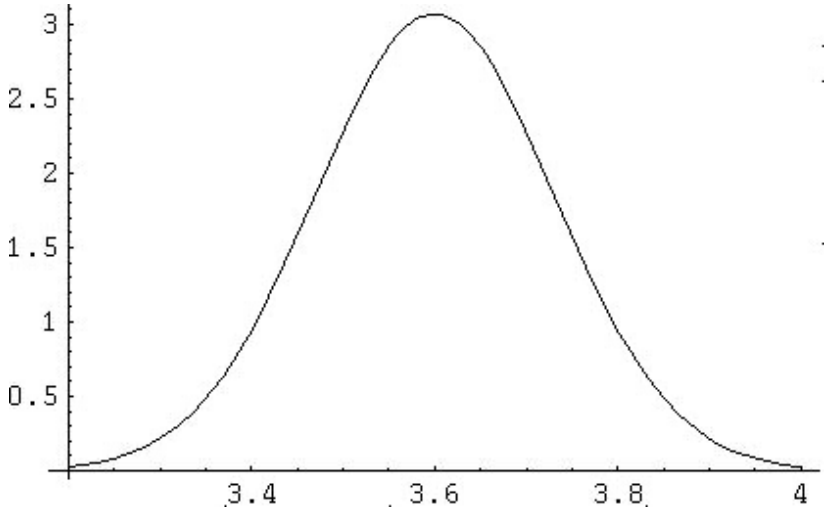
Naive Bayes sınıflandırma yöntemi, verilerin sınıflandırılması için kullanılan bir yöntemdir. Bu yöntemde, öğretim için sunulan verilerin mutlaka bir sınıfı/kategorisi bulunmalıdır ve bu verilere göre sisteme sunulan yeni test verilerinin hangi kategoride olduğu tespit edilmeye çalışılır. Öğretim verisi ne kadar çok ise, test verisinin gerçek kategorisinin tespit etme işlemi o kadar kesin sonuçlarla gerçekleşir. Naive Bayes sınıflandırma yöntemi, binary veya text gibi farklı veri tiplerinde kullanılabilir ve önemli olan veriler arasında nasıl bir oransal ilişki kurulabileceğidir. Bu yöntem, birçok farklı alanda kullanılabilir.

Naive Bayes sınıflandırıcısı, üç çeşidi olan Multinomial Naive Bayes, Bernoulli Naive Bayes ve Gauss Naive Bayes şeklinde çeşitleri bulunur.

Multinomial Naive Bayes, çoğunlukla belge sınıflandırma problemleri için kullanılır. Örneğin, bir belgenin hangi kategoride (spor, politika, teknoloji vb.) olduğu gibi sorulara cevaplar verir. Bu sınıflandırıcı, belgede bulunan kelimelerin sıklığını kullanır.

Bernoulli Naive Bayes, multinomial naive bayes benzeri ancak tahmin ediciler "Evet" ve "Hayır" şeklinde değerler alan boole değişkenlerdir. Örneğin, bir metinde bir kelime olup olmadığı gibi sorulara cevap verir.

Gauss Naive Bayes, tahmin ediciler sürekli bir değer aldıklarında ve ayrık olmadıklarında kullanılır. Bu değerler, Şekil 9'da gösterildiği üzere gauss dağılımından örneklenir.



Şekil 9. Gauss Dağılımı

Kaynak: Şeker (2011)

Naive Bayes sınıflandırıcıları, hızlı ve kolay uygulaması nedeniyle çeşitli sistemlerde sıklıkla kullanılır. Bu sistemler arasında, duygu analizi, spam filtreleme, öneri sistemleri, gerçek zamanlı tahmin, çok sınıflı tahmin ve metin sınıflandırma sayılabilir. Ancak, Naive Bayes sınıflandırıcılarının tahmincilerinin bağımsız olması gerektiği bir dezavantaj olarak görülür, çünkü gerçek yaşam durumlarında öngörücüler sıklıkla bağımlıdır ve bu da sınıflandırıcının performansını olumsuz etkileyebilir (Hatipoglu, 2018).

3.6.4. Linear Destek Vektör Makinesi

Son yıllarda, regresyon ve sınıflandırma problemlerinin çözümünde destek vektör makinelerinin yaygın olarak kullanılmaya başlandığına dikkat çekilebilir. Bu makineler, sınıflandırma işlemini yaparken yüksek düzeyde başarı sağlamak için yüksek boyutlu çekirdek fonksiyonları kullanırlar. Literatürde, destek vektör makinelerinin diğer

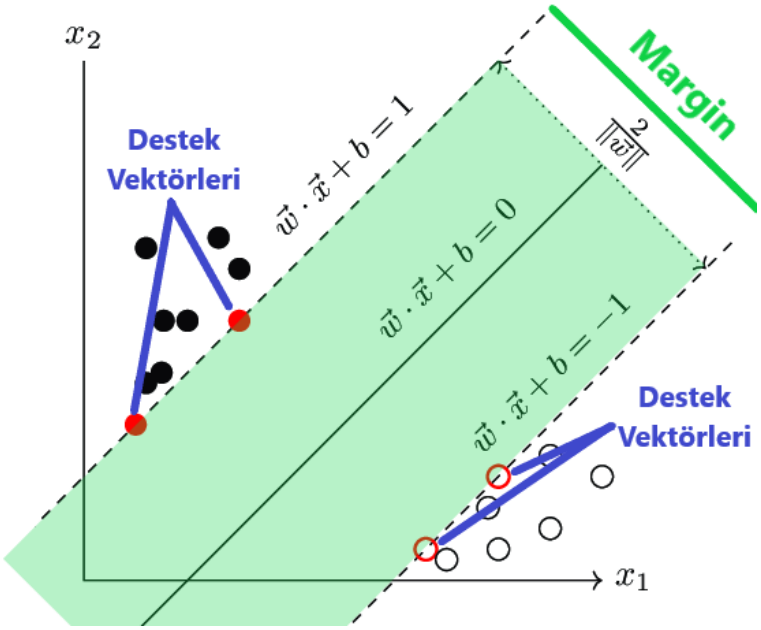
sınıflandırma algoritmalarına göre daha başarılı sınıflandırma yaptığına ilişkin çalışmalar yer almaktadır (Boser vd., 1992).

Destek vektör makineleri (Support Vector Machine, SVM), denetimli öğrenme yöntemlerinden biridir ve genellikle sınıflandırma problemlerinde kullanılır. SVM, bir düzlem üzerine yerleştirilmiş noktaları ayırmak için bir doğru çizer. Bu doğrunun, iki sınıfın noktaları için de maksimum uzaklıkta olmasını amaçlar. SVM, karmaşık ancak küçük ve orta ölçekteki veri setleri için uygundur (Melgani ve Bruzzone, 2004).

3.6.5. RBF Destek Vektör Makinesi

Sınıflandırma problemlerinde, iki farklı sınıf olarak siyahlar ve beyazlar bulunur. Bu problemlerde amaç, gelecek verinin hangi sınıfta olacağına karar verilmesidir. Bu karar verme sürecini gerçekleştirmek için, iki sınıfı birbirinden ayıran bir doğru çizilir ve bu doğrunun yanındaki alana "Margin" adı verilir.

Margin'in genişliği, iki veya daha fazla sınıfın nasıl iyi ayrıştırılacağını gösterir. Şekil 10'da örnek bir destek vektör makinesi gösterilmiştir.

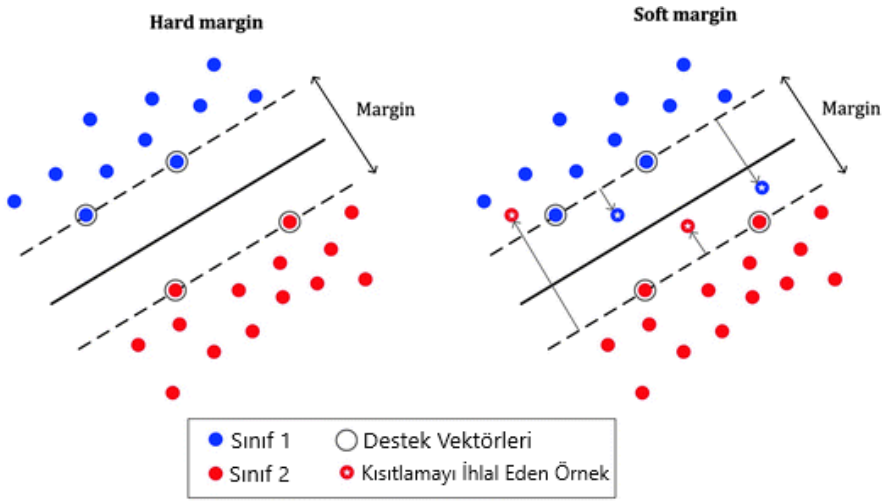


Şekil 10. Destek vektör makinesi

Kaynak: Akca (2020)

Eğer yeni bir değer için hesaplanan sonuç 0'dan küçükse, bu değer beyaz noktalara daha yakın olacaktır. Eğer sonuç 0'a eşit veya büyükse, bu değer siyah noktalara daha yakın olacaktır.

Margin değeri her zaman bu şekilde gösterilmeyebilir. Bazen veriler Margin bölgesine girebilir. Bu durum "Soft Margin" olarak adlandırılır. "Hard Margin" ise verilerin doğrusal olarak ayrıştırılabiliyor olması durumudur ve aykırı değerlere karşı duyarlıdır. Bu nedenle, duruma göre "Soft Margin" tercih edilebilir. Şekil 11'de marginlere örnek gösterilmiştir.



Şekil 11. Soft-Margin- Hard Margin

Kaynak: Akca (2020)

DVM içerisinde yer alan C hiper parametresi (Şekil 12), Margin'in daralmasını veya genişlemesini kontrol eder. C değeri ne kadar büyük olursa, Margin o kadar dar olur. Modelin aşırı öğrenmeye (overfit) düşmesi durumunda, C değerini azaltmak gerekmektedir.

Bazen düşük boyutlu veri setleri, karmaşık veri setlerini açıklamada yetersiz kalabilir. Bu durumda, boyutu artırmak işlemlerin artmasına neden olacağından tercih edilmez. Bu gibi durumlarda, Kernel Trick kullanılır. Kernel Trick, verileri belirli Kernel Fonksiyonları kullanarak daha anlamlı hale getirir. Yaygın olarak, Polynomial Kernel ve Gaussian RBF Kernel yöntemleri bu işlem için kullanılır (Akca, 2020).

Destek Vektör Makineleri (DVM), düzlem üzerindeki noktaların doğru veya hiperdüzlem ile ayrıştırılması ve sınıflandırılmasıdır.

• Küçük ve orta büyüklükteki veri setleri için uygundur ve ölçeklemeye (scale) duyarlıdır, bu nedenle veriler ölçeklendirilmelidir.

• Hard Margin ve Soft Margin arasındaki denge C hiperparametresi ile kontrol edilir. C değeri büyüdükçe, Margin daralır. Eğer model overfit olmuşsa, C değerini azaltmak gerekir.

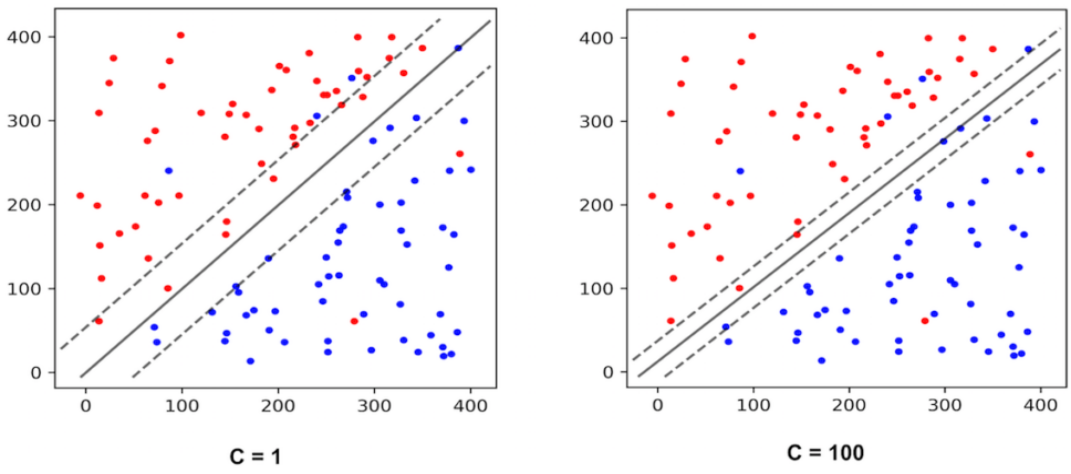
• İki boyutta açıklanamayan değişimleri fikir edinen hilelere Kernel Trick denir.

• İki boyutta açıklanamayan veri setlerini daha fazla boyutta açıklamak için kullanılan Kernel Trick metoduna Polynomial Kernel denir.

• Eğer model overfit olmuşsa, derecesi düşürülür ve eğer underfit olmuşsa derece yükseltilir. Coef0 hiperparametresi ile yüksek dereceli denklemlerden ne kadar etkilenebileceği ayarlanabilir.

• Her bir noktanın belirli bir noktaya ne kadar benzediğini normal dağılım ile hesaplayan ve buna göre sınıflandıran Kernel Trick metoduna RBF Kernel denir.

• Gamma değeri, dağılım genişliğinin kontrol edildiği değerdir. Gamma değeri ne kadar küçük olursa, dağılım o kadar geniş olur. Eğer model overfit olmuşsa, gamma değerini düşürmek ve eğer underfit olmuşsa gamma değerini yükseltmek gerekir.



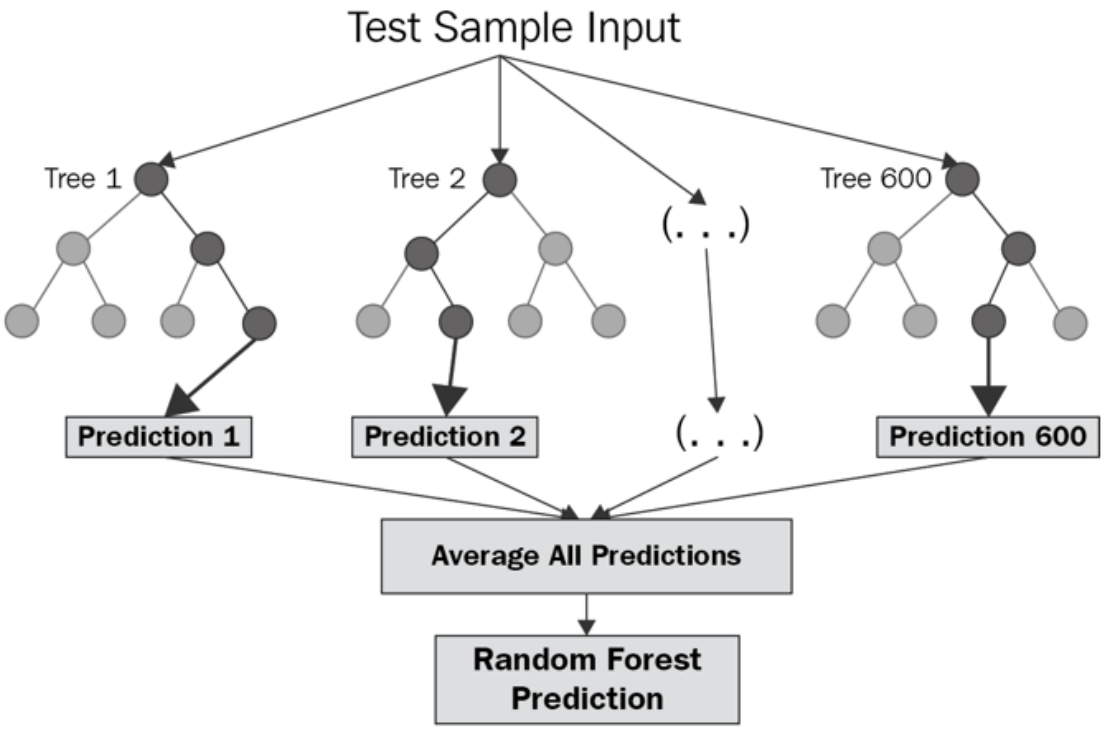
Şekil 12. C hiper parametresi

Kaynak: Akca (2020)

3.6.6. Random Forest Algoritması

Karar ağaçları, verilerin sınıflandırılması veya tahmin edilmesi için kullanılan bir yöntemdir. Bir karar ağacı, veri kümesinden öğrenilen bilgilere dayalı olarak, verilere göre karar vermek için kullanılan bir ağaç yapısıdır. Her bir düğüm, bir karar verme işlemini temsil eder ve her bir yaprak düğümü, bir sınıflandırma veya tahmin sonucunu temsil eder.

Random Forest, bir dizi karar ağacından oluşur ve her bir ağaç, veri kümesinden rastgele seçilen bir alt kümeyle eğitilir. Bu, her bir ağacın, veri kümesinin farklı bir kısmını öğrenmesine ve farklı karar verme yolları geliştirmeye olanak verir. Sonuç olarak, Random Forest, veri kümesinde bulunan düzensizlikleri ve öğrenme zorluklarını azaltarak, daha iyi tahminler yapmayı amaçlar.



Şekil 13. Rastgele orman ağacı algoritması

Kaynak: Şenol (2021)

Random Forest'ın birçok avantajı vardır. Öncelikle, karar ağaçlarının birleştirilmesiyle oluştuğu için, overfitting (aşırı uyum) riski düşüktür. Ayrıca, karar ağaçlarının her birinin farklı veri kümeleriyle eğitilmesi nedeniyle, Random Forest, veri kümesinde bulunan gürültüleri azaltır ve daha yüksek performans gösterir. Bunun yanı sıra, Random Forest, birçok değişkeni ve sınıflandırma problemlerini işleyebilir ve çok hızlı bir şekilde çalışır.

Random Forest, birçok farklı uygulama alanında kullanılabilir. Örneğin, sınıflandırma ve tahmin problemlerinde kullanılabilir ve sıklıkla iklim bilimleri, sağlık bilimleri ve finansal piyasalarda kullanılır. Random Forest örneği Şekil 13'te gösterilmiştir.

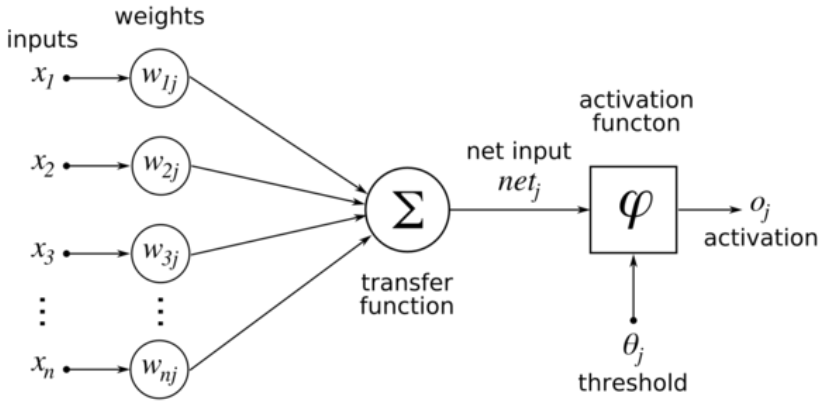
3.6.7. Neural Network Algoritması

Neural network (sinir ağı) yapay sinir ağı modelidir ve girdi verilerine göre bir çıktı üretmeyi hedefler. Bu model, biyolojik sinir hücrelerinden esinlenerek tasarlandı ve insan beyninin nasıl öğrendiğini modellemek amacıyla kullanılır.

Bir neural network, birkaç katmandan oluşur. Girdi katmanı, girdi verilerinin aldığı ilk katmandır. Bu veriler, sinir hücrelerine benzer "nöronlar" tarafından işlenir ve çıktı katmanına gönderilir. Çıktı katmanı, network'ün çıktısını veren son katmandır. Bu katman arasında birkaç "gizli katman" olabilir, bu katmanlar verileri işler ve çıktı katmanına gönderir. Bir neural network, verileri işlerken ağırlıklar ve eşik değerleri gibi parametreler kullanır. Bu parametreler, verileri işlerken kullanılan matematiksel formülleri belirler ve network'ün nasıl öğrendiğini etkiler. Network, veriler üzerinde eğitilir ve bu parametreler otomatik olarak ayarlanır, bu sayede network girdi verilerine göre çıktı üretebilir hale gelir. Neural network'ler, çeşitli uygulamalar için kullanılabilir.

Örneğin, görüntü tanıma, ses tanıma, metin çevirisi gibi uygulamalarda kullanılabilir. Ayrıca, tahminleme, sınıflandırma ve kategorizasyon gibi işlemlerde de kullanılabilir. Neural network'lerin eğitilmesi, bir optimizasyon problemidir ve genellikle "backpropagation" adı verilen bir yöntem kullanılarak gerçekleştirilir. Bu yöntem, network'ün hatasını azaltmayı amaçlar ve network'ün parametrelerini günceller. Bu sayede network, veriler üzerinde daha iyi bir performans gösterir. Neural network'ler,

günümüzde çeşitli uygulamalarda kullanılır ve bu uygulamalar genellikle çok sayıda veri işleme gerektirir. Şekil 14'te sinir ağı algoritma süreciyle ilgili bir örnek gösterilmiştir.



Şekil 14. Sinir ağı algoritması

Kaynak: Uslu (2016)

Bu nedenle, neural network'lerin çalışması için büyük işlem gücü ve bellek gereksinimi vardır. Bu nedenle, neural network'lerin eğitimi ve çalıştırılması için özel olarak tasarımı yapılmış bilgisayar donanımları kullanılır. Neural network'ler, eğitim aşamasında veri setleri kullanılır. Bu veri setleri, network'ün öğreneceği verileri içerir ve genellikle çok büyük olur. Veri setleri, network'ün öğrenme kapasitesini etkiler ve network'ün performansını da belirler.

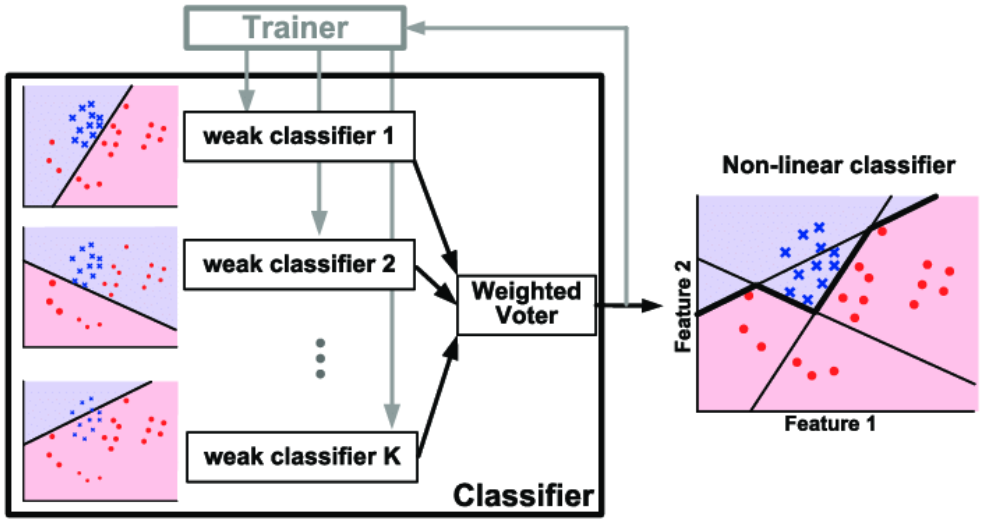
Neural network'ler, çeşitli katmanlardan oluşur ve bu katmanlar farklı işlevleri yerine getirir. Örneğin, bir katman sadece verileri işleyebilirken, diğer bir katman sadece sınıflandırma işlemi yapabilir. Bu nedenle, neural network'lerin tasarımı, uygulamanın amacına göre değişebilir.

Sonuç olarak, neural network'ler yapay sinir ağı modelleridir ve verileri işleyerek çıktı üretmeyi hedefler. Bu model, çeşitli uygulamalarda kullanılabilir ve eğitimi için büyük işlem gücü ve veri setleri gerekir. Neural network'lerin tasarımı ise uygulamanın amacına göre değişebilir.

3.6.8. AdaBoost Algoritması

Bu algoritma, bir dizi sınıflandırıcıyı (çoğunlukla karar ağaçları) birleştirir ve sonuç olarak daha güçlü bir sınıflandırıcı oluşturur.

AdaBoost, her bir sınıflandırıcı için bir ağırlık değeri atar ve bu ağırlık değerlerine göre sınıflandırıcıların tahminlerini birleştirir. Ağırlık değerleri, her bir sınıflandırıcının doğruluk oranına göre belirlenir, ancak daha az doğru tahminler daha yüksek ağırlık değerine sahip olur. Bu sayede, AdaBoost algoritması, daha doğru tahminler yapan sınıflandırıcıların etkisini daha fazla azaltır ve daha az doğru tahminler yapan sınıflandırıcıların etkisini daha fazla artırır.



Şekil 15. AdaBoost algoritması

Kaynak: (Wang vd., 2015)

AdaBoost algoritması, düşük performanslı sınıflandırıcıları iyileştirmeyi amaçlar ve genellikle diğer makine öğrenimi algoritmalarından daha iyi performans gösterir. Bununla birlikte, AdaBoost algoritmasının bazı dezavantajları da vardır. Örneğin, algoritma özelleştirilmiş hiperparametreler gerektirir ve öğrenme hızı diğer algoritmalara göre daha yavaş olabilir. Ayrıca, veri setinde sınıf dengesizliği olsa bile, AdaBoost algoritması sınıflar arasındaki dengesizliği dikkate almaz ve bu durumda yanlış tahminleri daha fazla önemli hale getirebilir.

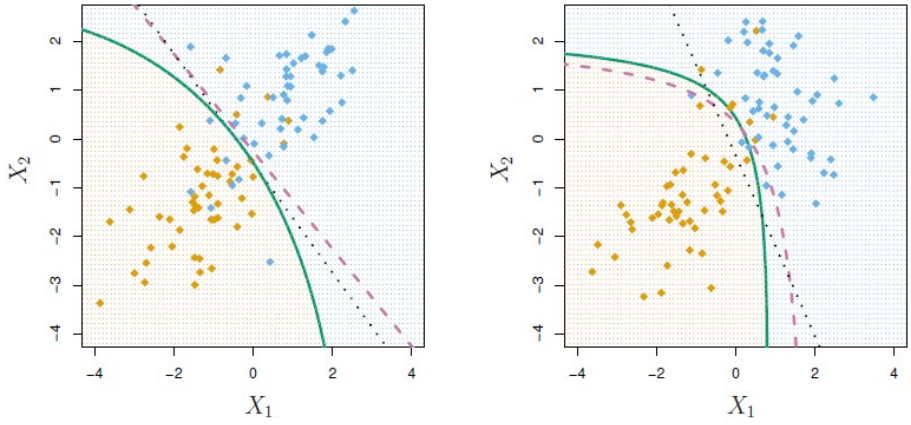
AdaBoost algoritmasının kullanımı oldukça yaygındır ve çeşitli uygulamalar için kullanılabilir. Örneğin, görüntü sınıflandırma, metin sınıflandırma ve müşteri satın alma davranışı tahmini gibi problemlerde kullanılabilir. Şekil 15'te örnek bir AdaBoost öğrenimi gösterilmiştir.

3.6.9. QDA (Kuadratik Diskriminant Analizi) Algoritması

Bu algoritma, veri kümesinin sınıfları arasında ikili (2 sınıflı) ya da çoklu (çok sınıflı) sınıflandırma yapmak için kullanılır. QDA algoritması, bir önceki sınıflandırma algoritması olan LDA (Linear Discriminant Analysis) algoritmasına benzer, ancak LDA algoritmasında ortalama değerler kullanılırken, QDA algoritmasında ortalama değerlerin yanı sıra kovaryans matrisleri de kullanılır.

QDA algoritması, sınıflandırma işlemi için iki adım kullanır. İlk olarak, veri kümesinin sınıfları arasındaki farklılıkları ölçmek için bir "diskriminant fonksiyonu" oluşturulur. Bu diskriminant fonksiyonu, veri kümesindeki her bir örnek için sınıflandırılması gereken sınıfı tahmin etmek için kullanılır. İkinci adımda ise, veri kümesindeki her bir örnek için oluşturulan diskriminant fonksiyonu kullanılarak sınıflandırma yapılır.

QDA algoritması, LDA algoritmasına göre daha doğru tahminler yapabilir ancak daha yüksek bir maliyeti vardır. Bu nedenle, QDA algoritması genellikle LDA algoritmasından daha küçük veri kümelerinde kullanılır. Ayrıca, QDA algoritması LDA algoritmasına göre daha hassas olabilir ve daha fazla özelleştirme seçeneğine sahiptir. Şekil 16'da ayrım işleminin yapıldığı eğri görseli verilmiştir.



Şekil 16. QDA algoritması

Kaynak: Datacadamia (2017)

QDA algoritmasının avantajları:

-LDA algoritmasına göre daha doğru tahminler yapabilir.
-Daha fazla özelleştirme seçeneğine sahiptir.

QDA algoritmasının dezavantajları:

-Daha yüksek bir maliyeti vardır.
-Genellikle LDA algoritmasından daha küçük veri kümelerinde kullanılır.

QDA algoritmasının uygulama alanları:

-Özellikle ikili sınıflı sınıflandırma problemlerinde kullanılabilir.

-Örneğin, bir banka müşterilerinin kredi riski tahmininde kullanılabilir.
-Ayrıca, sağlık alanında hastalık teşhisinde de kullanılabilir.

QDA algoritması LDA algoritmasına göre daha doğru tahminler yapabilen ancak daha yüksek bir maliyeti olan bir sınıflandırma algoritmasıdır. Bu algoritma ikili ve çoklu sınıflı sınıflandırma problemlerinde kullanılabilir ve özellikle ikili sınıflı sınıflandırma problemlerinde iyi sonuçlar verir.

3.7. Veri Önleme Aşamaları

Bu çalışmada gerçek bir WEB sitesine ait Cloud WAF sisteminden farklı zamanlara ait veriler alınmıştır. WAF cihazından temin edilen veriler JSON formatında dışarıya aktarılmıştır. Aktarılan veri üzerinde çeşitli düzenleme ve veri temizleme işlemleri yapılmıştır. Bu işlemler ilk olarak veri seti üzerinde kolay işlem yapabilmek adına verinin Microsoft Excel platformuna aktarılması ve ardından modeli eğitmek için gerekli olmayan verilerin veri setinden çıkartılması şeklinde düzenlenmiştir. Veri seti Microsoft Excel üzerinde sırasıyla DateTime, Method, UserAgent, SourceIP, DestinationDomain, Score, Status kolonları olacak şekilde düzenleme sağlanmıştır.

3.7.1. Veri Kümesinin Oluşturulması

Veri madenciliği çalışmalarında etkili bir sonuç alabilmek ve veri setinden bilinmeyen, faydalı verileri tespit edebilmek için ham veri kümesi olduğu gibi kullanılamaz. Veri seti herhangi bir makine öğrenmesi veya yapay zekâ temelli öğrenme modeli ile uygulanmadan önce bir takım iyileştirme çalışmaları yapılmalıdır.

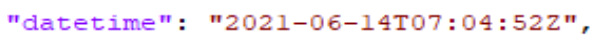
Bu çalışmalar verinin daha iyi tanınmasına ve uygulanacak model ile ilişkisinin saptanmasına yardımcı olacağı gibi aynı zamanda veride henüz tespit edilmemiş pek çok faydalı parametreyi de göz önüne serecektir. Tüm bunlar olduktan sonra veri işleme daha başarılı olacak ve uygulanan model başarımı etkili sonuçlar verecektir.

Bu bilgiler ışığında veri seti üzerinde yapılması gereken bir takım iyileştirme (veri ön işleme) çalışmaları bulunur. Bu çalışmaların tamamlanmasının ardından sınıflandırma modeline uygulanacak temiz bir veri seti elde edilmiş olunur. Yapılan kontroller ve düzenlemeler sonucu veri seti yukarıdaki başlıkta da belirtildiği gibi yedi kolondan oluşan son halini alır.

3.7.2. Veri Temizleme ve Aykırı Veri Tespiti

Veri seti henüz MS Excel üzerindeyken filtre özelliği ile “NaN”, “None”, “Null” ve boşluk gibi değerlerin varlığı kontrol edilir. Bu değerlere sahip hücreler tespit edilir. Bu verilerin kaldırılması yerine bir sonraki adımda gösterildiği gibi eksik verilerin tamamlanması işlemi gerçekleştirilir.

Şekil 17’de veri seti üzerinde “DateTime” kolonunda tarih ve saat veri değerlerinde tarih ve saati ifade eden değerlerden farklı olarak tarih ve saat arasında “T” ve saatin son hanesinde “Z” karakterinin olduğu görülmektedir.



Şekil 17. Veride istenmeyen karakterler

Bu değerler veri setinden silinmiştir. Buna ek olarak veri setinde “UserAgent” kolonundaki bazı değerlerde modeli eğitirken sorun olabileceği düşünülen “,” – “+” gibi değerler üzerinde de temizleme yapılmıştır.

Veri temizleme adımlarında son olarak yapılan işlem ise veri seti üzerindeki “Domain” ve “DestinationIP” kolonlarındaki verilerin gerçek veriler yerine anonimleştirilen veriler ile değiştirilmesi işlemi olmuştur. Bu işlem ile gerçek bir üniversite verilerini kullanmak yerine alan adları ve ip adresleri gerçek bir durum tespit edilmeyecek şekilde güncellenmiştir. Alan adı “example.com” olarak alt alan adları ise konusunda göre örneğin ödeme sistemine ait web sitesinin alt alan adı “ab.example.com” olarak değiştirilmiştir.

Veri seti üzerindeki temel temizleme işlemi bu şekilde gerçekleştirildikten sonra eksik verilerin tamamlanması fazına geçmeden önce aykırı verilerin de tespit edilip kaldırılması veya eksik veri tamamlama aşamasında bu verilerin de tamamlanması işleminin yapılması gerekmektedir. Aykırı veri tespitinde istatistiksel yöntemlerde “5 sayı özeti”, grafiksel yöntemlerde ise “box-plot”, “Q-Q grafiği”, “histogram” gibi yöntemler bulunmaktadır. Ancak WAF üzerinden gelen IP skoru bilgisinin alt sınırının 0, üst sınırının ise 100 olduğu bilindiği için MS Excel üzerinde küçükten büyüğe sıralanmış veri kolonu üzerinde negatif ve 100’den büyük değerlerin temizlenmesi bu kolon için aykırı veri tespiti yapılmış olması anlamına gelmektedir.

3.7.3. Eksik Verilerin Tamamlanması

Eksik veriler, veri kümelerinde değeri olmayan, bilinmeyen değerlerdir. Eksik veriler veri setlerinde bulunması, bilgi keşfi operasyonlarında sorunlara neden olur. Çoğu veri madenciliği yöntemi, eksiksiz verilerle çalışmaya dayanır, bu yüzden veri setinde işlem yapmadan önce eksik veriler tespit edilmeli ve veri setinden çıkartılmalı veya başka bir yöntemle doldurulmalıdır.

Tek değişkenden kaynaklanan eksik veriler (Şekil 18.a), tek bir öznitelikte meydana gelen eksik verilerdir. Bu tür eksik veriler, elektronik sensör hatalarında sıklıkla görülür. Monoton eksik veri yapısı (Şekil 18.b), bir değerin hesaplanamaması sonucu diğer değişkenlerin de tespit edilememesiyle oluşur. Bu tür eksik veriler, medikal veri setlerinde sıklıkla görülür. Rastgele eksik veri yapısı (Şekil 18.c), veri kümesinde rastgele pozisyonlarda eksik verilerin olmasıyla oluşur. Bu tür eksik veriler, sistematik bir tutarlılık göstermeyen durumlarda ortaya çıkabilir.

Şekil 18. Eksik veri yapısı

Veri ön işleme aşamalarının en önemli adımlarından bir tanesi olan eksik veri tamamlanması aşamasında yine eğitim modelinde tahmin yaparken baz alınan kolonda eksik veri bulunmasının sorun teşkil edebileceği düşünüldüğünden "Score" kolonu üzerindeki eksik veriler incelenmiştir.

Nihai tahmin edilmiş veri seti inceleneceği zaman kolonlarda eksik veri görüntüsü olmaması için hesaplamalarda ve eğitim modeline aktarımda bir işlevi olmamasına rağmen alan adı (Domain) kolonu da incelemeye dahil edilmiştir.

Eksik verilerin tamamlanması işleminde Lineer Regresyon ve YSA temelli veri doldurma yöntemleri sıklıkla kullanılır. Başka kullanılan yöntemler de şunlardır: alt, üst veya sabit değerlerle doldurma, ortalama, medyan ve tepe medyan değerleri tespit edilerek doldurma, kategorik verilerle doldurma ve lineer regresyon yöntemiyle doldurma. Lineer regresyon modeli, regresyon modelinin eksik verilere uyarlanmış halidir. Eksik verilerin yer aldığı kolon bağımlı değişken olarak, diğer kolonlar ise bağımsız değişken olarak tanımlanır ve bu şekilde eksik veriler tahmin edilmeye çalışılır. Ancak, bu yöntemin dezavantajı, eksik verilerin regresyon ile diğer değişkenlerle doldurulması nedeniyle asıl tahmin edilecek değişken üzerinde veri setinde aşırı öğrenme (overfitting) olasılığıdır.

Python programlama dili ile eksik verilerin tahmininde kullanılacak bağımlı değişkenler için data-frame oluşturulur. Ardından eksik, kayıt veriler tespit edilir. Tahminde kullanılacak iki değişken arasındaki korelasyon incelenir ve lineer regresyon yöntemi ile tahmin yapılarak değerler doldurulur. Skor ve durum kolonları arasındaki ilişkiden yararlanarak regresyon yapılmış ve skor alanındaki eksik veriler doldurulmuştur. Aynı yöntem alan adı kolonu ile ziyaretçinin ip adresinin tutulduğu kolon arasındaki ilişkiye de bakılarak uygulanmış ve lineer regresyon ile iki kolon arasındaki ilişkiye göre verilerin doldurulması sağlanmıştır (Li vd., 2014).

3.7.4. Özellik Seçimi

Özellik seçimi, veri setinde bulunan en yararlı özniteliklerin belirlenmesi ve seçilmesi sürecidir. Öznitelikler, hedeflenen model çıktısını oluşturacak kolonlardır. Veri setinde bulunan kolonlar aslında öznitelik kümesidir.

Özellik seçimi aşaması, nitelik seçimi veya değişken seçimi olarak da isimlendirilir. Özellik seçimi yaparak veri seti üzerinde çalışılmakta olan problem için optimum özellikler seçilmiş olur ve bu sayede veri kümesindeki özellik sayısı da azaltılmış olur. Azaltma işlemi ilk bakışta olumsuz bir süreç gibi düşünülse de analiz işleminde birçok avantaj sağlar.

Bu avantajlardan bazıları; özellik kümesinin boyutunun düşürülmesi, algoritma hızının artırılması, verinin kaliteli olması, hafıza optimizasyonu ve model başarımıdır.

Özellik seçimi yapabilmek için literatürde pek çok farklı yöntem bulunur. Ancak anlaşılır ve yapısı basit olması sebebiyle "Temel Bileşenler Analizi – Principal Component

Analysis (PCA)" yaygın olarak kullanılan özellik seçimi yöntemlerinden bir tanesidir. PCA ile çok değişkenli bir veri seti içerisindeki veriyi daha az değişkenle ve minimum bilgi kaybıyla ortaya dökmenin matematiksel bir tekniğini sunar. Bu yöntemin temel amacı özellik azaltmaktır ve bu işlemi en az veri kaybıyla yapabilecek bir dönüşüm tekniği kullanır. Temel bileşenler analizinin üç özelliği bulunur.

1. Korelasyonsuzdur.

2. Birinci temel bileşen toplam değişkenliği açıklayan en net değişkendir.

3. Sonraki temel bileşen kalan değişkenliği en net açıklayan değişkendir.

Ham veri seti üzerindeki niteliklerin bir kısmı manuel olarak MS Excel ortamındayken kaldırılmıştı. Ancak diğer niteliklerin özellik seçimi yoluyla azaltılması için bir çalışma yapılmış ve bu çalışmada temel bileşenlerin seçilerek azaltılması sağlanmıştır.

3.7.5. İlişkisel Madencilik Uygulaması

Veri madenciliğinde kümeleme, sınıflandırma, birliktelik kuralları ve ilişki analizi veri üzerinden anlamlı veriler elde etme amaçlı yapılan işlemlerdir. Birliktelik kuralı, geçmiş verilerin analiziyle bu veriler arasındaki birliktelik davranışlarını belirlemeyi amaçlayan bir yöntemdir ve gelecekteki çalışmaları destekler. AIS, APRIORI, SETM, OCD ve KARMA gibi farklı ilişkisel madencilik yöntemleri vardır. Bu çalışmada, veri setinde Apriori ilişkisel madencilik algoritması kullanılmıştır. Apriori algoritması, 1994 yılında Agrawal ve Srikant tarafından geliştirilmiştir ve adını önceden bilinen bilgileri (priority) kullanmasından almaktadır (Al-Maolegi ve Arkok, 2014).

Birliktelik kuralında, veri seti üzerinde öğeler arasındaki birlikteliği hesaplamak için "Destek" ve "Güven" kriterleri kullanılır. Destek, veride öğeler arasındaki bağıntının ne kadar sık olduğunu gösterir. Örneğin, bir internet sitesini ziyaret eden ziyaretçilerin veya saldırganların sıklıkla hangi diğer internet sitelerini de ziyaret ettiğini inceleyecek olursak:

• Destek (A), A sitesinin tüm internet trafiği içindeki oranıdır. Matematiksel olarak, Destek (A) = A sayısı / toplam trafik sayısı şeklinde ifade edilir.

• Destek (A,B), A ve B sitelerinin bir arada tüm internet trafiği içinde yer alma olasılığıdır. Matematiksel olarak, Destek (A,B) = (A,B) sayısı / toplam internet trafik sayısı şeklinde ifade edilir.

Güven kriteri ise, B sitesinin hangi olasılıkla A sitesiyle birlikte ziyaret edileceğini gösterir. Matematiksel olarak,

• Güven (A,B) = (A,B) sayısı / A'yı içeren internet trafik sayısı

• Güven (A → B) = Destek (A,B) / Destek (A)

Her bir kural, bir destek ve güven değeri ile ifade edilir ve kuralların güvenilirliği destek ve güven değerleriyle doğru orantılıdır.

A→B [destek=3%, güven=60%] birliktelik kuralı, analiz edilen tüm internet ziyaret trafiğinden %3'ünde A ve B sitelerinin birlikte ziyaret edildiğini gösterir. Ayrıca, A sitesini ziyaret eden kullanıcıların %60'ının aynı trafikte B sitesini de ziyaret ettiğini belirtir. Bu birliktelik kuralı için aşağıdaki adımlar takip edilir:

1. Minimum destek sayısı ve minimum güven değerleri belirlenir.
2. Öğe kümelerinin her bir öğesi için destek değeri hesaplanır.
3. Minimum destek değerinden düşük olan öğeler devre dışı bırakılır.
4. Tekli birliktelikler dikkate alınarak ikili birliktelikler oluşturulur.
5. Minimum destek değerinden düşük olan öğe kümeleri çıkartılır.
6. Üçlü birliktelikler oluşturulur.
7. Minimum destek değerini geçmeyen üçlü birliktelikler çıkartılır.
8. Üçlü birlikteliklerden birliktelik kuralları çıkartılır.

Veri seti nitelikleri Tablo 1'deki gibidir.

Tablo 1. Veri seti nitelikleri

Nitelik	Açıklama	Veri Türü
DateTime	Tarih, Saat	Date
Method	GET, POST	String
UserAgent	İsteğin Geldiği UserAgent	String
SourceIP	Ziyaretçinin IP Adresi	Char
DestinationDomain	Ziyaret Edilen Alan Adı	String
Score	WAF tarafından atanan tehdit puanı	Int
Status	Erişimin Durumu	Bool

Şekil 19'da veri setinin ham hali yer almaktadır. Veri seti üzerinde gerçekleştirilen dönüşüm işlemleri sonrasında CSV formatından ARFF formatına dönüşüm sağlanmıştır.

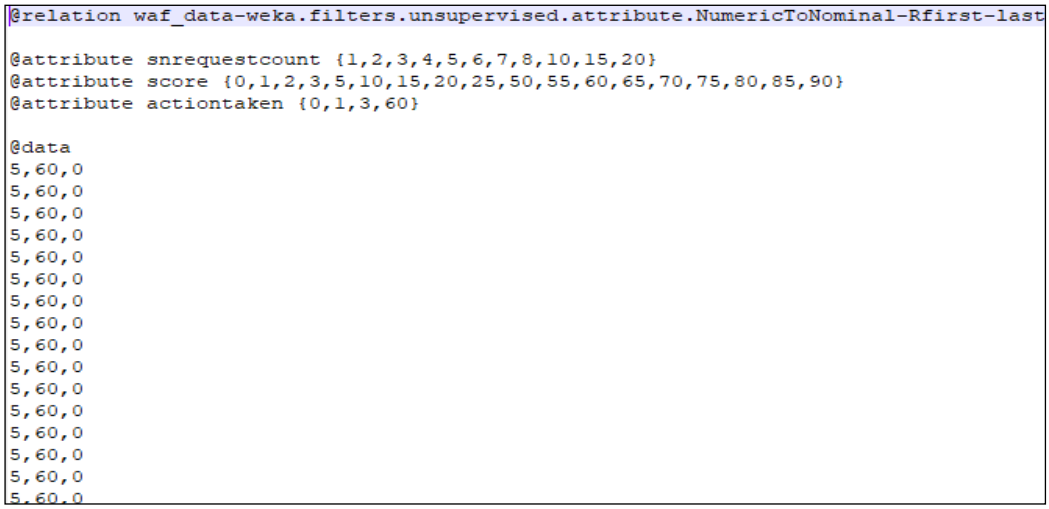
```
DateTime,Method,UserAgent,SourceIP,DestinationSite,Score,ActionTaken
2021-05-14 22:54:35,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.47,example.com,60,0
2021-05-14 23:04:26,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.47,example.com,60,0
2021-05-14 23:05:26,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.47,example.com,60,0
2021-05-14 23:05:26,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.47,example.com,60,0
2021-05-14 23:06:46,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.47,ab.example.com,60,0
2021-05-14 23:11:23,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.52,ab.example.com,60,0
2021-05-14 23:18:35,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.52,bs.example.com,60,0
2021-05-14 23:26:11,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.52,tn.example.com,60,0
2021-05-14 23:26:44,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.52,ab.example.com,60,0
2021-05-14 23:29:36,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.52,example.com,60,0
2021-05-14 23:29:37,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.52,example.com,60,0
2021-05-14 23:29:37,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.52,example.com,60,0
2021-05-14 23:29:37,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.52,example.com,60,0
2021-05-14 23:29:37,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.52,example.com,60,0
2021-05-14 23:30:08,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.47,example.com,60,0
2021-05-14 23:30:14,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.47,example.com,60,0
2021-05-14 23:30:14,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.52,example.com,60,0
2021-05-14 23:30:14,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.52,example.com,60,0
2021-05-14 23:11:23,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.52,ab.example.com,60,0
2021-05-14 23:11:23,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.52,ab.example.com,60,0
2021-05-14 23:11:22,GET,Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm),157.55.39.52,ab.example.com,60,0
```

Şekil 19. Veri ön işleme aşamaları

ARFF formatına dönüştürülen veri seti şekil 20' de görülmektedir. Apriori algoritmasının uygulanması için uygun formata dönüştürülen veri seti ile nitelikler arasındaki ilişkiler belirlenebilecektir.

Şekil 20' de C# programlama dili ile Weka kütüphaneleri kullanılarak bir apriori ilişkisel madencilik uygulaması geliştirilmiştir. Nitelikler arasındaki ilişki veya bağıntı veri seti

incelendiğinde açıkça görülebiliyor olsa da ilişkisel madencilik sonucunda hangi nitelikler arasında apriori 'ye göre ilişki olduğu görülmektedir.

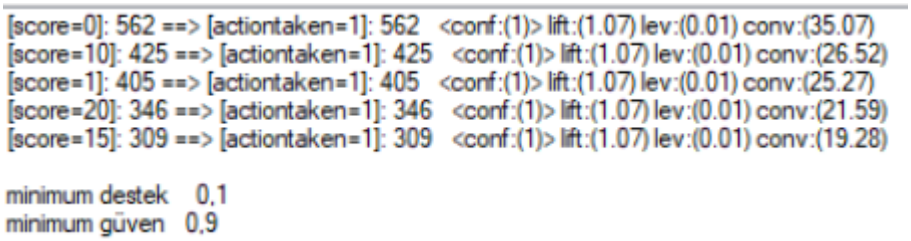


```
@relation waf_data-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last

@attribute snrequestcount {1,2,3,4,5,6,7,8,10,15,20}
@attribute score {0,1,2,3,5,10,15,20,25,50,55,60,65,70,75,80,85,90}
@attribute actiontaken {0,1,3,60}

@data
5,60,0
5,60,0
5,60,0
5,60,0
5,60,0
5,60,0
5,60,0
5,60,0
5,60,0
5,60,0
5,60,0
5,60,0
5,60,0
5,60,0
5,60,0
5,60,0
```

Şekil 20. Veri ön işleme aşamaları



```
[score=0]: 562 ==> [actiontaken=1]: 562   <conf:(1)> lift:(1.07) lev:(0.01) conv:(35.07)
[score=10]: 425 ==> [actiontaken=1]: 425   <conf:(1)> lift:(1.07) lev:(0.01) conv:(26.52)
[score=1]: 405 ==> [actiontaken=1]: 405   <conf:(1)> lift:(1.07) lev:(0.01) conv:(25.27)
[score=20]: 346 ==> [actiontaken=1]: 346   <conf:(1)> lift:(1.07) lev:(0.01) conv:(21.59)
[score=15]: 309 ==> [actiontaken=1]: 309   <conf:(1)> lift:(1.07) lev:(0.01) conv:(19.28)

minimum destek   0,1
minimum güven   0,9
```

Şekil 21. Veri ön işleme aşamaları

Apriori algoritması veri seti üzerine uygulandığında Score ve Status (ActionTaken) nitelikleri arasında bir ilişki kurulduğu gözlenmiştir. Bu iki nitelik arasında bir bağıntı olduğu görülmüştür. Tehdit skoru ne kadar yüksek veya düşükse web sitesine erişimin de oranda mümkün olup olmadığına dair bir ilişki kurulduğu görülmektedir.

Apriori algoritmasını çalıştırmadan önce yapılan ön işlemlerde veri seti CSV formatından ARFF formatına dönüştürülürken hangi niteliklerin işlemlere dahil edileceği belirtilmiş ve numerik değerlere dönüşüm işlemi yapılmıştır. Bu yüzden apriori çalıştığında numerik nitelikler arasında ilişki kurularak belirtilen minimum destek ile minimum güven değerlerine göre algoritma çalışmaktadır.

Karmaşık bir veri setinde veya niteliklerin sahip oldukları senaryolarda ne tür görevlerinin olduğunun tam olarak bilinemediği veri setleri üzerinde apriori ilişkisel madencilik uygulaması yapılması önemlidir. Bu çalışmadaki veri setinde skor ve dakikada gelen istek sayısına göre erişimin engellendiği veya engellenmediğinin yorumunu yapmak kolay olmuştur. Bu tahmin bir kez de apriori ilişkisinde doğrulanmış ve nihai veri setine geçiş için son aşama da tamamlanmıştır.

3.7.6. Veri Seti

Veri ön işleme adımlarında gerçekleştirilen tüm işlemlerde amaç veri üzerinde bilinmeyen veya fark edilmemiş özelliklerin ortaya çıkartılması ve veri seti üzerinde yapılacak sınıflandırma işlemi için optimum başarımı elde etmektir.

Bu sayede de çeşitli ön işleme aşamaları veri seti üzerinde uygulanmış, gereksiz görülen nitelikler çıkartılmış, sınıflandırma algoritmalarında sorun teşkil edeceği düşünülen niteliklere ait verilerde temizleme yapılmış, eksik olan veriler uygun yöntemlerle doldurulmuştur.

Bu aşamada sınıflandırma algoritmaları ile çalışabilir hale getirilmiş veri setinden örnek görüntü Tablo 2'de verilmiştir.

Tablo 2. Çalışmada kullanılacak veri seti

DateTime	Method	UserAgent	Source IP	Domain	Score	Status
2021-05-14 22:54:35	GET	Mozilla 5.0	157.55.39.47	example.com	60	0
2021-05-14 23:04:26	GET	Bingbot 2.0	157.55.39.47	example.com	60	0
2021-05-14 23:05:26	GET	Mozilla 5.0	157.55.39.47	example.com	60	0
2021-05-14 23:05:26	GET	Mozilla 5.0	157.55.39.47	example.com	60	0
2021-05-14 23:06.46	GET	Bingbot 2.0	157.55.39.47	example.com	60	0
2021-04-10 14:39:05	GET	Mozilla 5.0	157.55.39.5	tn.example.com	55	0
2021-04-10 14:40:37	GET	KHTML Like Gecko	157.55.39.5	tn.example.com	35	1
2021-04-10 14:40:51	GET	Python-requests	52.146.43.17	pos.example.com	35	1
2021-05-18 01:46:14	GET	Python-requests	52.146.43.17	example.com	10	1

3.8. Verinin İşlenmesi ve IP İtibarı

Cloud WAF ürününden temin edilen internet siteleri ziyaretçi günlükleri üzerinde yapılan veri ön işleme teknikleri ve elde edilen temiz veri seti önceki bölümde detayları belirtildiği gibi oluşturulmuştur. Veri seti belirli tarihlere göre farklı tarayıcılardan ve robotlardan gelen isteklerin geldiği web sitesi alan adı ve kaynak ip adresi ile WAF cihazı tarafından gelen isteğe verilen skora göre WAF tarafından erişimin verilip verilmediği sonucuna varılır. Gelen isteğin zararlı mı yoksa zararsız mı olup olmadığına WAF cihazı kendi yapısı içerisinde karar vermektedir. Bu karar mekanizmasında OWASP gibi siber güvenlik standart kuruluşları ile USOM (Ulusal Siber Olaylara Müdahale Merkezi) gibi kuruluşların kara liste ve beyaz liste kayıtlarındaki bilgiler, veri tabanlarına eklenen güncel zafiyetler ve zararlı istek türlerine dayalı çalışan bir algoritma vardır. Bu algoritma WAF üreticileri tarafından farklı kaynakları ve kendi yorumlama mekanizmasını kullanabilmektedir.

Bir internet sitesi trafiğindeki güvenlik skoru puanı eğer 50 değerinin üzerindeyse içerisindeki kural setlerinin çok sert olduğu anlaşılır ve güvensiz olarak yorumlanır ancak isteğin engellenip engellenmeyeceği kararını dakikada gelen istek sayısı belirler eğer hem puan yüksek hem de istek sayısı fazlaysa trafik engellenir. Bu engellenme durumunda veri setindeki durum (status) niteliği false (0) yani Reject dönmektedir. Eğer güvenlik tehdit puanı 50 değerinin altında bir değer geliyorsa ve dakikada gelen istek sayısı da düşükse veri setindeki durum (status) niteliği true (1) yani Access dönmektedir.

Bir isteğin zararlı veya zararsız olması durumu bu şekilde belirlenmektedir. İstek sayısı az olup, güvenlik tehdit puanı yüksek olabilir böyle bir durumda da WAF, isteği geçirecektir ancak kullanıcının karşısına Challange (meydan okuma) şeklinde bir seçenek çıkararak robot olup olmadığını anlayacaktır.

Böylece kullanıcı robot olmadığını doğrular ve erişim hakkı kazanır. Bu istek geçirildiği için status 1 olarak işaretlenir ancak arka tarafta WAF cihazı bu isteği Challenge yaparak geçirdiğinin bilgisini tutar.

Güvenlik tehdit puanının oluşmasında ayrıca WAF cihazının konfigürasyonu da önemli rol oynamaktadır. Çok hassas konfigüre edilmiş bir cihazda güvenlik kural setleri

sensörleri hassas davranacak ve gelen her isteği daha ayrıntılı inceleyecektir. Bu durumda aslında zararsız olan çoğu kullanıcı tarayıcısında bulunan bir botnet virüsü nedeniyle zararlı kullanıcı veya bot olarak algılanabilir. Bu durum false-positive olarak değerlendirildiğinden WAF üzerindeki hassasiyet ayarının orta seviyelerde olması tercih edilmektedir. Veri setinin oluşturulduğu WAF cihazının konfigürasyonu da orta (Medium) hassasiyet seviyesindedir.

Belirtilen bilgiler ışığında veri seti bölüm 3.7.6'da gösterilen nihai halini almıştır. Bu aşamadan sonra veri seti üzerinde çeşitli modellerle makine öğrenmesi yöntemlerinden sınıflandırma algoritmaları çalıştırılacaktır.

3.9. Sınıflandırmayla IP İtibar Analizinin Uygulanması

Bu bölümde sınıflandırma işlemine uygun hale getirilmiş veri seti üzerinde En Yakın Komşu Algoritması, Doğrusal Destek Vektör Makinesi Algoritması, RBF Destek Vektör Makinesi Algoritması, Rastgele Orman Ağacı Algoritması, Karar Ağaçları Algoritması, Yapay Sinir Ağı Algoritması, Adaboost Algoritması ve Naive Bayes Algoritmaları ile sınıflandırma yapılmıştır.

Yapılan sınıflandırma işleminde veriler eğitim ve test verisi olarak iki bölüme ayrılmıştır. Verilerin eğitim ve test verisi olarak ayrılması işlemi, 10 katlı çapraz doğrulama yöntemi uygulanarak gerçekleştirilmiştir. Uygulanan yöntemin sonucu olarak karmaşıklık matrisi ve %95 güven aralığında performans metrik değerleri alınmıştır. Bu sonuçlar Sonuç ve Öneriler bölümünde tablolar halinde listelenmektedir.

4. SONUÇ VE ÖNERİLER

Bu tez çalışmasında internet trafiği içerisindeki zararlı isteklerin tahmin edilmesi için sekiz farklı makine öğrenmesi algoritması üzerinde testler gerçekleştirilmiştir. Testlerde farklı algoritmaların kullanılmasının yanı sıra IP adreslerine dayalı itibar kriteri de ele alınarak çalışmalar iki şekilde yapılmıştır.

Öncelikle makine öğrenmesi algoritmalarında test ve eğitim işlemleri hedeflenen tahmin için IP itibarı (With IP Reputation) kriterine dayalı ardından IP itibarı göz ardı edilerek (Without IP Reputation) çalıştırılmıştır.

Yapılan deneylerde; En Yakın Komşu Algoritması, Doğrusal Destek Vektör Makinesi Algoritması, RBF Destek Vektör Makinesi Algoritması, Rastgele Orman Ağacı Algoritması, Karar Ağaçları Algoritması, Yapay Sinir Ağı Algoritması, Adaboost Algoritması ve Naive Bayes Algoritmaları kullanılmıştır.

Uygulama bölümünde anlatıldığı şekilde algoritmalar üzerinde test ve eğitim işlemleri gerçekleştirilmiş ve tahminler elde edilmiştir.

Algoritmaların başarımları sırasıyla En Yakın Komşu Algoritması, Doğrusal Destek Vektör Makinesi Algoritması, RBF Destek Vektör Makinesi Algoritması, Rastgele Orman Ağacı Algoritması, Karar Ağaçları Algoritması, Yapay Sinir Ağı Algoritması, Adaboost Algoritması ve Naive Bayes Algoritması şeklindedir. Algoritma başarımlarına ait detaylı çıktılar aşağıdaki gibidir.

4.1. IP İtibarı Olmadan Yapılan Sınıflandırma (Without IP Reputation)

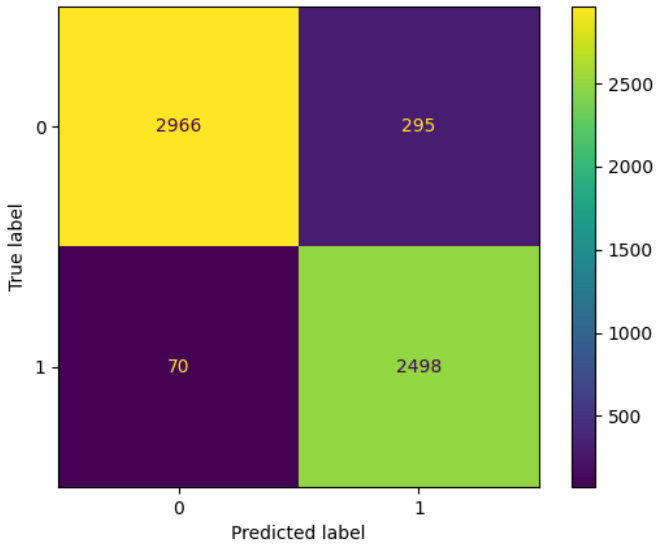
Gerçekleştirilen deneylerden IP İtibarı kriteri sınıflandırmaya dahil edilmeden yapılan çalışmalarda kullanılan her bir sınıflandırma algoritmasına göre sonuçlar aşağıdaki tablolarda yer almaktadır.

En yakın komşu algoritması IP itibarı olmadan yapılan sınıflandırmada Tablo 3'te yer alan metriklere göre sonuçlar üretmiştir.

Tablo 3. En yakın komşu algoritması

Metrik Adı	Sonuç
F1 Skoru	0.94±0.04
Precision	0.91±0.07
Recall	0.97±0.03
F1 Weighted	0.94±0.04
Top-K-Accuracy	1.00±0.00
Balanced Accuracy	0.94±0.04
Accuracy	0.94±0.04
F1 Micro	0.94±0.04
Jaccard	0.88±0.07
AVG Precision	0.95±0.04
ROC AUC	0.97±0.02
F1 Macro	0.94±0.04

En yakın komşu algoritması IP itibarı olmadan yapılan sınıflandırmada Şekil 22'de yer aldığı gibi karmaşıklık matrisi üretilmiştir.



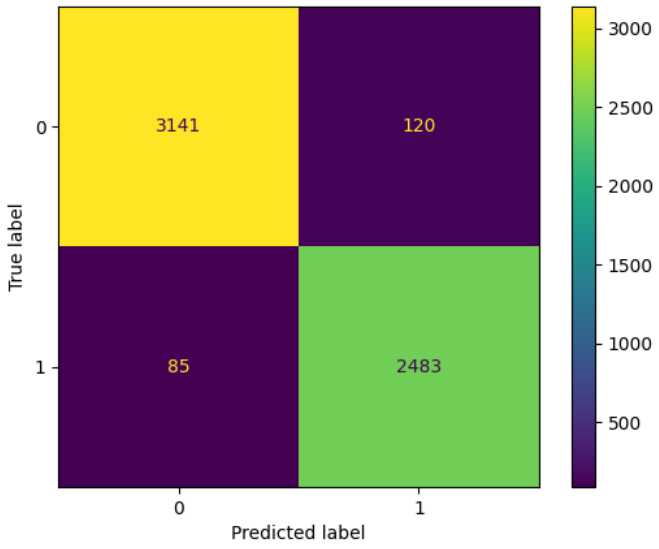
Şekil 22. En yakın komşu algoritması karmaşıklık matrisi

Doğrusal Destek Makinesi algoritması IP itibarı olmadan yapılan sınıflandırmada aşağıdaki tabloda yer alan metriklere göre sonuçlar üretmiştir.

Tablo 4. Doğrusal destek makinesi algoritması

Metrik Adı	Sonuç
F1 Skoru	0.96±0.04
Precision	0.96±0.06
Recall	0.97±0.02
F1 Weighted	0.96±0.04
Top-K-Accuracy	1.00±0.00
Balanced Accuracy	0.97±0.03
Accuracy	0.96±0.04
F1 Micro	0.96±0.04
Jaccard	0.93±0.07
AVG Precision	0.98±0.02
ROC AUC	0.99±0.01
F1 Macro	0.96±0.04

Doğrusal Destek Vektör Makinesi IP itibarı olmadan yapılan sınıflandırmada aşağıdaki şekilde yer aldığı gibi karmaşıklık matrisi üretilmiştir.



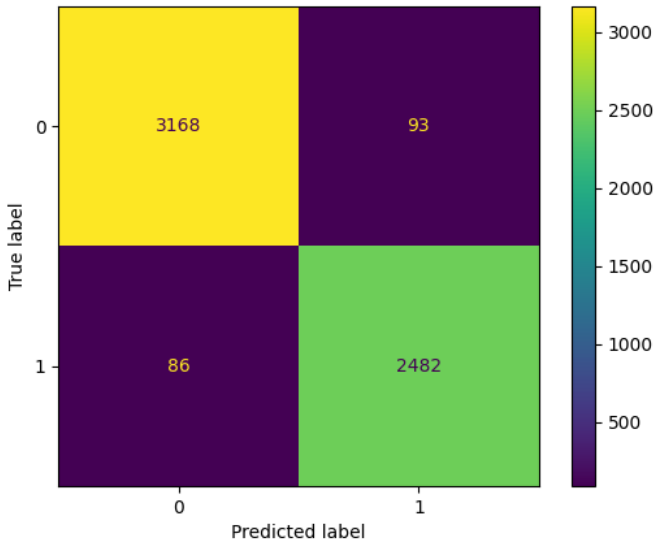
Şekil 23. Doğrusal vektör makinesi karmaşıklık matrisi

RBF Doğrusal Destek Makinesi algoritması IP itibarı olmadan yapılan sınıflandırmada Tablo 5'te yer alan metriklere göre sonuçlar üretmiştir.

Tablo 5. RBF doğrusal vektör makinesi algoritması

Metrik Adı	Sonuç
F1 Skoru	0.97±0.03
Precision	0.97±0.04
Recall	0.97±0.02
F1 Weighted	0.97±0.03
Top-K-Accuracy	1.00±0.00
Balanced Accuracy	0.97±0.03
Accuracy	0.97±0.03
F1 Micro	0.97±0.03
Jaccard	0.94±0.05
AVG Precision	0.98±0.02
ROC AUC	0.98±0.02
F1 Macro	0.97±0.03

RBF Doğrusal Destek Vektör Makinesi IP itibarı olmadan yapılan sınıflandırmada Şekil 24'te yer aldığı gibi karmaşıklık matrisi üretilmiştir.



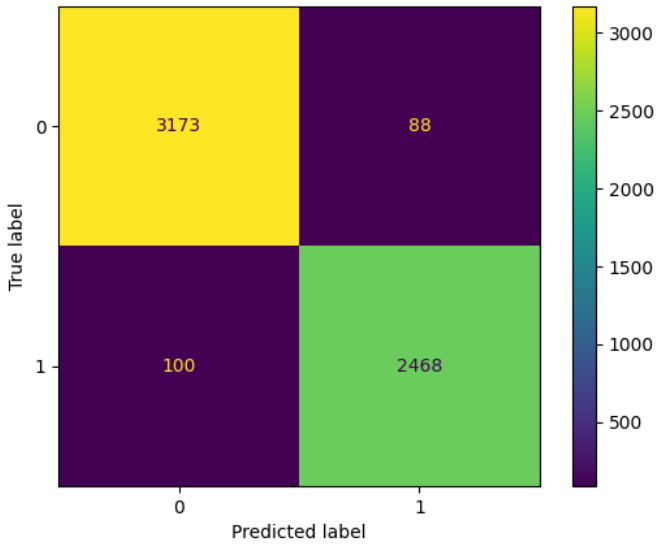
Şekil 24. RBF doğrusal destek vektör makinesi karmaşıklık matrisi

Karar Ağacı algoritması IP itibarı olmadan yapılan sınıflandırmada Tablo 6'da yer alan metriklere göre sonuçlar üretmiştir.

Tablo 6. Karar ağacı algoritması

Metrik Adı	Sonuç
F1 Skoru	0.96±0.03
Precision	0.97±0.04
Recall	0.96±0.04
F1 Weighted	0.97±0.03
Top-K-Accuracy	1.00±0.00
Balanced Accuracy	0.97±0.03
Accuracy	0.97±0.03
F1 Micro	0.97±0.03
Jaccard	0.93±0.06
AVG Precision	0.96±0.05
ROC AUC	0.97±0.03
F1 Macro	0.97±0.03

Karar ağacı algoritması IP itibarı olmadan yapılan sınıflandırmada Şekil 25'te yer aldığı gibi karmaşıklık matrisi üretilmiştir.



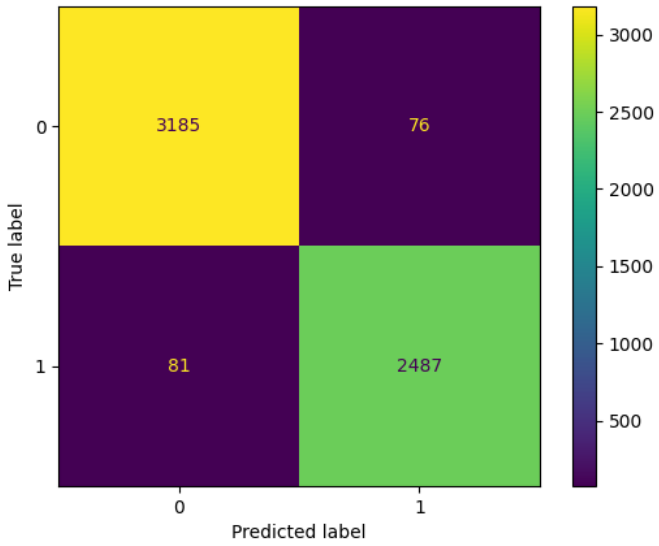
Şekil 25. Karar ağacı algoritması karmaşıklık matrisi

Rastgele Orman Ağacı algoritması IP itibarı olmadan yapılan sınıflandırmada Tablo 7'de yer alan metriklere göre sonuçlar üretmiştir.

Tablo 7. Rastgele orman ağacı algoritması

Metrik Adı	Sonuç
F1 Skoru	0.97±0.03
Precision	0.98±0.03
Recall	0.97±0.04
F1 Weighted	0.98±0.02
Top-K-Accuracy	1.00±0.00
Balanced Accuracy	0.98±0.02
Accuracy	0.98±0.02
F1 Micro	0.98±0.02
Jaccard	0.95±0.05
AVG Precision	0.99±0.02
ROC AUC	0.99±0.01
F1 Macro	0.98±0.02

Rastgele Orman Ağacı algoritması IP itibarı olmadan yapılan sınıflandırmada Şekil 26'da yer aldığı gibi karmaşıklık matrisi üretilmiştir.



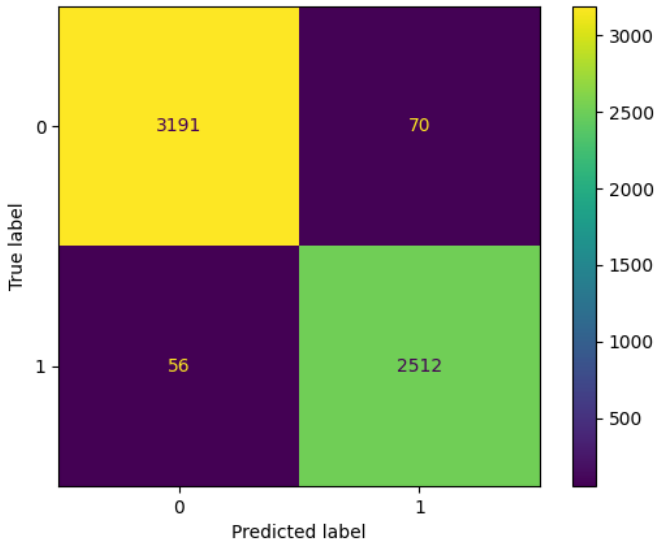
Şekil 26. Rastgele orman ağacı algoritması karmaşıklık matrisi

Yapay Sinir Ağı Algoritması IP itibarı olmadan yapılan sınıflandırmada Tablo 8'de yer alan metriklere göre sonuçlar üretmiştir.

Tablo 8. Yapay sinir ağı algoritması

Metrik Adı	Sonuç
F1 Skoru	0.98±0.02
Precision	0.98±0.04
Recall	0.98±0.01
F1 Weighted	0.98±0.02
Top-K-Accuracy	1.00±0.00
Balanced Accuracy	0.98±0.02
Accuracy	0.98±0.02
F1 Micro	0.98±0.02
Jaccard	0.96±0.04
AVG Precision	0.99±0.03
ROC AUC	0.99±0.01
F1 Macro	0.98±0.02

Yapay Sinir Ağı Algoritması IP itibarı olmadan yapılan sınıflandırmada Şekil 27'de yer aldığı gibi karmaşıklık matrisi üretilmiştir.



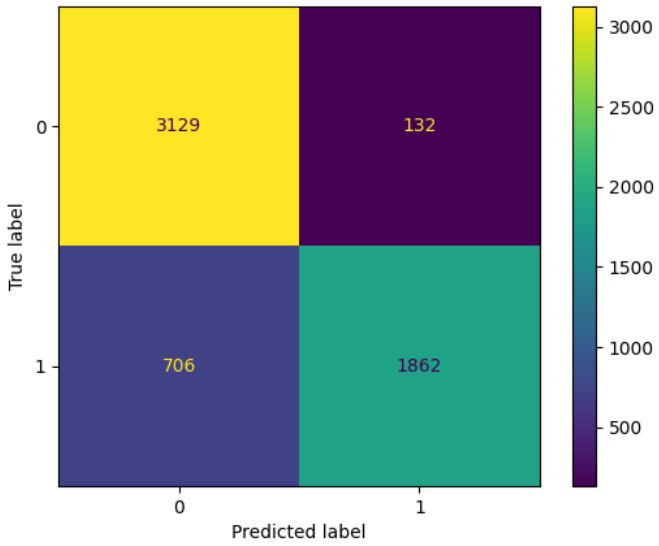
Şekil 27. Yapay sinir ağı algoritması karmaşıklık matrisi

Adaboost Algoritması IP itibarı olmadan yapılan sınıflandırmada Tablo 9'da yer alan metriklere göre sonuçlar üretmiştir.

Tablo 9. Adaboost algoritması

Metrik Adı	Sonuç
F1 Skoru	0.76±0.25
Precision	0.87±0.16
Recall	0.73±0.26
F1 Weighted	0.83±0.15
Top-K-Accuracy	1.00±0.00
Balanced Accuracy	0.84±0.13
Accuracy	0.86±0.12
F1 Micro	0.86±0.12
Jaccard	0.69±0.25
AVG Precision	0.97±0.04
ROC AUC	0.98±0.02
F1 Macro	0.82±0.16

Adaboost Algoritması IP itibarı olmadan yapılan sınıflandırmada Şekil 28'de yer aldığı gibi karmaşıklık matrisi üretilmiştir.



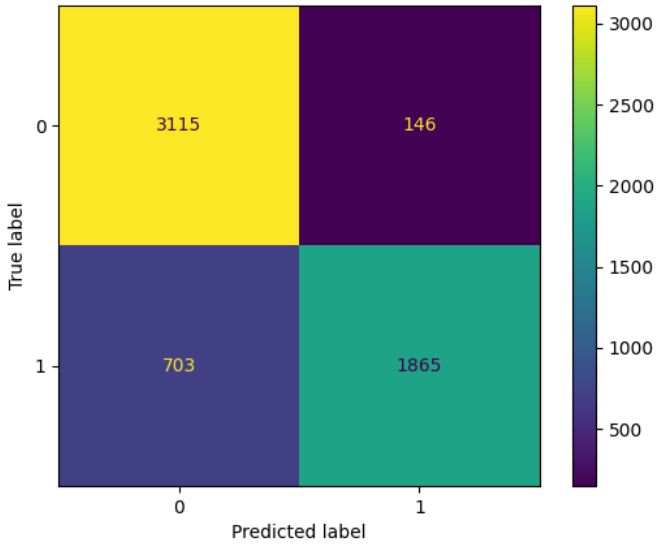
Şekil 28. Adaboost algoritması karmaşıklık matrisi

Naive Bayes Algoritması IP itibarı olmadan yapılan sınıflandırmada Tablo 10'da yer alan metriklere göre sonuçlar üretmiştir.

Tablo 10. Naive bayes algoritması

Metrik Adı	Sonuç
F1 Skoru	0.75±0.25
Precision	0.89±0.16
Recall	0.73±0.26
F1 Weighted	0.83±0.15
Top-K-Accuracy	1.00±0.00
Balanced Accuracy	0.84±0.13
Accuracy	0.85±0.12
F1 Micro	0.85±0.12
Jaccard	0.69±0.25
AVG Precision	0.97±0.04
ROC AUC	0.98±0.02
F1 Macro	0.82±0.16

Naive Bayes IP itibarı olmadan yapılan sınıflandırmada Şekil 29'da yer aldığı gibi karmaşıklık matrisi üretilmiştir.



Şekil 29. Naive bayes algoritması karmaşıklık matrisi

4.2. IP İtibarı Kriteriyle Yapılan Sınıflandırma (With IP Reputation)

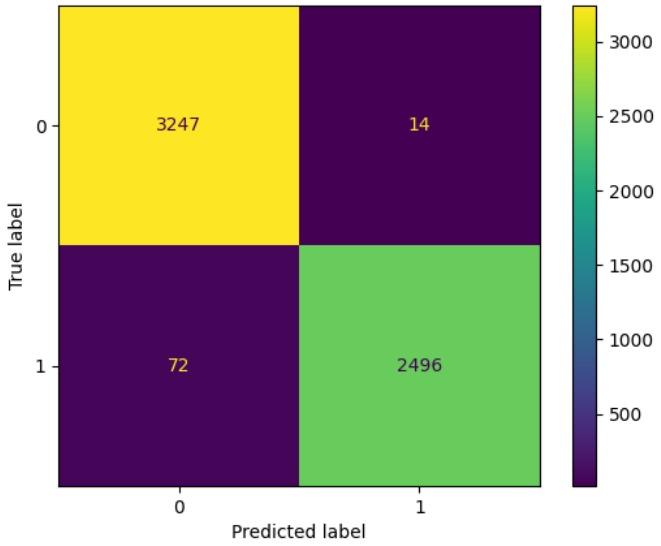
Gerçekleştirilen deneylerden IP İtibarı kriteri sınıflandırmaya dahil edilerek yapılan çalışmalarda kullanılan her bir sınıflandırma algoritmasına göre sonuçlar Tablo 11,12,13,14,15,16,17,18'de yer almaktadır.

En yakın komşu algoritması için IP itibarı kriteri dahil edilerek yapılan sınıflandırmada sonuçlar Tablo 11'de yer alan metriklere göre üretmiştir.

Tablo 11. En yakın komşu algoritması

Metrik Adı	Sonuç
F1 Skoru	0.98±0.03
Precision	0.99±0.01
Recall	0.97±0.05
F1 Weighted	0.99±0.03
Top-K-Accuracy	1.00±0.00
Balanced Accuracy	0.98±0.03
Accuracy	0.99±0.03
F1 Micro	0.99±0.03
Jaccard	0.97±0.05
AVG Precision	0.99±0.02
ROC AUC	0.99±0.01
F1 Macro	0.98±0.03

En yakın komşu algoritması IP itibarı kriteri dahil edilerek yapılan sınıflandırmada Şekil 30'da yer aldığı gibi karmaşıklık matrisi üretilmiştir.



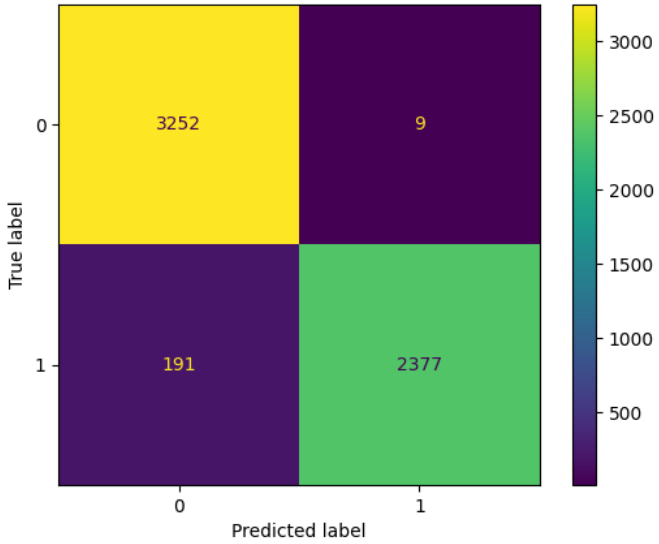
Şekil 30. En yakın komşu algoritması karmaşıklık matrisi

Doğrusal Destek Vektör Algoritması için IP itibarı kriteri dahil edilerek yapılan sınıflandırmada sonuçlar Tablo 12'de yer alan metriklere göre üretmiştir.

Tablo 12. Doğrusal destek vektör algoritması

Metrik Adı	Sonuç
F1 Skoru	0.95±0.07
Precision	1.00±0.01
Recall	0.93±0.11
F1 Weighted	0.96±0.05
Top-K-Accuracy	1.00±0.00
Balanced Accuracy	0.96±0.05
Accuracy	0.97±0.05
F1 Micro	0.97±0.05
Jaccard	0.92±0.11
AVG Precision	0.99±0.01
ROC AUC	0.99±0.01
F1 Macro	0.96±0.05

Doğrusal Destek Vektör Algoritması IP itibarı kriteri dahil edilerek yapılan sınıflandırmada Şekil 31'de yer aldığı gibi karmaşıklık matrisi üretilmiştir.



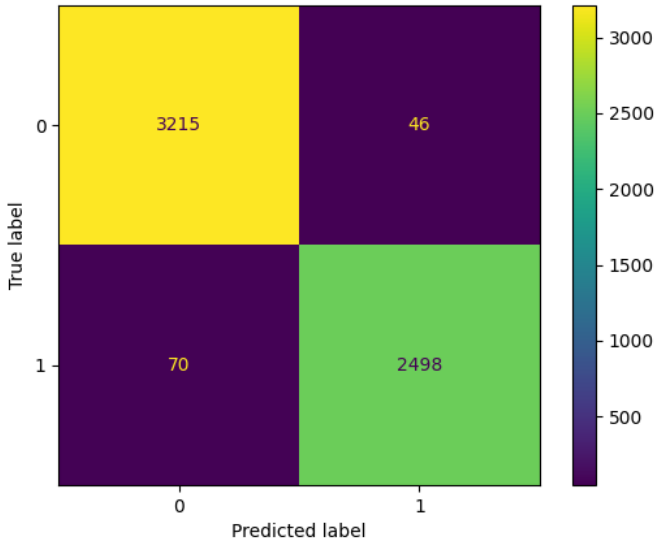
Şekil 31. Doğrusal vektör makinesi algoritması karmaşıklık matrisi

RBF Doğrusal Destek Vektör Algoritması için IP itibarı kriteri dahil edilerek yapılan sınıflandırmada sonuçlar Tablo 13'te yer alan metriklere göre üretmiştir.

Tablo 13. RBF doğrusal destek vektör algoritması

Metrik Adı	Sonuç
F1 Skoru	0.98±0.04
Precision	0.98±0.02
Recall	0.97±0.05
F1 Weighted	0.98±0.03
Top-K-Accuracy	1.00±0.00
Balanced Accuracy	0.98±0.03
Accuracy	0.98±0.03
F1 Micro	0.98±0.03
Jaccard	0.96±0.06
AVG Precision	0.99±0.01
ROC AUC	1.00±0.01
F1 Macro	0.98±0.03

RBF Doğrusal Destek Vektör Algoritması IP itibarı kriteri dahil edilerek yapılan sınıflandırmada Şekil 32'de yer aldığı gibi karmaşıklık matrisi üretilmiştir.



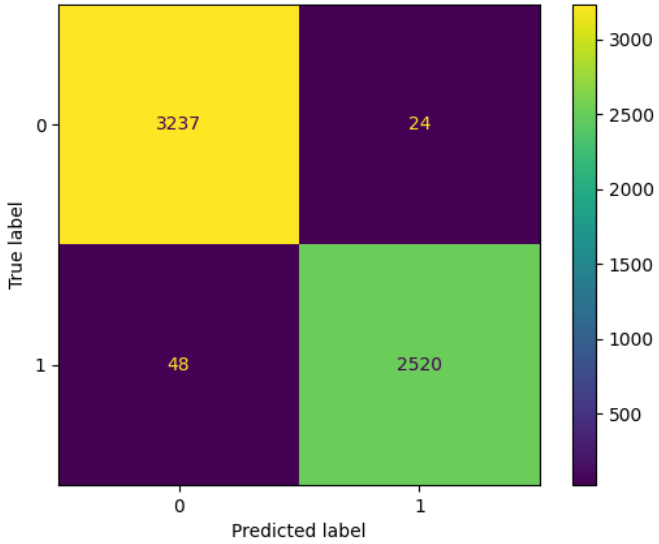
Şekil 32. RBF destek vektör makinesi algoritması karmaşıklık matrisi

Karar Ağacı Algoritması için IP itibarı kriteri dahil edilerek yapılan sınıflandırmada sonuçlar Tablo 14'te yer alan metriklere göre üretmiştir.

Tablo 14. Karar ağacı algoritması

Metrik Adı	Sonuç
F1 Skoru	0.99±0.02
Precision	0.99±0.02
Recall	0.98±0.03
F1 Weighted	0.99±0.02
Top-K-Accuracy	1.00±0.00
Balanced Accuracy	0.99±0.02
Accuracy	0.99±0.02
F1 Micro	0.99±0.02
Jaccard	0.97±0.04
AVG Precision	0.99±0.02
ROC AUC	0.99±0.01
F1 Macro	0.99±0.02

Karar Ağacı Algoritması IP itibarı kriteri dahil edilerek yapılan sınıflandırmada Şekil 33'te yer aldığı gibi karmaşıklık matrisi üretilmiştir.



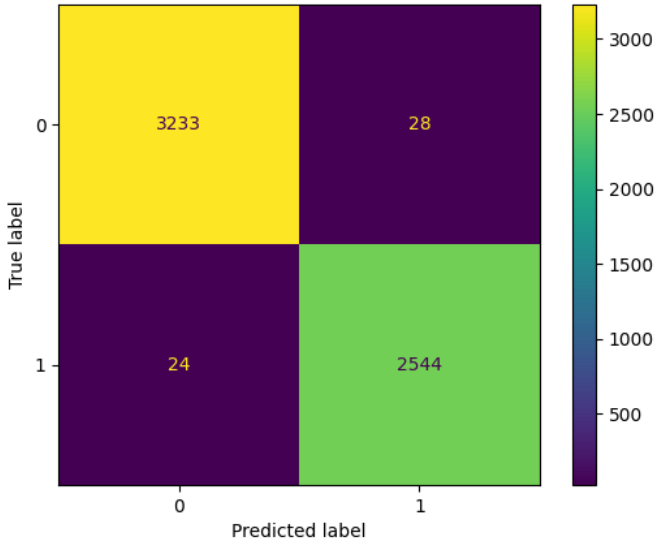
Şekil 33. Karar ağacı algoritması karmaşıklık matrisi

Rastgele Orman Ağacı Algoritması için IP itibarı kriteri dahil edilerek yapılan sınıflandırmada sonuçlar Tablo 15'te yer alan metriklere göre üretmiştir.

Tablo 15. Rastgele orman ağacı algoritması

Metrik Adı	Sonuç
F1 Skoru	0.99±0.01
Precision	0.99±0.02
Recall	0.99±0.01
F1 Weighted	0.99±0.01
Top-K-Accuracy	1.00±0.00
Balanced Accuracy	0.99±0.01
Accuracy	0.99±0.01
F1 Micro	0.99±0.01
Jaccard	0.98±0.03
AVG Precision	1.00±0.01
ROC AUC	1.00±0.01
F1 Macro	0.99±0.01

Rastgele Orman Ağacı Algoritması IP itibarı kriteri dahil edilerek yapılan sınıflandırmada Şekil 34'te yer aldığı gibi karmaşıklık matrisi üretilmiştir.



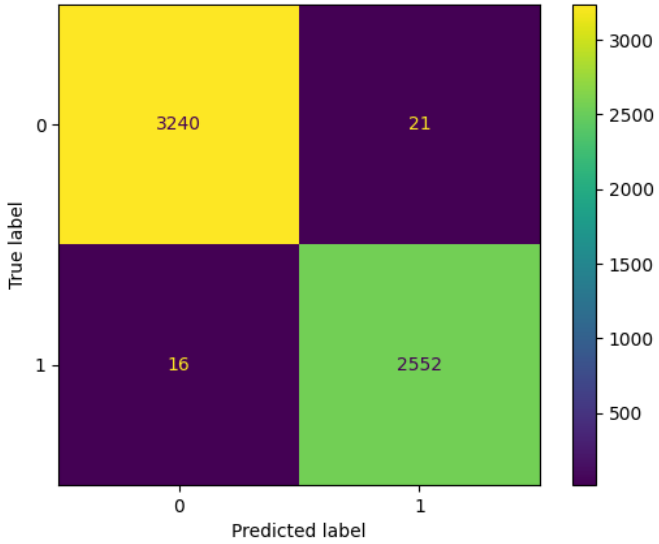
Şekil 34. Rastgele orman ağacı algoritması karmaşıklık matrisi

Yapay Sinir Ağı algoritması için IP itibarı kriteri dahil edilerek yapılan sınıflandırmada sonuçlar Tablo 16'da yer alan metriklere göre üretmiştir.

Tablo 16. Yapay sinir ağı algoritması

Metrik Adı	Sonuç
F1 Skoru	0.99±0.01
Precision	0.99±0.02
Recall	0.99±0.01
F1 Weighted	0.99±0.01
Top-K-Accuracy	1.00±0.00
Balanced Accuracy	0.99±0.01
Accuracy	0.99±0.01
F1 Micro	0.99±0.01
Jaccard	0.98±0.03
AVG Precision	1.00±0.01
ROC AUC	1.00±0.01
F1 Macro	0.99±0.01

Yapay Sinir Ağı Algoritması IP itibarı kriteri dahil edilerek yapılan sınıflandırmada Şekil 35'te yer aldığı gibi karmaşıklık matrisi üretilmiştir.



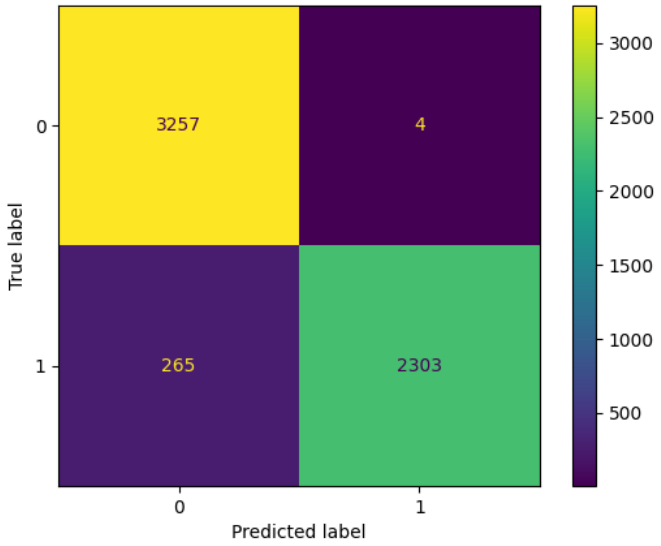
Şekil 35. Yapay sinir ağı algoritması karmaşıklık matrisi

Adaboost Algoritması için IP itibarı kriteri dahil edilerek yapılan sınıflandırmada sonuçlar Tablo 17'de yer alan metriklere göre üretmiştir.

Tablo 17. Adaboost algoritması

Metrik Adı	Sonuç
F1 Skoru	0.94±0.08
Precision	1.00±0.00
Recall	0.94±0.08
F1 Weighted	1.00±0.00
Top-K-Accuracy	0.90±0.13
Balanced Accuracy	0.95±0.06
Accuracy	1.00±0.00
F1 Micro	0.95±0.06
Jaccard	0.95±0.06
AVG Precision	0.95±0.06
ROC AUC	0.90±0.13
F1 Macro	0.99±0.02

Adaboost Algoritması IP itibarı kriteri dahil edilerek yapılan sınıflandırmada Şekil 36'da yer aldığı gibi karmaşıklık matrisi üretilmiştir.



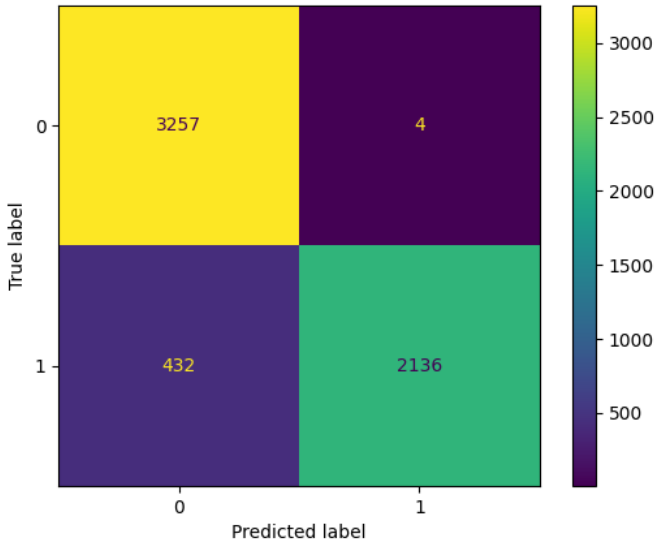
Şekil 36. Adaboost algoritması karmaşıklık matrisi

Naive-Bayes Algoritması için IP itibarı kriteri dahil edilerek yapılan sınıflandırmada sonuçlar Tablo 18'de yer alan metriklere göre üretmiştir.

Tablo 18. Naive bayes algoritması

Metrik Adı	Sonuç
F1 Skoru	0.86±0.23
Precision	0.93±0.15
Recall	0.86±0.23
F1 Weighted	0.93±0.15
Top-K-Accuracy	0.83±0.24
Balanced Accuracy	0.91±0.14
Accuracy	1.00±0.00
F1 Micro	0.92±0.12
Jaccard	0.93±0.10
AVG Precision	0.93±0.10
ROC AUC	0.83±0.24
F1 Macro	0.99±0.01

Naive-Bayes Algoritması IP itibarı kriteri dahil edilerek yapılan sınıflandırmada Şekil 37'de yer aldığı gibi karmaşıklık matrisi üretilmiştir.



Şekil 37. Naive bayes algoritması karmaşıklık matrisi

4.3. Sınıflandırma Sonuçlarının Karşılaştırılması

Tablo 19-20-21'de sınıflandırmada kullanılan tüm algoritmaların başarımları özet şekilde verilmiştir.

Tablo 19. F1 sonuçlarının karşılaştırılması

Algoritma Adı	IP İtibarıyla F1 Skoru	IP İtibarsız F1 Skoru
En Yakın Komşu	0.98±0.03	0.94±0.04
Doğrusal DVM	0.95±0.07	0.96±0.04
RBF DVM	0.98±0.04	0.97±0.03
Karar Ağaçları	0.99±0.02	0.96±0.03
Rastgele Orman Ağacı	0.99±0.01	0.97±0.03
Yapay Sinir Ağı	0.99±0.01	0.98±0.02
Adaboost	0.94±0.08	0.76±0.25
Naive Bayes	0.86±0.23	0.75±0.25

Tablo 20. Precision sonuçların karşılaştırılması

Algoritma Adı	IP İtibarıyla Precision Skoru	IP İtibarsız Precision Skoru
En Yakın Komşu	0.99±0.01	0.91±0.07
Doğrusal DVM	1.00±0.01	0.91±0.07
RBF DVM	0.98±0.02	0.97±0.04
Karar Ağaçları	0.99±0.02	0.97±0.04
Rastgele Orman Ağacı	0.99±0.02	0.98±0.03
Yapay Sinir Ağı	0.99±0.02	0.98±0.04
Adaboost	1.00±0.00	0.87±0.16
Naive Bayes	0.93±0.15	0.89±0.16

Tablo 21. Accuracy sonuçların karşılaştırılması

Algoritma Adı	IP İtibarıyla Accuracy Skoru	IP İtibarsız Accuracy Skoru
En Yakın Komşu	0.99±0.03	0.94±0.04
Doğrusal DVM	0.97±0.05	0.96±0.04
RBF DVM	0.98±0.03	0.97±0.03
Karar Ağaçları	0.99±0.02	0.97±0.03
Rastgele Orman Ağacı	0.99±0.01	0.98±0.02
Yapay Sinir Ağı	0.99±0.01	0.98±0.02
Adaboost	0.95±0.06	0.86±0.12
Naive Bayes	0.93±0.10	0.85±0.12

Bu çalışmada farklı makine öğrenmesi yöntemleri ile sınıflandırma işlemi ağ trafiğinin tahmini için gerçekleştirilmiştir. Deneysel sonuçlar algoritmaların öğrenmede dikkate aldıkları kriterlere (ip itibarına) göre farklı başarımlar sergilediğini göstermiştir. F1 skor sonuçlarına göre IP itibarıyla yapılan deneylerde, IP itibarsız yapılan deneylere göre Adaboost algoritmasında %18, Naive Bayes algoritmasında %9, En Yakın Komşu algoritmasında %4 başarı artımı gerçekleştiği görülen başlıca algoritmalardır. Precision skoru sonuçlarına göre IP itibarıyla yapılan deneylerde, IP itibarsız yapılan deneylere göre Adaboost algoritmasında %13, Doğrusal DVM algoritmasında %9, En Yakın Komşu algoritmasında %8 başarı artımı gerçekleştiği görülen başlıca algoritmalardır. Accuary skoru sonuçlarına göre IP itibarıyla yapılan deneylerde, IP itibarsız yapılan

deneylere göre Adaboost algoritmasında %9, Naive Bayes algoritmasında %8, En Yakın Komşu algoritmasında %5 başarı artımı gerçekleştiği görülen başlıca algoritmalardır.

Gelecekte derin öğrenme yöntemlerine başvurularak ağ trafiği tahmin edilmesinde güvenlik ürünlerinde iz kaydı meydana gelmeden önceki kötücül trafiğin bıraktığı izler tahmin edilerek denetimsiz bir öğrenme yapısı ile ağ trafiği tahmini çalışmaları yapılarak gelecek yönünü oluşturabilir.

KAYNAKLAR

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

Alan, A., Karabatak, M., Mühendisliği Anabilim Dalı, Y., Bilimleri Enstitüsü, F., Üniversitesi, F., Mühendisliği Bölümü, Y., & Fakültesi, T. (2020). Veri Seti - Sınıflandırma İlişkisinde Performansa Etki Eden Faktörlerin Değerlendirilmesi. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, *32* (2), 531-540. https://doi.org/10.35234/FUMBD.738007

Alkan, M. A. (2019). *Makine Öğrenimi Nedir?* https://www.endustri40.com/makine-ogrenimi-nedir

Al-Maolegi, M., & Arkok, B. (2014). An Improved Apriori Algorithm For Association Rules. *International Journal on Natural Language Computing*, *3* (1), 21-29. https://doi.org/10.5121/ijnlc.2014.3103

Alqudah, N., & Yaseen, Q. (2020). Machine Learning for Traffic Analysis: A Review. *Procedia Computer Science*, *170*, 911-916. https://doi.org/10.1016/j.procs.2020.03.111

Ayık, Y. Z., ÖZDEMİR, A., & YAVUZ, U. (2010). LİSE TÜRÜ VE LİSE MEZUNİYET BAŞARISININ, KAZANILAN FAKÜLTE İLE İLİŞKİSİNİN VERİ MADENCİLİĞİ TEKNİĞİ İLE ANALİZİ. *Atatürk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, *10* (2), 441-454. https://dergipark.org.tr/tr/pub/ataunisosbil/issue/2820/38029

Beechey, M., Kyriakopoulos, K. G., & Lambotharan, S. (2021). Evidential classification and feature selection for cyber-threat hunting. *Knowledge-Based Systems*, *226*, 107120. https://doi.org/https://doi.org/10.1016/j.knosys.2021.107120

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). *A Training Algorithm for Optimal Margin Classifiers*.

Bozkır, A., Sezer, E., & Gök, B. (2009, Ocak). *Öğrenci Seçme Sınavında (ÖSS) Öğrenci Başarımını Etkileyen Faktörlerin Veri Madenciliği Yöntemleriyle Tespiti*.

Çetin, F., Yarimtepe, O., Tuglular, T., Yüksek, İ., Enstitüsü, T., Bölümü, B., İzmir, Kelimeler, A., Duvarları, G., Politikaları, G., & Anomali, P. (2023). *Güvenlik Duvarları için Politika Anomali Belirleme Algoritmasının Deneysel Uygulaması*.

Dandıl, E., & İlhan, K. (2019). Yapay Bağışıklık Algoritmaları ile Web Trafik Verilerinde Anomali Tespiti. *European Journal of Science and Technology*, 46-56. https://doi.org/10.31590/ejosat.636309

Datacadamia. (2017). *Statistics - Quadratic discriminant analysis (QDA)*. https://datacadamia.com/data_mining/discriminant_analysis_quadratic

Dong, S. (2021). Multi class SVM algorithm with active learning for network traffic classification. *Expert Systems with Applications*, *176*, 114885. https://doi.org/https://doi.org/10.1016/j.eswa.2021.114885

Emhan, Ö., & Akın, M. (2019). Filtreleme Tabanlı Öznitelik Seçme Yöntemlerinin Anomali Tabanlı Ağ Saldırısı Tespit Sistemlerine Etkisi. *DÜMF Mühendislik Dergisi*, *10* (2), 549-559. https://doi.org/10.24012/dumf.565842

Fan, Z., & Liu, R. (2017). Investigation of machine learning based network traffic classification. *2017 International Symposium on Wireless Communication Systems (ISWCS)*, 1-6. https://doi.org/10.1109/ISWCS.2017.8108090

Hatipoglu, E. (2018). *Machine Learning — Classification — Naive Bayes — Part 11 | by Ekrem Hatipoglu | Medium*. https://medium.com/@ekrem.hatipoglu/machine-learning-classification-naive-bayes-part-11-4a10cd3452b4

Jemal, I., Cheikhrouhou, O., Hamam, H., & Mahfoudhi, A. (2020). SQL Injection Attack Detection and Prevention Techniques Using Machine Learning. Içinde *International Journal of Applied Engineering Research* (C. 15, Issue 6). http://www.ripublication.com

Jeon, D., & Tak, B. (2022). BlackEye: automatic IP blacklisting using machine learning from security logs. *Wireless Networks*, *28* (2), 937-948. https://doi.org/10.1007/s11276-019-02201-5

Koyun, S. (2020). *Karar Ağacı Nedir? - Yönetimde Örnekler ve Avantajlar - Sezgin KOYUN*. https://www.sezginkoyun.com/karar-agaci-nedir/

Labayen, V., Magaña, E., Morató, D., & Izal, M. (2020). Online classification of user activities using machine learning on network traffic. *Computer Networks*, *181*. https://doi.org/10.1016/j.comnet.2020.107557

Landauer, M., Skopik, F., Wurzenberger, M., & Rauber, A. (2020). System log clustering approaches for cyber security applications: A survey. *Computers & Security*, *92*, 101739. https://doi.org/https://doi.org/10.1016/j.cose.2020.101739

Li, L., Su, X., Zhang, Y., Hu, J., & Li, Z. (2014). Traffic prediction, data compression, abnormal data detection and missing data imputation: An integrated study based on the decomposition of traffic time series. *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, 282-289. https://doi.org/10.1109/ITSC.2014.6957705

Melgani, F., & Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, *42* (8), 1778-1790. https://doi.org/10.1109/TGRS.2004.831865

Örnek, Ö., VATAN, S., SARIOĞLU, S., & YAZICI, A. (2018). Trafik Ağlarında Anomali Tespiti. *Eskişehir Osmangazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*. https://doi.org/10.31796/ogummf.440285

Şeker, Ş. E. (2011). *Normal Dağılım (Normal Distribution, Gauss Distribution)*. https://bilgisayarkavramlari.com/2011/06/08/normal-dagilim-normal-distribution-gauss-distribution/

Şenol, Ö. (2021). *Random Forests*. https://medium.com/yaz%C4%B1l%C4%B1m-ve-bili%C5%9Fim-kul%C3%BCb%C3%BC/random-forests-92fd17d9aa4f

Shafiq, M., Yu, X., Bashir, A. K., Chaudhry, H. N., & Wang, D. (2018). A machine learning approach for feature selection traffic classification using security analysis. *Journal of Supercomputing*, *74* (10), 4867-4892. https://doi.org/10.1007/s11227-018-2263-3

Shanmugam, B., & Idris, N. B. (2009). Improved intrusion detection system using fuzzy logic for detecting anamoly and misuse type of attacks. *SoCPaR 2009 - Soft Computing and Pattern Recognition*, 212-217. https://doi.org/10.1109/SoCPaR.2009.51

Shawn. (2022). *Linear Regression*. https://www.freecodecamp.org/news/search/?query=Linear%20Regression

Shirwaikar, R., & Bhandari, C. (2013). *K-means Clustering Method for the Analysis of Log Data*. http://www.cise.u

Sun, G., Liang, L., Chen, T., Xiao, F., & Lang, F. (2018). Network traffic classification based on transfer learning. *Computers and Electrical Engineering*, *69*, 920-927. https://doi.org/10.1016/j.compeleceng.2018.03.005

TAKAOĞLU, M., & ÖZER, Ç. (2019). SALDIRI TESPİT SİSTEMLERİNE MAKİNE ÖĞRENME ETKİSİ. *Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi*, 11-22. https://doi.org/10.33461/uybisbbd.558192

Tanrıkulu, H., & Sazlı, M. H. (2009). *Saldırı Tespit Sistemlerinde Yapay Sinir Ağlarının Kullanılması*. Yüksek Lisans Tezi, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

Uslu, M. (2016). *Yapay Sinir Ağları (YSA) Nedir ? – Kod5.org*. https://kod5.org/yapay-sinir-aglari-ysa-nedir/

Vaarandi, R. (2003). A data clustering algorithm for mining patterns from event logs. *Proceedings of the 3rd IEEE Workshop on IP Operations and Management, IPOM 2003*, 119-126. https://doi.org/10.1109/IPOM.2003.1251233

Wang, Z., Zhang, J., & Verma, N. (2015). Realizing Low-Energy Classification Systems by Implementing Matrix Multiplication Directly Within an ADC. *IEEE Transactions on Biomedical Circuits and Systems*, *9* (6), 825-837. https://doi.org/10.1109/TBCAS.2015.2500101

Yamansavascilar, B., Guvensan, M. A., Yavuz, A. G., & Karsligil, M. E. (2017). Application identification via network traffic classification. *2017 International Conference on Computing, Networking and Communications, ICNC 2017*, 843-848. https://doi.org/10.1109/ICCNC.2017.7876241

Yıldırım, M. Z., Çavuşoğlu, A., Şen, B., & Budak, İ. (2014). *Yapay Sinir Ağları ile Ağ Üzerinde Saldırı Tespiti ve Paralel Optimizasyonu*.

Zhang, J., Xiang, Y., Wang, Y., Zhou, W., Xiang, Y., & Guan, Y. (2013). Network traffic classification using correlation information. *IEEE Transactions on Parallel and Distributed Systems*, *24* (1), 104-117. https://doi.org/10.1109/TPDS.2012.98

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	: Furkan DANIŞ

EĞİTİM DURUMU

Lisans Öğrenimi	: *2018, KTO Karatay Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği*
Yüksek Lisans Öğrenimi	: *2023, KTO Karatay Üniversitesi, Lisansüstü Eğitim Enstitüsü, Elektrik – Bilgisayar Mühendisliği*
Bildiği Yabancı Diller	: *İngilizce*

İŞ DENEYİMİ

Stajlar	: *2016, Stajyer Mühendis, Dalisto Adsoft*
	2017, Stajyer Mühendis, Ontos GmbH
Çalıştığı Kurumlar	: *2012, Kurucu, DNS Tech Company LLC*
	2018, Siber Güvenlik Takım Lideri, Altın Lale Grup
	2019-Halen, Siber Güvenlik Mühendisi, Kuveyt Türk

Tarih: 18 Ocak 2023