TÜRKİYE CUMHURİYETİ
ANKARA ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ

ADLİ BİLİŞİMDE MAKİNE ÖĞRENMESİ: MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE TERÖR OLAYLARININ TAHMİN EDİLMESİ ÇALIŞMASI

Burak GÖRMEZ

DİSİPLİNLERARASI ADLİ BİLİMLER ANABİLİM DALI
YÜKSEK LİSANS TEZİ

DANIŞMAN
Doç. Dr. Gazi Erkan BOSTANCI

ANKARA
2021

TÜRKİYE CUMHURİYETİ
ANKARA ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ

ADLİ BİLİŞİMDE MAKİNE ÖĞRENMESİ: MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE TERÖR OLAYLARININ TAHMİN EDİLMESİ ÇALIŞMASI

Burak GÖRMEZ

DİSİPLİNLERARASI ADLİ BİLİMLER ANABİLİM DALI
YÜKSEK LİSANS TEZİ

DANIŞMAN
Doç. Dr. Gazi Erkan BOSTANCI

ANKARA
2021

ETİK BEYAN

Ankara Üniversitesi
Sağlık Bilimleri Enstitüsü Müdürlüğü'ne,

Yüksek Lisans tezi olarak hazırlayıp sunduğum "Adli Bilişimde Makine Öğrenmesi: Makine Öğrenmesi Algoritmaları İle Terör Olaylarının Tahmin Edilmesi Çalışması" başlıklı tez; bilimsel ahlak ve değerlere uygun olarak tarafımdan yazılmıştır. Tezimin fikri tümüyle tez danışmanım ve bana aittir. Tezde yer alan deneysel çalışma/araştırma tarafımdan yapılmış olup, tüm cümleler, yorumlar bana aittir.

Yukarıda belirtilen hususların doğruluğunu beyan ederim.

Öğrencinin Adı Soyadı: Burak GÖRMEZ
Tarih: 12.05.2021
İmza:

KABUL VE ONAY

Ankara Üniversitesi Sağlık Bilimleri Enstitüsü
Disiplinlerarası Adli Bilimler Anabilim Dalı
Adli Bilişim İkinci Öğretim Tezli Yüksek Lisans Programında

Burak GÖRMEZ tarafından hazırlanan

"Adli Bilişimde Makine Öğrenmesi: Makine Öğrenmesi Algoritmaları İle Terör Olaylarının Tahmin Edilmesi Çalışması" adlı tez çalışması

aşağıdaki jüri tarafından **YÜKSEK LİSANS TEZİ** olarak

OY BİRLİĞİ ile kabul edilmiştir.

22.06.2021

Doç.Dr. Gazi Erkan BOSTANCI
Ankara Ü. Mühendislik Fakültesi
Jüri Başkanı

Doç.Dr. Mehmet Serdar GÜZEL
Ankara Ü. Mühendislik Fakültesi
Üye

Doç.Dr. İhsan Tolga MEDENİ
Ankara Yıldırım Beyazıt Ü. İşletme Fakültesi
Üye

Tez hakkında alınan jüri kararı, Ankara Üniversitesi Sağlık Bilimleri Enstitüsü Yönetim Kurulu tarafından onaylanmıştır.

Prof.Dr. Fügen AKTAN
Sağlık Bilimleri Enstitüsü Müdürü

İÇİNDEKİLER

ÖNSÖZ

Günümüz medeniyetine yönelik en önemli tehditlerden ve insanoğlunun karşı karşıya kaldığı en önemli sorunlardan biri terörizmdir. Dolayısıyla, terörizme müdahale birçok ülke için ulusal güvenliğin ana gündemlerinden biri haline gelmiştir. Aynı zamanda, bilgi teknolojilerinin gelişmesiyle birlikte öngörülerde bulunarak stratejik kararlar alabilecek uygulamaların sağlık, ekonomi, güvenlik ve istihbarat gibi çok çeşitli alanlarda karar alınması amacıyla kullanılması mümkün hale gelmiştir. Bu doğrultuda, güvenlik güçleri ve ilgili kurumlar tarafından alınan tedbirlere ve yapılan sosyal çalışmalara ek olarak, veri biliminin de terörizmle mücadelede katkı sağlayabileceği dikkate alınarak bu çalışmada terörizm verileri ve çeşitli makine öğrenmesi algoritmaları kullanılarak söz konusu verilerin sınıflandırılması suretiyle herhangi bir saldırıdan sorumlu olabilecek terörist grubun mümkün olan en yüksek doğrulukla tahmin edilebilmesine yönelik modeller geliştirilmiş ve sonuçları incelenmiştir.

Yüksek lisans eğitimim süresince çalışmaya yönlendiren ve tez çalışmalarım boyunca bilimsel katkı, tecrübe ve deneyimlerini benden esirgemeyen değerli hocam Doç. Dr. Gazi Erkan BOSTANCI'ya, tez yazımımda bana yardımcı olan değerli çalışma arkadaşlarıma, ev ve sosyal hayatımda bana hep destek olan, yardımını esirgemeyen, moral motivasyonumu artıran aileme sonsuz teşekkürü borç bilirim.

Burak GÖRMEZ
ANKARA-2021

KISALTMALAR

API	Application Programming Interface
AUC	ROC Eğrisi Altında Kalan Alan
CFS	Correlation-Based Öznitelik Seçim Algoritması
CPU	Central Process Unit
DN	Doğru Negatif
DP	Doğru Pozitif
DS	Decision Stump Algoritması
DT	Karar Ağacı
DVM	Destek Vektör Makineleri
GPU	Graphics Processing Unit
GTD	Global Terrorism Database
KNN	K Nearest Neighborhood
LR	Lojistik Regresyon
NB	Naive Bayes
OOB	Out of Bag Veri Seti
PCA	Principal Component Analysis
PGIS	Pinkerton Global Intelligence Services
RBF	Radial Basis Function
RELU	Rectified Linear Unit
RF	Random Forest
ROC	Alıcı İşlem Karakteristiği Eğrisi
SVM	Support Vector Machine
YN	Yanlış Negatif
YP	Yanlış Pozitif
YSA	Yapay Sinir Ağları

ŞEKİLLER

ÇİZELGELER

1. GİRİŞ

Dünya, geçtiğimiz yıllar boyunca kayda değer sayıda terör olayına tanık olmuştur. Mevcut durumda da günümüz medeniyetine yönelik en önemli tehditlerden ve insanoğlunun karşı karşıya kaldığı en önemli sorunlardan biri, tüm dünyadaki insanların yaşam kalitesini etkileyen terörizmdir. Barış ve ülke güvenliği ile toplumun normal rutini üzerinde doğrudan etkisi bulunan ve siviller üzerinde korku yaratmanın bir yolu olarak kullanılmakta olan terörizm toplu şiddet olaylarının bir türüdür. Kökeni Latince "*terre*" kelimesine dayanmakta olan ve büyük korku, dehşet gibi anlamlara gelen terörizm, dünyanın dört bir yanındaki ülkelerin karşı karşıya olduğu en büyük tehditlerden biridir. En basit haliyle terörizmde, korku yaratmak suretiyle bazı siyasi, maddi veya dini hedeflere ulaşmak amacıyla kasıtlı olarak ayrım gözetmeksizin genel nüfusa yönelik yasadışı güç ve şiddet kullanılması söz konusu olmaktadır (Saha ve ark., 2017).

Gençleri, kadınları ve bir bütün olarak toplumu hedef almakta olan terörizmde son yirmi yılda sürekli bir artış gözlemlenmektedir. Terör olaylarının sebepleri arasında ise siyasi ve sosyal adaletsizlik bulunduğunun iddia edilmesi, şiddete yönlendiren inanışların takip edilmesi ve cehalet sayılabilir. Ancak, sebebi her ne olursa olsun terörizm, can kaybına sebep olmakta, ayrıca ülkelerin iş gücü, altyapı ve güvenlik harcamalarının artması sonucunu doğurmaktadır (Krieger ve Meierrieks, 2011). Dolayısıyla, ülke ekonomisi üzerinde de doğrudan bir etkisi bulunmaktadır. Ayrıca terör olayları, ön planlama, karmaşıklık ve işbirliğini içermektedir. Bu sebeple, terörizme müdahale birçok ülke için ulusal güvenliğin ana gündemlerinden biri haline gelmiş olup, toplumdaki bireylerin yaşamlarını korumak ve genel olarak yaşam kalitesini artırmak için terörizmle mücadelede farklı tedbirler uygulanmaktadır.

Terörizmle mücadelede, gerçekleşen saldırıların faillerinin tespit edilmesi ve gelecekteki saldırıları önlemek için teknikler ve taktikler geliştirilmesi en somut tedbirlerdendir. Dolayısıyla, ülkelerin terörle mücadeleden sorumlu kurumları tarafından, fail grupları bastırmak için gerekli önlemlerin alabilmesi ve önlemler

sayesinde terörizm riskinin azaltabilmesi için terör saldırılarından sorumlu terörist grupların etkin bir şekilde belirlenmesi önem arz etmektedir. Ayrıca, bir terör olayından hemen sonra sorumlu olabilecek terörist gruplar hakkında tahmin yapılabilmesi reaktif bir strateji uygulanabilmesi açısından kolluk kuvvetleri için de yararlı olacaktır. Ancak, özellikle karmaşık, senkronize ve iyi planlanmış terör eylemleri dikkate alındığında, çoğu zaman olayı gerçekleştiren terörist gruba ilişkin bulgular üzerinden tahminde bulunulabilmesi çeşitli faktörlere bağlı belirsizlikler içeren bir görevdir.

Diğer taraftan, bilgi teknolojilerinin gelişmesiyle birlikte verilerin üretilmesi, toplanması ve saklaması eskine kıyasla daha az maliyetli ve daha kolay bir hale gelmiştir. Ancak bu kolaylık ve maliyet avantajı ile birlikte teknoloji kullanımının artması sebebiyle hızla artan veriler neticesinde çok fazla miktarda veri depolanması durumu ortaya çıkmıştır ki bu da veri içerisinde saklı bulunan ve çalışmanın amacına göre faydalı olabilecek bilgilerin ortaya çıkarılması işlemini zorlaştırmıştır. Bu noktada öngörülerde bulunarak stratejik kararlar alabilecek yazılımların kullanılmasının birçok avantajı bulunmaktadır. Literatürdeki çalışmalar incelendiğinde bu tür yazılımların matematik ve istatistik bilimlerinin de yardımıyla sağlık, ekonomi, güvenlik ve istihbarat gibi çok çeşitli alanlarda öngörüye katkıda bulunarak karar alınması amacıyla kullanılmakta olduğu anlaşılmaktadır. Dolayısıyla, bilgi teknolojilerinin, terörle mücadelede de önemli bir araç olarak kullanılması mümkündür. Şöyle ki, bilgi teknolojisi tekniklerinin, terörizm ile ilgili verilerin toplanması, analiz edilmesi ve işlenmesinde kullanılması suretiyle değerli bulgular elde edilerek tahminler yapılabilmesi mümkündür. Örneğin, veri madenciliği, doğal dil işleme, bağlantı analizi, sosyal ağ analizi gibi güncel teknikler kullanılarak ham veriler yasal kovuşturma açısından uygun bir istihbarat olarak sunulabilir. Özellikle veri madenciliği ve makine öğrenmesi, insan yardımı kullanılarak bulunması çok zor olan devasa veri kümesindeki bağlantıları ve modelleri keşfetmeye yardımcı olacağından devlet kurumları ve özel kuruluşlar tarafından terörizmle mücadelede yoğun olarak kullanılabilir. Dolayısıyla, güvenlik güçleri ve ilgili kurumlar tarafından alınan tedbirlere ve yapılan sosyal çalışmalara ek olarak, veri biliminin sağlık, finans, askeriye ve eğitim gibi alanlar başta olmak üzere her alanda kullanılmakta olduğu da

dikkate alındığında, terörizmle mücadelede ve adli olaylarda sağlayacağı katkı göz ardı edilemez.

Terörizmle mücadelede katkı sağlayacağı değerlendirilen makine öğrenmesi en basit tabiri ile geliştirilen yazılımların veriden öğrendikleri deneyimlerle problem çözmeyi öğrenmesi olarak tanımlanabilir (Samuel, 1988). Yani bu durumda, yazılımlar problem çözmeyi veriyi kullanarak kendi kendine öğrenmektedir. Makine öğrenmesinde sonuçların, sınıflandırılmasına, kümelenmesine veya tahmin edilmesine yönelik algoritmalar kullanılmaktadır. Bilinmeyen bir değişkenin, bilinen sınıflardan hangisine ait olduğunu bulmayı amaçlayan sınıflandırma ise makine öğrenmesinde sıklıkla kullanılan yöntemlerden biridir.

Bu kapsamda, terörizm verileri ve çeşitli makine öğrenmesi algoritmaları kullanılarak oluşturulan modeller aracılığı ile söz konusu verilerin sınıflandırılması suretiyle herhangi bir saldırıdan sorumlu olabilecek terörist grubun mümkün olan en yüksek doğrulukla tahmin edilebilmesi, ilgili kurumların terörizmle mücadelesi açısından oldukça değerli, diğer adli incelemeler açısından ise yol gösterici olacaktır. Ayrıca, geçmiş terör saldırılarına ilişkin bazı verilerin yakın gelecekteki terörist eylemlerle doğrudan bağlantılı olabileceği veya aynı örgüt tarafından ardışık olaylar gerçekleştirilebileceği dikkate alındığında olay gerçekleştikten sonra makine öğrenmesi modelleri sayesinde işgücü ve zamandan tasarruf sağlayarak biran evvel fail gruplar üzerine yoğunlaşılmasıyla olası yeni saldırıların engellenmesi de mümkün olabilecektir.

Bu tez çalışmasında, bir terör olayı meydana geldikten sonra olayın hangi terörist grup tarafından yapıldığına ilişkin tahminde bulunulması ve incelemelerin bu doğrultuda detaylandırılabilmesi amacıyla Küresel Terörizm Veritabanı üzerinde çeşit makine öğrenmesi algoritmaları ile sınıflandırma işlemi gerçekleştirilmiş ve sonuçları değerlendirilmiştir. Bu doğrultuda, *Python* programlama dili kullanılarak Küresel Terörizm Veri Tabanı üzerinden sorumlu terörist grupların tahmin edildiği uygulama gerçekleştirilmiş olup, söz konusu uygulamada yapay sinir ağları (ANN/YPA) ve

Naive Bayes algoritması (NB), lojistik regresyon algoritması (LR), k-en yakın komşu algoritması (KNN), karar ağacı algoritması (DT), rastgele orman algoritması (DT) ve destek vektör makineleri (SVM/DVM) kullanılmıştır. Çalışma kapsamında öncelikle veri seti analiz edilerek makine öğrenmesi algoritmaları açısından anlamlı hale getirilmesi için veri seti üzerinde veri ön işleme süreci gerçekleştirilmiş, devamında ise bahsedilen algoritmalarla modeller oluşturularak modellerin sonuçları karşılaştırılmıştır.

Bu tezin amacı doğrultusunda kavramsal temellerin oluşturulması için GEREÇ VE YÖNTEM bölümünde, veri ön işleme süreçlerine, makine öğrenmesine ve adli bilişimle ilişkisine, tez çalışmasında kullanılan algoritmalar hakkındaki temel bilgilere ve algoritmalar kullanılarak geliştirilen modellerin performans ölçüm kriterlerine değinilmiştir. Küresel Terörizm Veritabanı'ndan elde edilen verilerin analiz edilmesine ve hazırlanmasına, algoritmalar kullanılarak terör olaylarının hangi terörist gruplar tarafından gerçekleştirdiğinin tahmin edilmesine yönelik modeller oluşturulmasına, geliştirilmesine ve bulguların elde edilmesine BULGULAR bölümünde değinilmiştir. TARTIŞMA bölümünde ise BULGULAR bölümünde yapılan çalışmalar sonucunda elde edilen çıktılar ile bu çıktıların analiz edilmesi üzerinde durulmuştur. Son olarak, SONUÇ VE ÖNERİLER bölümünde, tezin tamamını kapsayan bir değerlendirme yapılarak, gerek terörizm gerekse de adli bilişim ile ilgili gelecekte yapılacak çalışmalar açısından yol gösterici olması için önerilerde bulunulmuştur.

Bu kapsamda, bu tez çalışmasının amacı, örnek bir veri üzerinden sorumlu grupları ve ilgili gerçekleri bulacak modelleri uygulayıp test ederek ham verilerin güvenlik uzmanlarının çalışmalarında kullanılabilecek yararlı bilgiye dönüştürülmesi ve bu bilgiler doğrultusunda tahminlerde bulunulabilmesi aşamalarının gösterilmesidir.

1.1. Literatür Özeti

Terörizm toplumları ve insanların yaşamlarını son derece kötü bir şekilde etkilemektedir. Bu nedenle, hemen hemen her ülke terör saldırılarını önlemek için önleyici bir mekanizma geliştirmeye odaklanılmaktadır. Dolayısıyla konu, nedenlerini anlamak ve terörist faaliyetlerin olasılığını azaltarak etkili bir terörle mücadele mekanizmasının nasıl geliştirileceğini anlamak için kapsamlı bir şekilde incelenmektedir. Ayrıca, son yıllarda makine öğrenimi terörist saldırılarla ilgili sınıflandırma sorunlarını çözmek için yaygın olarak kullanılmakta ve araştırmacılar tarafından makine öğrenmesine ile ilgili olarak önerilen çeşitli sınıflandırma yöntemleri bulunmaktadır.

Bu kapsamda, tez konusu çalışmanın uygulanmasından önce, sınıflandırma ve tahmin için kullanılmakta olan makine öğrenmesi yöntemleri ve konu ile ilgili olarak yapılan önceki çalışmalar üzerinde yeterli araştırma yapılmış olup, bu bölümde söz konusu araştırmanın özeti sunulmaktadır.

Iqbal ve ark. (2013), ABD'nin farklı eyaletlerinde işlenen suçların hangi kategoriye ait olduğunu tahmin etmek için NB ve DT algoritmalarını incelemiş ve bu iki algoritmayı karşılaştırmıştır. Çalışmada sınıflandırıcıların doğruluğunu test etmek için giriş veri setine hem NB hem de DT için ayrı ayrı 10-kat çapraz doğrulama uygulanmış olup, DT algoritmasının, NB algoritmasına kıyasla daha başarılı bir şekilde suç kategorilerini % 83,951 doğruluk ile tahmin ettiği sonucuna ulaşılmıştır (Iqbal ve ark., 2013).

Gohar ve ark. (2014), Pakistan'daki terörist grupların sınıflandırılması ve tahmini için NB, KNN, ID3 ve DS olmak üzere dört ayrı algoritmadan oluşan ve çoğunluk oyuna dayalı olan yeni bir topluluk sistemi önermiştir. Çalışma sonucunda, geleneksel algoritmaların ayrı ayrı uygulanması ile elde edilen sonuçlar ile oluşturulan yeni sistemin sonuçları karşılaştırılmış ve yeni sistemin ayrı ayrı uygulanan

algoritmalara kıyasla daha iyi bir doğruluk seviyesi elde ettiği belirtilmiştir (Gohar ve ark., 2014).

Khorshid ve ark. (2015), Orta Doğu ve Kuzey Afrika'da 2004-2008 yılları arasında meydana gelen terör olaylarının faili terörist grupların tahmin edilmesi amacıyla SVM, DT ve NB algoritmalarını kullanan bir model geliştirmiştir. Çalışmada veri seti olarak GTD kullanılmıştır (Khorshid ve ark., 2015).

Ozgul ve ark. (2009), terörist grupların sınıflandırılması ve tahmin edilmesi amacıyla NB, KNN, *Iterative Dichotomiser*, ve DS olmak üzere dört farklı algoritma kullanılarak bir topluluk sistemi önermiş olup, söz konusu sistem ile bireysel modellere kıyasla daha iyi sonuçlar elde edildiği ifade edilmiştir (Ozgul ve ark., 2009).

Toure ve Gangopadhyay (2016), farklı konumların maruz kalabileceği terörizm riskinin hesaplanabilmesine yönelik bir model geliştirmek için gerçek zamanlı bir sistemden olay verilerini toplamışlardır. Ayrıca, söz konusu kişiler tarafından olası terör olaylarının tahmin edilmesine yönelik bir dizi kural da önerilmiştir. Bu çalışmada, %96'ya ulaşan bir doğruluğun elde edildiği belirtilmektedir (Toure ve Gangopadhyay, 2016).

Saha ve ark. (2017), yapılan başka bir çalışmada, gerçekleşen terör olayları sonrasında saldırı türü, kullanılan silah türü ve hedef türüne yönelik tahminde bulunulmasını sağlayan bir model geliştirmiş olup, bahse konu modelin %79 ila %86 aralığında bir doğruluğa ulaştığını belirtilmiştir (Saha ve ark., 2017).

Mo ve ark. (2017), GTD veri tabanı üzerinden terör olaylarının tahmin edilmesine yönelik bir çalışma yapmış olup, söz konusu çalışmada SVM, NB ve LR algoritmaları kullanılmış ve %78'e varan doğruluk elde edilmiştir (Mo ve ark., 2017).

Verma ve ark. (2018), SVM, ANN, NB, DT ve RF algoritmaları kullanılarak saldırı türü, saldırı bölgesi ve silah türü hakkında tahminlerde bulunabilen bir model geliştirmiş olup, modelin yaklaşık %90 doğruluğa ulaştığı bildirilmiştir (Verma ve ark., 2018).

Li ve ark. (2018), terörist grupların davranışlarını ve dinamiklerini anlayarak gerçekleşen bir olaydan sorumlu olabilecek grubun tahmin edilmesi amacıyla sosyal ağ analizi ve örüntü tanıma yaklaşımlarını kullanan bir çalışma gerçekleştirmiş olup, çalışma sonucunda yüksek oranlarda doğru tahminlere ulaşıldığı ifade edilmiştir (Li ve ark., 2018).

Agarwal ve ark. (2019), GTD'nin veri setini analiz ederek terörizmle mücadelede etkili olabilecek farklı faktörler hakkında tahminlerde bulunmaya odaklanmıştır. Bu doğrultuda, söz konusu çalışmada, veri setinin anlaşılması ve terörist olayların başarıya ulaşıp ulaşmadığı ile eylemin hangi örgüt tarafından gerçekleştirildiği gibi hususların tahmin edilebilmesi için SVM, RF ve LR algoritmalarından yararlanılmıştır (Agarwal ve ark., 2019).

Agarwal ve ark. (2019), farklı terörist faaliyetleri gruplamak ve tahmin etmek için KNN ve RF algoritmaları kullanılarak sınıflandırıcı modeller geliştirmişlerdir. Söz konusu çalışmada GTD veri seti kullanılmıştır (Kalaiarasi ve ark., 2019).

Maniraj ve ark. (2019), terörist grupların büyümesini veya küçülmesini zaman, bölge, hedef motivasyonu ve saldırılarda kullanılan silah türüne göre inceleyen bir sistem geliştirmiş olup, çalışmalar kapsamında GTD veri seti analiz edilmiştir (Maniraj ve ark., 2019).

Christie (2019), tez çalışmasında makine öğrenmesi algoritmalarını kullanarak Pakistan'da gerçekleşen ve sorumluluk üstlenilmemiş terör olaylarının dinamiklerinin anlaşılması üzerine bir çalışma gerçekleştirmiştir. Söz konusu çalışma kapsamında, saldırı türü, hedef, silah türü gibi terörist olaylarla ilgili nitelikler üzerine tahminlerde

bulunularak sorumluluk üstlenilmemiş olan terör olayları bilinen terörist gruplarla eşleştirilmeye çalışılmıştır (Christie, 2019).

Bu tez çalışmasının amacı ise örnek bir veri üzerinden sorumlu grupları ve ilgili gerçekleri bulacak modelleri uygulayıp test ederek ham verilerin güvenlik uzmanlarının çalışmalarında kullanılabilecek yararlı bilgiye dönüştürülmesi ve bu bilgiler doğrultusunda tahminlerde bulunulabilmesi aşamalarının gösterilmesidir.

2. GEREÇ ve YÖNTEM

Bu tez çalışmasında, GTD veri setinin analiz edilmesiyle oluşturulan veri seti üzerine bir makine öğrenmesi yöntemi olan sınıflandırma işlemine ait algoritmalarının uygulanması suretiyle oluşturulan modeller kullanılarak herhangi bir terör olayı gerçekleştiğinde olayı gerçekleştiren terörist gruba yönelik tahminlerde bulunulabilmesi üzerinde durulmuştur. Bahse konu algoritmaların uygulamasına geçilmeden önce bu bölümde, çalışma ile ilgili temel konularının kısa bir açıklamasının verilmesi amaçlanmaktadır. Bu doğrultuda, öncelikle makine öğrenmesi süreci ve bu sürecin adli bilişimle ilişkisi hakkında bazı bilgilerden bahsedilmiştir. Daha sonra çalışmada kullanılan veri setinin hazırlanması yöntemlerine ve uygulanan algoritmalara yönelik genel bilgilere değinilmiştir. Son olarak ise oluşturulan modellerin performanslarının ölçülmesinde kullanılan kriterler ve geliştirilmesinde kullanılan yöntemler hakkında bilgiler verilmiştir.

2.1. Makine Öğrenmesi ve Adli Bilişim

Yaşadığımız çağ dikkate alındığında, hemen hemen her ülkede sağlık, eğitim, finans, güvenlik ve ulaşım gibi birçok alanda teknoloji kullanımının oldukça yaygın bir hal aldığı görülmektedir. Teknoloji kullanımının bu denli yaygınlaşması ise gerçek ve tüzel kişiler tarafından üretilmekte olan veri miktarının her geçen gün artması sonucunu doğurmuştur. Aynı zamanda, yine teknolojinin gelişmesiyle birlikte verilerin depolanması ve verilere ulaşılması eskisine kıyasla hem daha kolay hem de daha az maliyetli olmaya başlamıştır. Bunlara bağlı olarak, dünyamızda veri finansman kadar değerli sayılmaya başlamıştır. Öyle ki, çoğu elektronik ortamda hizmet vermekte olan yeni nesil banka ve aracı kuruluşlar fon transferi, pay senedi alımı gibi işlemleri komisyon ücreti almadan yapmaya başlamışlardır. Bahsedilen işlemlerin ücretsiz olarak yapılmasının temelinde ise fazla miktarda veriye sahip olma istediğinden dolayı müşteri tabanının genişletilmesi amacı yatmaktadır. Çünkü daha fazla miktarda veriye sahip olunması durumunda stratejik kararlar alınması ve ileriye dönük tahminlerde bulunulabilmesi mümkün olmaktadır. Örneğin, Amerika Birleşik

Devletleri'nde yatırımcıların yoğun olarak talep gösterdiği pay senetlerinden ülkede gerçekleşen seçim sonuçlarının tahmin edilebilmesi dahi söz konusu olmaktadır. Böyle bir durumda da bahse konu tahminin yapılabilmesi imkânını sağlayan müşteriye ve müşterinin yaptığı işlemlere ilişkin veriler, müşteriden alınacak komisyon gibi ücretlerden daha kıymetli olmaktadır. Ancak hemen belirtmek gerekir ki yukarıda bahsedildiği şekilde veri miktarında meydana gelen artış nedeniyle veri üzerinden bahsedilen tahminlerin yapılabilmesi olayı insanların tek başlarına yapabilecekleri konumda değildir. Esas itibariyle, devasa veriler üzerinden insanların tahminde bulunabilmesi, çok büyük tecrübe, ilgili alanda uzmanlık ve öngörü yeteneği gerektirmektedir. Bu tecrübe, uzmanlık ve yeteneğe her insanın sahip olamayacağı gerçeği göz ardı edilerek bu vasıflara sahip olunduğu farz edilse bile insanlar tarafından devasa verinin analiz edilmesi ve tahminde bulunulması sürecinin alacağı süre dikkate alındığında bu işlemin çok da etkili olmayacağının söylenmesi mümkündür. Bu noktada, temelinde veriden öğrenme felsefesi olan ve birçok alanda kullanılan makine öğrenmesi verilerin analiz edilmesi konusunda ön plana çıkmaktadır.

Sanayileşmenin ilk aşamalarında ve devamındaki dönemlerde kullanılmakta olan makineler ile günümüzde kullanılmakta olan makineler karşılaştırıldığında bilgisayarların hayatımıza girmesinin makine kavramını çok farklı bir boyuta taşıdığı anlaşılmaktadır. Öyle ki artık günümüzde makine denildiğinde içerisinde yazılımların bulunduğu bilgisayar, telefon, klima, araba ve televizyon gibi birçok donanım akla gelmektedir. Makine öğrenmesi bağlamında üzerinde durulması gereken husus ise bahse konu donanımlara öğrenme yeteneğinin bilgisayar yazılımları ile kazandırılıyor olmasıdır. Söz konusu yazılımlarda ise donanımların bir görevi yapmak üzere açıkça programlanmasının yerine görevi yapmayı öğrenebilen programların geliştirilmesi yaklaşımı kullanılmaktadır. Bir başka deyişle, geleneksel programlamada, bir görevin yerine getirilmesi için programcılar tarafından gerekli her adım kodlanmakta iken makine öğrenmesinde algoritmalar veri üzerinde eğitim alarak aynı görevi kendileri çözmeyi öğrenmektedir (Samuel, 1988). Benzer şekilde, Blum tarafından da makine öğrenmesinin; verilerden kural öğrenme yeteneği olan, değişiklikler kapsamında kendini güncelleyebilen ve edindiği deneyimlerle birlikte performansını ilerleten

yazılımları tasarlamakla ilgilenmekte olduğu belirtilmiştir (Blum, 2021). Dolayısıyla, makine öğrenmesinde veri kullanılarak yazılımların kendi kendine bir öğrenme gerçekleştirmesi söz konusudur. Bu çerçevede makine öğrenmesi alanında özetle;

- Öğrenebilen makine denildiğinde bilgisayarlı sistemlerin anlaşılması ve bu sistemlerde öğrenmeyi sağlayan unsurun bilgisayar yazılımları olması,
- Yapılan tanımlarda deneyimle öğrenme ve performansın artması konularının vurgulanarak performansın deneyimlerle birlikte artmasının öğrenme olarak kabul edilmesi,
- Deneyimin esas itibariyle öğrenme olayının gerçekleştirilebileceği veri setleri olması,
- Öğrenmenin, veri setinde bulunan ve işin amacına göre faydalı olacak bilgilerin ortaya çıkarılması ve meydana gelen değişikliklere yazılımın kendi kendine uyum sağlayabilmesi olduğu,
- Hangi düzeyde öğrenmenin gerçekleştiğinin test edilmesi amacıyla bazı kriterlerin kullanılması,
- Öğrenme düzeyinin geliştirilmesi için oluşturulan modellerde parametre ayarlaması yapılması,

hususlarının konunun önemli noktaları olarak sayılması mümkündür.

Makine öğrenmesi temel olarak denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme olmak üzere üç kategoriye ayrılmaktadır. Denetimli öğrenme, hem giriş hem de çıkış parametrelerinden oluşan örneklere dayalı olarak geliştirilen model ile yeni girdilerin çıktılarla eşleştirilmesinin yapılmasıdır. Öncelikle belirtmek gerekir ki makine öğreniminde söz konusu örnekleri temsil eden veriler *Excel* dosyasındaki sütunlara benzeyen özniteliklerden oluşmaktadır. Öznitelik ölçülebilir bir özelliktir ve terörist olayların tahmin edilmesi gibi bir uygulamada saldırı türü, saldırıda kullanılan silah türü, saldırının hedef türü gibi niteleyici özellikleri içerebilir, bu durumda çıktı değişkeni ise eylemi gerçekleştiren terörist grup olabilir. Dolayısıyla, denetimli öğrenmede, modelin geliştirilmesinde kullanılan örneklerdeki öznitelikler

analiz edilir ve bu öznitelikler ile etiketi bulunan yani sınıfı belirli olan çıktı değişkeni arasındaki ilişki anlaşılmaya çalışılır. Bu işlem sonucunda, modelin daha önce görmediği örneklerden çıktıları tahmin etmek için kullanılabilecek bir fonksiyon üretilir. Daha sonra bu fonksiyon kullanılarak modelin geliştirilmesinde kullanılmayan ancak çıktı değerleri bilinen test verisi kullanılarak tahminlerde bulunulur ve elde edilen sonuçlar gerçek sonuçlarla karşılaştırılır. Bu karşılaştırma, tahminlerin gerçek değerlerden ne kadar saptığını ölçen kayıp fonksiyonları kullanılarak yapılır. Sonuç olarak, tahminlerin yeterli düzeyde doğru olduğunun anlaşılması durumunda algoritmanın genelleştiği yorumu yapılır.

Denetimli öğrenmenin aksine denetimsiz öğrenmede modelin oluşturulmasında kullanılacak eğitim verisinin etiketlenmiş herhangi bir çıktısı bulunmamaktadır. Yani, bu yöntemde yalnızca giriş verileri bulunmakta olup temel amaç veri kalıplarının anlaşılmasıdır. Bu yöntem çoğunlukla verilerin kümelenmesi ve veri setinde bulunan anomalilerin algılaması ile ilgili uygulamalarda kullanılmaktadır.

Pekiştirmeli öğrenmede ise yazılımların yüklü olduğu ajanların bir görevi yerine getirmek üzere doğrudan programlanmasının yerine söz konusu yazılımların çevreleriyle etkileşime girerek öğrenmesi söz konusudur. Bu öğrenme işlemi, karşılaşılan durum karşısında alınan aksiyona bağlı olarak ödül kazanılması veya ceza yenilmesi yoluyla gerçekleşmekte olup, temel olarak deneme yanılma yöntemine dayanmaktadır.

Literatürde herhangi bir problemin makine öğrenmesi yardımıyla çözülmesi için benzer yaklaşımlar bulunmaktadır. Makine öğrenmesinin uygulanma aşamaları;

1) Modelin geliştirileceği veri setinin toplanması,
2) Toplanan veri setinin algoritmalar açısından uygun hale getirilmesi için verilerin temizlenmesi, uygun formata dönüştürülmesi ile eğitim ve test verisine ayrılması gibi işlemlerden oluşan veri ön işleme aşamasının gerçekleştirilmesi,

3) Algoritmalar ve yukarıda belirtilen aşamalar sonucunda oluşturulan eğitim veri seti kullanılarak modelin oluşturulması,

4) Model oluşturulduktan sonra yukarıda belirtilen şekilde elde edilen test veri seti kullanılarak modelin değerlendirilmesinin yapılması,

5) Sonuçların istenilen düzeyde olması durumunda uygulamanın canlı ortama taşınması

olarak özetlenebilecektir (Kartal, 2015).

Yukarıda açıklanan makine öğrenmesinin, adli bilişim uygulamalarına olası etkilerini değerlendirmeden önce adli bilişim alanının genel işleyiş mekanizmasından bahsedilmesi doğru olacaktır. Bu doğrultuda, teknolojinin ilerlemesi nasıl ki veri bilimi ve makine öğrenmesi gibi alanların gelişmesine katkı sağladıysa aynı zamanda suç ve suçlu unsurlarının da elektronik ortamda yaygınlaşmaya başlamasına sebep olmuştur. Yani bilgisayar gibi cihazların hayatımıza girmesiyle birlikte artık günümüzde elektronik ortamda gerçekleşen ve bu ortamda iz bırakmakta olan suçlar bulunmakta olup, bahse konu bilgisayar benzeri cihazlar suçun hem bir unsuru hem de aracı olarak karşımıza çıkmaktadır. Bu noktada adli bilişim; bilgisayar, cep telefonu ve harici diskler gibi veri depolama kabiliyeti olan elektronik cihazlar bünyesinde bulunan verilerin yasal mercilerce kabul edilen yöntemler kullanılarak delillendirilmesi üzerine yoğunlaşmaktadır. Bu amaç ile yapılacak çalışmalar ile ilgili olarak kabul edilmiş bir süreç olsa bile duruma göre uygulamada farklılıklar görülebilmesi mümkündür. Ancak genel olarak kabul edilmiş olan adli bilişim süreci;

- Delil özelliği taşıyabilecek unsurların toplanması ve tanımlanması,
- Toplanan unsurların koruma altına alınması,
- Yasal mercilerce kabul edilebilir okuma-yazma korumalı cihazlarla adli kopyaların yani verilerin elde edilmesi,
- Adli kopyalar üzerinde yazılımlar ile inceleme yapılması ve elde edilen hususların analiz edilmesi,

- Tüm çalışmalar kapsamında ortaya çıkan hususların kabul edilebilir formatta raporlanması

aşamalarından oluşmaktadır. Dolayısıyla, adli bilişimde söz konusu aşamalarda belirtilen işlemlerin uygulanması ile üzerinde çalışılan olaya ilişkin verilerin elde edilmesi ve neticesinde ilgili olayın açıklığa kavuşturulması amaçlanmakta olup, bu noktada ise adli bilişimde makine öğrenmesi yöntemlerinin uygulama alanı ortaya çıkmaktadır.

Bu doğrultuda, makine öğrenmesinin adli olaylarla ilgili olarak çok çeşitli uygulamalarının olması söz konusudur. En temel anlamda, makine öğrenmesi algoritmaları; suçlara ait büyük verilerin analiz edilmesi, suçlu davranışlarının ortaya çıkarılması, suç yapılarının tespit edilmesi gibi amaçlarla kullanılabilir. Çünkü suç tespiti için makine öğrenmesi algoritmaları kullanılması araştırmacılara sosyal ağlar, kablolu ağlar ve bulut ortamı gibi çok farklı kaynaklardan veri sorgulaması yapılması ve çok büyük boyutlara ulaşsa dahi bu verilerin analiz edilmesi imkânını sağlamaktadır. Bu durum esasen suçlular, terörist gruplar gibi bazı unsurların suç niyeti ve faaliyeti gibi bazı davranışlarının tahmin edilmesi amacıyla büyük miktarda verinin makine öğrenmesi algoritmaları ile analiz edilmesini ifade etmektedir. Örneğin, hırsızlık, kara para aklama ve siber saldırılar gibi alanlarda gelecekte suç oluşturacak davranışların tahmin edilmesi amacıyla geçmişte yaşanan olaylara ilişkin verilerden öğrenme işlemi gerçekleştirilebilmektedir. Ayrıca, adli olaylarda, kullanılan yazılımlar aracılığı ile uzaktan veri analizi yapılması da mümkün olup, adli olayları araştırmakta olan kişiler açısından makine öğrenmesinin en önemli avantajlarından bir tanesi de suça ilişkin verinin hassas bir şekilde anlaşılabilir ve kolay yorumlanabilir bir formata getirilebilmesidir (Nath, 2006). Burada önemli olan husus esasen araştırmacıların üzerinde çalışacağı verilerin elde edilmesi ve geçmiş olaylara ilişkin nitelikli verilerin ilgili kurumlarda tutuluyor olmasıdır.

Yukarı bahsedilen adli incelemeler alanında makine öğrenmesinin sağlayabileceği imkanlara, profil oluşturma ve tahminde bulunma faaliyetleri ile

suçların ne zaman ve nerede meydana gelebileceğine yönelik öngörüde bulunulması veya gerçekleşen bir suçta faillerin kimler olabileceğine yönelik tahminde bulunulması, benzer imkanlar tıp, finans ve adli olaylarla bağlantılı olan terörizmle mücadele gibi bir çok alanı için de söz konusudur. Öyle ki son zamanlarda ilgili kurumlar tarafından terörizmin farklı faktörlerinin incelenmesi amacıyla makine öğrenmesi algoritmaları sıklıkla kullanılmaktadır. Bu tez çalışmasında da, bahse konu alandaki çalışma ve gelecek uygulamalara ışık tutması açısından terörizm ile ilgili veri seti üzerinden detaylarına bölüm 3.'de yer verilen makine öğrenmesi modelleri oluşturulmuştur.

2.2. Sınıflandırma Algoritmaları

Bölüm 2.1'de, hem girdilerin hem de çıktıların veri setinde birlikte bulunduğu makine öğrenmesi kategorisinin denetimli öğrenme olduğu ifade edilmişti. Bu bölümde de, sınıflandırma hakkındaki genel bilgilere değinildikten sonra denetimli öğrenme kategorisine ait olan ve genel olarak tez çalışmasında kullanılan sınıflandırma algoritmaları hakkında temel bilgiler verilmektedir.

Verileri önceden belirlenmiş sınıflardan birine atamayı amaçlayan sınıflandırma, daha önce terörizm, istihbarat ve iç güvenlik gibi alanlarda da yaygın olarak kullanılmış olup, makine öğrenmesinin temel alanlarından biridir. Sınıflandırma algoritmaları ile oluşturulan modellerde, gerçekte bir sınıfa ait olan ancak hangi sınıfa ait olduğu model tarafından başlangıçta bilinmeyen veri parçasının bilinen gruplardan birine yerleştirilmesi amaçlanmaktadır (Harrington, 2012). Bu amacı yerine getirebilmek içinse daha önce modelin oluşturulmasında temel alınan eğitim veri setinde bulunan ve girdilerin hangi sınıfa ait olduğu yani etiketi bilinen örneklerden faydalanılmaktadır. Dolayısıyla, sınıflandırmada modelin oluşturulma aşamasında hem girdi parametreleri hem de çıktı değerleri bilinmektedir.

Veri setlerinin sınıflandırılması için çeşitli yaklaşımlar kullanılmakta olup, sınıflandırma algoritmalarının olasılık tabanlı ve olasılık tabanlı olmayanlar olarak

değerlendirilmesi mümkündür. Olasılık tabanlı sınıflandırıcılarda herhangi bir örneğin belirli bir sınıfa ait olma olasılığı gösterilmekte iken olasılık tabanlı olmayan yani ayrık sınıflandırıcılarda söz konusu örneğin ait olduğu tahmin edilen sınıfı gösterilmektedir. Ayrıca, bahse konu sınıflandırıcılar ikili veya çoklu sınıflandırıcılar olarak da ele alınabilmektedir. Burada ikili sınıflandırma, belirli bir veri setinde yer alan verilerin belirlenen sınıflandırma kuralı ile iki gruba ayrılmasını ifade etmekte iken çoklu sınıflandırıcılar ise ikiden fazla gruba ayrılmasını ifade etmektedir.

Bu çerçevede, tez çalışmasında kullanılan algoritmalara ilişkin genel bilgiler aşağıda sunulmaktadır.

2.2.1. Naive Bayes Sınıflandırıcısı

Naive Bayes algoritması, kolay uygulanması ve anlaşılması ile büyük veri setleri için kullanışlı olması nedeniyle yoğunlukla kullanılmakta olup, şartlı olasılık yaklaşımı ile sınıflandırma işleminin gerçekleştirildiği bir yöntemdir. Belirli bir sınıfta bir özniteliğin varlığının başka herhangi bir özniteliğin varlığı ile ilgili olmadığının varsayıldığı ve temelinde *Bayes* teoremi yer almakta olan bu algoritmada, elde bulunan ve hâlihazırda sınıflandırılmış olan verilerin kullanılması suretiyle yeni bir verinin mevcuttaki hangi sınıfa ait olduğuna ilişkin olasılık hesaplaması yapılmaktadır. Bu işlem sırasında, diğer sınıflandırma işlemlerinde olduğu gibi öğrenme işlemi eğitim veri seti üzerinden gerçekleştirilmektedir. *Naive Bayes* algoritmasının, basit bir şekilde uygulanabilmesinin yanı sıra bazı durumlara sofistike sınıflandırma yöntemlerini dahi geride bırakabildiği bilinmektedir.

Algoritmanın işleyişi ise ilk olarak veri setinin frekans tablosuna dönüştürülmesi ile başlamaktadır. Daha sonra olasılıklar hesaplanarak olasılık tablosu oluşturulmaktadır. Son olarak da her bir sınıfın sonsal olasılığını hesaplamak için aşağıda şekil 2.1'de verilen formül uygulanmaktadır. Örneklemin hangi sınıfa ait olduğuna yönelik tahmin ise en yüksek sonsal olasılığa sahip sınıf olarak yapılmaktadır.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Şekil 2.1 Naive Bayes Formülü.

Bahsedilen işleyiş örnek üzerinden gösterilmesi durumunda daha kolay anlaşılabilecektir. Bu doğrultuda, kullanılan silah türüne göre terörist eylemin hangi örgüt tarafından gerçekleştiğine yönelik tahminde bulunulması açısından eğitim veri setinden oluşturulan frekans tablosunun çizelge 2.1'deki gibi olduğu varsayılırsa, bu frekans tablosu kullanılarak oluşturulacak olasılık tablosu çizelge 2.2'deki gibi olmaktadır.

Çizelge 2.1 NB – Örnek Frekans Tablosu.

FREKANS TABLOSU		Eylemi Gerçekleştiren Örgüt	
		A Örgütü	B Örgütü
Kullanılan Silah Türü	Bomba	3	2
	Makineli Silah	4	0
	Kimyasal	2	3

Çizelge 2.2 NB – Örnek Olasılık Tablosu.

OLASILIK TABLOSU		Eylemi Gerçekleştiren Örgüt		
		A Örgütü	B Örgütü	Özniteliklerin Yani Silah Türünün Olasılığı P(x)
Kullanılan Silah Türü	Bomba	3/9	2/5	5/14
	Makineli Silah	4/9	0/5	4/14
	Kimyasal	2/9	3/5	5/14
	Sınıfların Yani Terör Gruplarının Önsel Olasılığı P(c)	9/14	5/14	

Bu durumda, yeni gerçekleşen bir eylemde kullanılan silah türü bomba ise terör gurubunun tahmin edilmesi noktasında ilgili formülün uygulanma adımları şu şekilde olmaktadır:

- Örnek kapsamında eylemde bomba kullanıldığında A Örgütü veya B Örgütü olma olasılığı yani sonsal olasılık bulunması gerekmektedir. Bu durum genel formülde P(c|x) olarak yani "x" olayı gerçekleştiğinde "c" olayının gerçekleşme olasılığı olarak ifade edilmektedir. Örnek açısından P(A Örgütü | Bomba) ve P(B Örgütü | Bomba) olarak ifade edilmelidir.

- Sonsal olasılığın hesaplanabilmesi için genel formülde P(c) yani sınıfların önsel olasılığı ile genel formülde P(x) yani özniteliklerin önsel olasılığının hesaplanması gerekmektedir. Bu hesaplamalar çizelge 2.2'de yer alan olasılık tablosunda gösterilmektedir.

- Formülde P(x | c) olarak ifade edilen olabilirliğin hesaplanması gerekmektedir. Bu durum örnek açısından A Örgütü'nün bomba ile veya ayrı ayrı olarak diğer silah türlerini kullanma olasılığını göstermektedir. Örnekte A Örgütü tarafından toplam 9 olay gerçekleştirilmiş ve bunların 3 tanesinde silah türü olarak bomba kullanılmıştır. Öyleyse P(Bomba | A Örgütü) = 3/9'dur. Bu değerler çizelge 2.2'de yer alan olasılık tablosunda da gösterilmekte olup P(Bomba | B Örgütü) = 2/5'dir.

- Tüm değişkenler elde edildikten sonra ilgili formülün uygulanması ile A Örgütü için ilgili olay kapsamında sonsal olasılık P(c | x) = P(A Örgütü | Bomba) = (0,33x0,64)/0,36 = 0,60 olarak hesaplanmaktadır. B örgütü içinse P(B Örgütü | Bomba) = (0,40x0,36)/0,36 = 0,40 olarak hesaplanmaktadır. Dolayısıyla *Naive Bayes* algoritması bahse konu olayın A Örgütü tarafından yapıldığı yönünde tahminde bulunacaktır.

Yukarıda bahsedilen örnekten de anlaşılabileceği üzere pratikte *Bayes* sınıflandırıcılar ile ilgili bazı dezavantajlar bulunmaktadır. Örneğin, söz konusu algoritmada önsel olasılıkların bilinmesini gerektirmekte olup, bilinmediği durumlarda arka plan bilgisi ve orijinal dağılımlar hakkında daha önceden mevcut olan veriler temelinde işlem yapılması gerekmektedir. Diğer bir dezavantaj ise belirli durumlarda en aza indirilmesi mümkün olsa da en iyi hipotezi bulmak için katlanılması

gereken hesaplama maliyetidir. Bahse konu algoritmanın, gürültülü verilere ve boş veya ilgisiz özniteliklere karşı dayanıklı olmasından dolayı farklı sınıflandırma problemlerini etkili bir şekilde çözebilmesi ise avantajlı noktası olarak öne çıkmaktadır.

2.2.2. Lojistik Regresyon Algoritması

Lojistik regresyon algoritması, adında her ne kadar regresyon ifadesi bulunsa da hem sınıflandırma hem de regresyon alanında kullanılan bir algoritmadır. Bu algoritma ile girdi parametresi olan öznitelikler yani bağımsız değişkenlerle örneklemlerin sınıfları yani bağımlı değişken tahmin edilmeye çalışılmaktadır. Diğer bir deyişle lojistik regresyon algoritmasında kategorik olması gereken bağımlı değişkenlerle bağımsız değişkenler arasındaki ilişkinin bulunması ve en iyi şekilde modellenmesi amaçlanmaktadır. Ayrıca, söz konusu algoritma, tahmin edilecek bağımlı değişken; iki kategoriye sahipse ikili regresyon, aralarında sıra bulunan ikiden fazla kategoriye sahipse sıralı lojistik regresyon, aralarında sıra bulunmayan ikiden fazla kategoriye sahipse çok kategorili lojistik regresyon olarak adlandırılmaktadır.

Yukarıda da belirtildiği gibi bağımlı değişkenin gerçekleşme olasılığı ile öznitelikler arasındaki ilişkinin anlaşılması amacıyla kullanılmakta olan lojistik regresyonda aşağıda şekil 2.2'de formülü verilen *sigmoid* fonksiyonu kullanılarak 0 ve 1 aralığında olan çıktı değeri elde edilir. Söz konusu çıktı değeri ve belirlenen eşik değere göre de sınıflandırma işlemi tamamlanır. Örneğin, bir sınıf için örneklemde yer alan öznitelikler kullanılarak elde edilen çıktı değeri 0,80 olarak bulunmuşsa ki bu durum bahse konu örneklemin %80 olasılıkla test edilen sınıfa ait olduğunu göstermektedir ve aynı zamanda eşik değer 0,5 olarak belirlenmişse, ilgili örneklem bu sınıfa dâhil edilecektir.

$$\frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

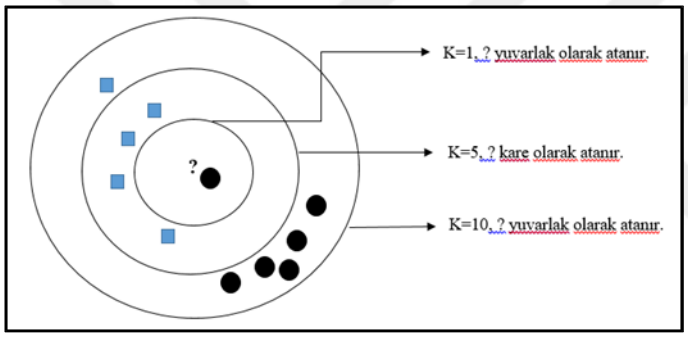
Şekil 2.2 Sigmoid Fonksiyonu.

Lojistik regresyon algoritması ile ilgili olarak göz önünde bulundurulması gereken konulardan biri bağımsız değişken olarak adlandırılan girdi parametrelerinin birbirinden de bağımsız olması yani aralarından yüksek düzeyde korelasyonun bulunmaması gerekliliğidir. Aksi durumda, yani özniteliklerin birbirlerinin fonksiyonu olarak yazılabilmesi durumunda eş doğrusallık oluşacaktır. Ayrıca bu algoritma özniteliklerin ölçeklendirilmesine ihtiyaç duymamaktadır. Ancak, lojistik regresyon algoritması öznitelikler çok fazla olduğunda veya özniteliklerde kategorik değişkenler fazla olduğunda çok iyi performans gösterememektedir.

2.2.3. K-En Yakın Komşu Algoritması

K-en yakın komşu algoritması sınıflandırma için yaygın olarak kullanılmakta olup mesafe hesabına dayalı bir algoritmadır. Söz konusu algoritma tembel öğrenme yöntemi ile sınıflandırma işlemini gerçekleştirmektedir. Şöyle ki, algoritmanın uygulanmasında, tüm eğitim verisi saklanmakta ve yeni bir örneklemin sınıflandırılması gerekene kadar model oluşturulmamaktadır. Yeni bir örneklemin sınıflandırılması ve tahmin edilmesi gerektiğinde ise diğer algoritmaların aksine ön işlemlerle belirlenen parametrelerin kullanılması yerine her defasında eğitim veri seti kullanılarak sınıflandırma işlemi tamamlanmaktadır. Böylece, eğitim veri setinin dağılımına yönelik herhangi bir olasılık varsayımı yapılmamaktadır. Dolayısıyla, k-en yakın komşu algoritması parametrik olmayan bir algoritmadır. Söz algoritmanın kullanılması için öncelikle sınıf değerleri belli olan örneklerden bir eğitim veri seti oluşturulması ve k parametresi ile kullanılacak mesafe fonksiyonun belirlenmesi gerekmektedir. Sonrasında, k-en yakın komşu algoritması kullanılarak herhangi bir örneklemin sınıflandırılması için belirlenen mesafe fonksiyonu kullanılarak sınıflandırılacak örneklemin eğitim setinde olan verilere olan uzaklığı hesaplanmaktadır. Hesaplama sonrasında belirlenen k adet örnek, eğitim kümesinden seçilmektedir. Seçilen bu yeni veri seti sınıflandırma veri seti olarak adlandırılmaktadır. Son olarak, hangi sınıfa ait olduğu tahmin edilmeye çalışılan örneklemin sınıfı, sınıflandırma veri setinde en çok bulunan sınıf olarak atanarak sınıflandırma işlemi tamamlanmaktadır (Harrington, 2012).

K-en yakın komşu algoritmasının uygulanmasında kullanılan mesafe ölçütleri veri setinde bulunan özniteliklerin kategorik, nümerik ve ikili değerlere sahip olması veya bunların hepsinden oluşması gibi durumlara göre değişebilmekte olup, *Öklid* ve *Manhattan* ölçütleri bahse konu algoritmada en sık kullanılmakta olan ölçütlerdir. Yukarıda bahsedilen k değerinin seçilmesi yüksek düzeyde doğruluk elde edebilmek için oldukça önemli olmasının yanı sıra problemli bir konudur. Literatürde bu işlem için çapraz doğrulama yapılması yani aynı veri seti için farklı k değerlerinin denenmesi şeklinde bir uygulama yapılması tavsiye edilmektedir. K değerinin seçilmesinin önemi aşağıda şekil 2.3'de gösterilmektedir. Öyle ki, söz konusu şekilde yeni gelen bir şeklin daire veya kare daire olarak sınıflandırılması konusunda k değerinin 1, 5 veya 10 olarak seçilmesi durumunda sonucun farklılaştığı görülmektedir.



Şekil 2.3 KNN Algoritmasında k Değerinin Seçimi.

Ayrıca, anlaşılması son derece basit olan k-en yakın komşu algoritmasında, tahminde bulunulabilmesine ilişkin hesaplamanın tamamı eğitim yerine test sırasında yapıldığından daha fazla hesaplama süresi söz konusu olabilmektedir. Ayrıca, sınıflar arasında dengesizlik bulunması, örneğin bir sınıfın diğer sınıflara oranla eğitim setinde daha fazla olması, durumunda bu sınıfa eğilimin olmaması için normalleştirme işlemin uygulanması gibi işlemler gerekebilmektedir.

2.2.4. Karar Ağacı Algoritması

Karar ağaçları, veri yapılarında kullanılmakta olan ağaç yapısı ile benzer şekilde kök, düğümler ve yapraklardan oluşmaktadır. Bu yapıdaki karar ağacı algoritmasında

amaç özniteliklerden çıkarılan karar kurallarının öğrenilmesi ile hedef değişkenin değerini tahmin edebilen bir modelin geliştirilmesidir. Bu amaç doğrultusunda bağımlı ve bağımsız değişkenler bir dizi test sorusu ve koşulun düzenlenmekte olduğu ağaç şeklinde bir yapıda temsil edilmektedir. Söz konusu ağaç yapısında, yaprak düğümleri test edilecek örneklerin dâhil edileceği sınıf etiketlerini, yaprak olmayan düğümler yani ağacın en üstünde yer alan ve kök düğüm olarak adlandırılmakta olan düğüm de dâhil olmak üzere karar düğümleri öznitelikler üzerinde uygulanan test işlemini ve test işlemi sonucunda hareket edilecek yönleri, her bir dal ise test işleminin sonucunu yani sınıf etiketlerini belirtmektedir.

Farklı öznitelik değerlerine sahip düğümleri ayırmak için öznitelik test koşullarını içermekte olan karar ağacı algoritmalarında, girdiler kök düğümden başlamak üzere girilmekte ve test edilen örneklemin öznitelik değerlerine ağaç yapısında belirlenmiş sorular sorulmak suretiyle alınan cevaplar doğrultusunda uygun dallar takip edilerek aşağı doğru ilerlenmektedir. Bu işleme yaprak düğüme ulaşılana kadar sürekli bir şekilde devam edilmekte olup, yaprak düğüme ulaşıldığında, test edilen örneklem söz konusu düğümde temsil edilmekte olan sınıf etiketine atanarak sınıflandırma işlemi sonlandırılmaktadır. Belirtilen şekilde ilerlemekte olan karar ağaçları kolayca görselleştirilebildiğinden alınan kararlar etkin bir şekilde izlenebilmekte ve anlaşılabilmektedir. Bu sebeple, karar ağacı algoritmaları sınıflandırma problemlerinde yaygın bir şekilde kullanılmaktadır.

Karar ağacı algoritması, ayrık olan ve olmayan verileri kullanabilmektedir. Bununla birlikte, veri setinde bulunan sınıf sayısına oranla eğitim veri setinin küçük olması yani dallara kıyasla çok fazla yaprak düğüm olması durumunda sınıflandırma hatasının arttığı gözlemlenmektedir. Ayrıca, söz konusu algoritma ezber yapmak suretiyle aşırı öğrenme problemine de eğilimlidir. Ezber yapılmasının önüne geçilmesi ve karar ağaçlarının çok karmaşık yapıda olmasının önlenmesi amacıyla budama işlemi yapılabilmekte olup, bu işlemde basitçe alt dalın yerine yaprak getirilmektedir.

Karar ağacı algoritmaları ile ilgili en önemli hususlardan birisi ağacın nasıl oluşturulacağına karar verilmesidir. Bu noktada, esnek ve doğruluk oranı yüksek bir karar ağacı elde etmek amacıyla çeşitli yaklaşımlar bulunmakta olup, bu yaklaşımlarda genel olarak ağacın aşamalı olarak gözlemlenmesi söz konusudur. Söz konusu yaklaşımların uygulanması konusunda, *entropi* ve bilgi kazanımı ile ID3 algoritması ön plana çıkmaktadır. Genel olarak ID3 algoritmasında kökten aşağı doğru hareket ederek sonuca en yakın durumun belirlenmesi tekniğini kullanmaktadır. *Entropi*, beklenmeyen durumların ortaya çıkma ihtimali üzerinde durmakta olup, eğitim seti homojen ise 0, eğitim setindeki değerler birbirine eşitse 1 değerini almaktadır (Kumar ve ark., 2019). Herhangi bir öznitelik için hesaplanan *entropi* değerinin küçük olması söz konusu özniteliğin ID3 algoritması için önemli bir öznitelik olduğunu ifade etmektedir. Diğer taraftan, bilgi kazanımı ise *entropinin* tam tersini ifade etmekte olup, karar ağaçlarının oluşturulması sırasında en yüksek bilgi kazanımı değerine sahip öznitelikler seçilebilmektedir. Ayrıca, tümevarım mantığıyla hareket etmekte olan karar ağacı algoritmaları ile özniteliklerin önem derecisi hakkında da bilgi sahibi olunabilmesi mümkündür.

2.2.5. Rastgele Orman Algoritması

Yukarıda bölüm 2.2.4.'de, karar ağacı algoritması ile ilgili en büyük problemlerden bir tanesinin söz konusu algoritma tarafından ezber yapılması olduğu belirtilmişti. Basitçe karar ağaçları koleksiyonu olarak ifade edilebilecek rastgele orman algoritmasında ise bahsedilen aşırı öğrenme probleminin önüne geçilmesi amaçlanmaktadır. Bu amaç doğrultusunda, hem veri setinden hem de özniteliklerden rastgele olacak şekilde alt gruplar seçilerek çok sayıda karar ağacı oluşturulmaktadır. Oluşturulan bu karar ağaçları orman olarak ifade edilen yapıda birleştirilmektedir. Diğer bir deyişle, bahse konu karar ağaçlarının oluşturulmasında rastgele seçilen öznitelikler kullanılırken, eğitilmesinde orijinal veri setinden yer değiştirme yapılarak elde edilen alt veri setleri kullanılmaktadır. Dolayısıyla, rastgele orman algoritmasında, karar ağacında olduğu gibi bütün özniteliklerin arasından en iyi dalı kullanarak düğümlerin dallara ayrılması ile tek bir ağaç elde edilmesi yerine rastgele

seçilen öznitelikler arasından seçim yapılarak düğümlerin dallara ayrılması ile istenilen sayıda ağaç oluşturulabilmesi söz konusudur. Birbirinden farklı ağaçlardan oluşan orman içerisindeki bu ağaçlarda ise karar ağaçları algoritmalarında olduğu gibi bir budama işlemi yapılmamaktadır ki bu durum algoritmanın doğruluk oranının artmasına katkı sağlamaktadır. Budama işlemi yapılmamasının sebebi ise yukarıda belirtilen hususlardan da anlaşılabileceği üzere bu algoritmada farklı alt grup eğitim setleri üzerinden öğrenme gerçekleştirilmekte olduğundan zaten aşırı öğrenme probleminin azaltılmakta olmasıdır. Diğer taraftan, bu yapının olumsuz yani ise hesaplama karmaşıklığının artmasına ek olarak karar ağaçlarında olduğu gibi oluşturulan ağaçlara ilişkin yorumlanabilecek tek bir çıktının verilememesidir.

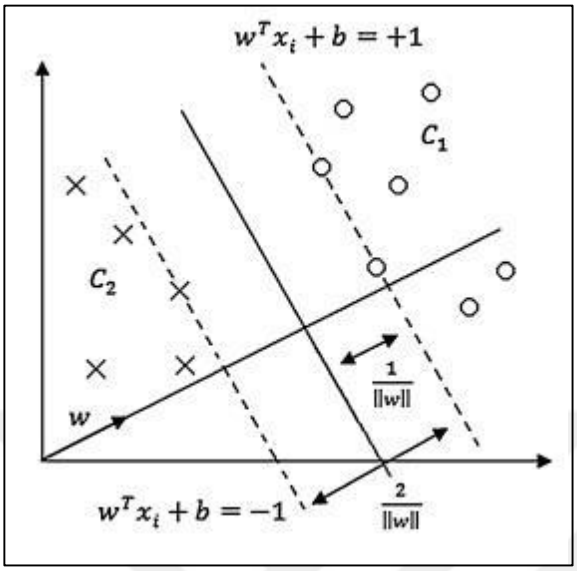
Örneklem üzerinden tahminde bulunulması aşamasında ise oluşturulan her bir karar ağacı bağımsız olarak tahminde bulunmakta olup, en çok tahminde bulunulan yani çoğunluğun oyuna sahip sınıf örneklemin sınıfı olarak belirlenmektedir.

Birçok zayıf tahmin edicilerin bir araya gelerek güçlü bir tahmin edici oluşturabileceği felsefesini temel alan topluluk yöntemlerinden biri olan rastgele orman algoritmasının bir diğer özelliği ise özniteliklerin önemine ilişkin bilgiyi sağlamasıdır ki bu husus tezin öznitelik seçimi ile ilgili kısmında da ele alınmaktadır.

2.2.6. Destek Vektör Makineleri

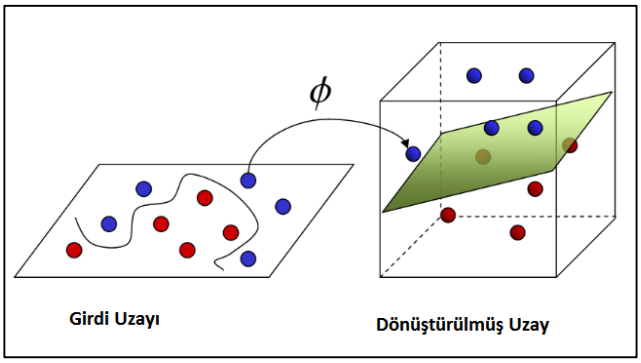
Destek vektör makineleri, regresyon ve sınıflandırma işlemlerinde kullanılabilmekte olup, genel olarak iki sınıfın üyelerinin ayırt edilebilmesi için kullanılabilecek en iyi sınıflandırma fonksiyonunu bulmayı amaçlamaktadır. Başka bir deyişle, destek vektör makineleri marjini en yüksek doğruyu seçerek sınıflandırma hatasını en aza indirirken iki sınıfı ayırabilmek için öznitelikleri kullanarak sınıfları en uygun şekilde ayırabilecek bir düzlem veya hiper düzlem bulmaya çalışmaktadır. Örnek olarak aşağıdaki şekil 2.4'de de görülebileceği üzere iki sınıf için bahse konu algoritmanın uygulanması amacıyla öncelikle hatayı minimize eden paralel iki adet vektör seçilerek aralarındaki mesafe maksimum yapılmakta, sonrasında ise söz konusu

iki vektörün orta notasından optimum hiper düzleme karar verilmektedir. Herhangi bir x verisinin sınıflandırılması ise y=wx+b hesaplanarak yapılmaktadır.



Şekil 2.4 SVM Algoritmasında Hiper Düzlemin Belirlenmesi (Alpaydin, 2004).

Bu yöntem genel olarak verilerin doğrusal olarak ayrışabildiği durumlarda kullanılmakta birlikte verilerin *sigmoid*, *polinomiyal*, doğrusal ve *radyal* tabanlı gibi çekirdek fonksiyonları aracılıyla doğrusal olarak ayrışabilecek duruma getirilmesinin mümkün olması sebebiyle doğrusal olarak ayrışamayan verilerin bulunması durumunda da uygulanabilmektedir (Alpaydin, 2004). Yani, çekirdek fonksiyonları kullanılarak doğrusal olmayan veri setleri için daha büyük bir boyuta taşıma işlemi yapılmaktadır. Örnek olarak aşağıda şekil 2.5'de verilerin 2 boyutlu uzaydan 3 boyutlu uzaya haritalanması durumu gösterilmekte olup, söz konusu şekilden de görülebileceği üzere veriler doğrusal olarak ayrılabilir duruma gelmiştir (Filiz, 2019).



Şekil 2.5 SVM Algoritmasında Verilerin Haritalanması (Moreira, 2011).

Vektör makinelerinin yukarıda yer alan tanımı ile diğer açıklayıcı ifadelere bakıldığında özellikle iki sınıfa ayırma işleminden bahsedilmektedir. Ancak bu durum destek vektör makinelerinin ikiden daha fazla sınıf bulunan uygulamalar için kullanılmasının mümkün olmadığı anlamına gelmemektedir. Dolayısıyla, destek vektör makineleri çoklu sınıfa sahip problemler için de uygulanabilmektedir. Bu noktada, sınıfların ayrılması için bire karşı hepsi yaklaşımıyla hareket edilebilmektedir. Şöyle ki, öncelikle bir sınıf etiketi seçilerek bu sınıfa 1, diğer sınıflara -1 etiketi verilmekte yani bir sınıf ile diğer tüm sınıfların karşılaştırması yapılmaktadır. Bahsedilen bu işlem, tüm sınıflar için ayrı ayrı uygulanmakta ve sınıf sayısına eşit sayıda sınıflandırıcı elde edilmektedir. Başka bir deyişle, sınıf sayısı kadar destek vektör makinesi eğitilmekte, tahmin aşamasında ise her bir destek vektör makinesinin çıktısı karşılaştırılarak en güvenli sonuç tercih edilmektedir.

Önsel bilgiye ihtiyacı olmadan yüksek genelleme performansı ile çalışabilmesinden dolayı destek vektör makineleri çok çeşitli alanda uygulanabilmektedir. Ancak destek vektör makinelerinin hesaplama maliyeti açısından verimsiz olduğunun da dikkate alınması gerekmektedir.

2.2.7. Yapay Sinir Ağları

İnsan beyninden ilham almakta olan ve evrensel fonksiyon tahmin edici olarak bilenen yapay sinir ağları hem doğrusal hem de doğrusal olmayan fonksiyonları

haritalayabilmektedir. Bir başka deyişle, yapay sinir ağları aralarında yeteri kadar gizli katman bulunması şartıyla bir x girişi ve y çıkışı arasındaki ilişkiyi temsil eden herhangi bir fonksiyonunun tahmin edilmesi amacıyla kullanılmaktadır (Lecun ve ark., 2015). Dolayısıyla sınıflandırma problemlerinde de kolaylıkla uygulanabilmektedir.

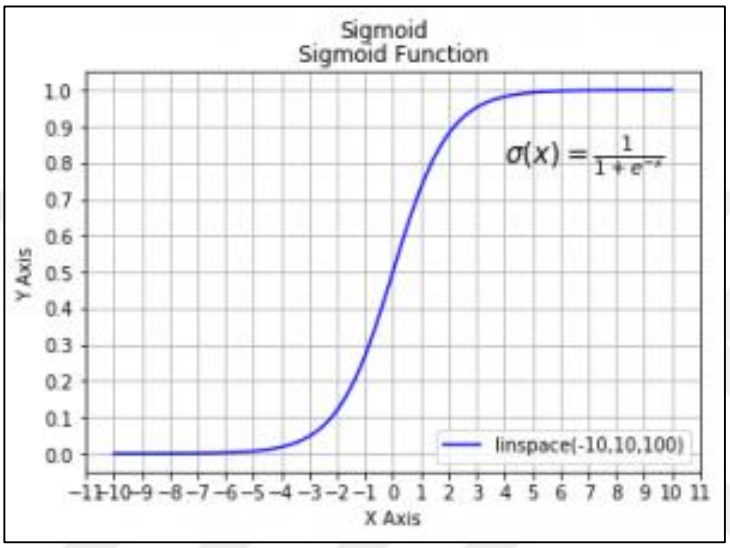
Yapay sinir ağları biyolojik nöronlardan esinlenen ve yapay nöron adı verilen hesaplama birimleri ile bu nöronlar arasındaki bağlantılardan oluşmaktadır. Ağda bulunan her bir nöron bir çıktı değeri üretmek üzere birkaç girdi değeri almaktadır. Yapay sinir ağlarında, nöronların her bir girdisi yani bağlantılar ağırlık olarak adlandırılan katsayılara sahiptirler (Lecun ve ark., 2015). Bu katsayılar, girdilerin önyargı değeri ile birlikte çıktıyı nasıl etkilediğini göstermektedir. Böylece, girdilerden çıktının elde edilmesinde kullanılan formül aşağıda şekil 2.6'da verilmekte olup, söz konusu formülde x girdileri, w girdilerin ağırlıklarını, b ise önyargı değerini temsil etmektedir. Söz konusu formülde, önyargı değeri ve ağırlıklar kullanılarak girdilerin ağırlıklı toplamlarının hesaplanması yapılmaktadır. Önyargı değeri, girdilerden bağımsız olduğundan formül ile temsil edilen doğrunun ötelenmesi için kullanılmakta olan bir parametredir ve bir yapay sinir ağındaki sayısı nöron sayısına eşittir (Rosenblatt, 1958).

$$y = \sigma(\sum_{i=0}^{n}(w_i * x_i + b)$$

Şekil 2.6 ANN Çıktı Elde Etme Formülü.

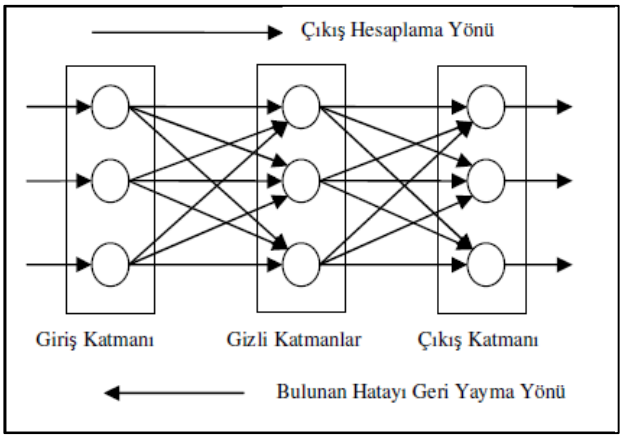
Yukarıda belirtilen formül ile ağırlıklı toplam hesaplandıktan sonra, nöronun çıktısını hesaplamak için bir doğrusal olmayan bir aktivasyon fonksiyonu σ kullanılmaktadır. Bu aktivasyon fonksiyonu ağırlıklı toplamı doğrusal olmayan hale getirmektedir. Aktivasyon fonksiyonun doğrusal olmaması ise ağın doğrusal olmayan karmaşık yapıdaki görevleri öğrenilebilmesi açısından oldukça önemlidir. Yapay sinir ağlarında genellikle kullanılmakta olan aktivasyon fonksiyonları; *sigmoid*, hiperbolik *tanjant* (tanh) ve *Rectified Linear Unit* (ReLU) fonksiyonlarıdır. Söz konusu fonksiyonlardan en yaygın kullanılanı olan *sigmoid* fonksiyonuna ilişkin grafik aşağıda şekil 2.7'de verilmekte olup, şekilden de anlaşılabileceği üzere bu fonksiyon

aldığı girdileri 0 ve 1 aralığında gerçek bir sayıya dönüştürmektedir. Dolayısıyla olasılık hesaplanması gibi yalnızca pozitif sonuç vermesi gereken yapay sinir ağlarında kolaylıkla kullanılabilmektedir. Hiperbolik t*anjant* fonksiyonu *sigmoid* fonksiyonuna benzemekle birlikte bu fonksiyonun çıktıları -1 ile 1 arasında olmaktadır. RELU fonksiyonu ise pozitif girişler için girdi değerini, negatif girişler içinse 0 değerini çıktıya atamaktadır.



Şekil 2.7 Sigmoid Fonksiyonu Grafiği (Makinist, 2018).

Aşağıda yer alan şekil 2.8'de giriş ve çıkış katmanları ile bir adet gizli katmana sahip olan basit bir yapay sinir ağı modeli gösterilmektedir. Yapay sinir ağlarında tek bir gizli katman bulunmasına yönelik kısıtlama bulunmamakta olup, tasarlanan mimariye bağlı olarak gizli katmanların sayısının artırılması mümkündür. Bu durumda, katman sayısı artıkça yapay sinir ağlarının karmaşıklığının artacak olmasından dolayı eğitim süresinin uzayacağı hususu göz önünde bulundurulmalıdır.



Şekil 2.8 Yapay Sinir Ağı Modeli (Dandil ve Cevik, 2012).

Yapay sinir ağlarının eğitilmesi sürecinde ise nöronların yukarıda belirtilen ağırlıkları ayarlanmaktadır. Yani eğitim sürecinde örnek veriler kullanılarak gerçek çıktılara en yakın sonucu verecek ağırlık değerleri bulunmaya çalışılmaktadır. Eğitim işleminin sonucunda elde edilen ve girdiler ile çıktı arasındaki y = f (x) ilişkisini tanımlayabilen fonksiyonun eğitim veri setinde bulunmayan verilere genelleştirilebilmesi önem arz etmektedir. Bu doğrultuda, bir yapay sinir ağının belirli bir girdi için doğru çıktıyı tahmin etme yeteneğinin etkin bir şekilde ölçülmesi gerekmekte olup, bu işlem kayıp fonksiyonu adı verilen fonksiyonlarla yapılmaktadır. Kayıp fonksiyonu, en basit tabiriyle gerçek çıktı ile ağ tarafından tahmin edilen çıktı arasındaki farkı hesaplamaktadır. Yaygın olarak kullanılmakta olan kayıp fonksiyonlarına bakıldığında ise bir gerçek ve tahmini değer arasındaki farkın karesini hesaplayan ortalama kare hatası fonksiyonunun öne çıkmakta olduğu görülmektedir.

Bu çerçevede, bir sinir ağının eğitilmesinin amacının, nöronların ağırlık ve önyargı değerlerini ayarlayarak seçilen kayıp fonksiyonunun değerini en aza indirmek olduğu söylenebilecektir. Eğitim işleminin temelinde ise çoğunlukla geri yayılım algoritması yer almaktadır. Geri yayılım algoritmasının kullanıldığı bir eğitim işlemi ağırlıkların rastgele bir değere atanması ile başlanmaktadır. Sonrasında ise kayıp fonksiyonunun değeri minimize edilene kadar ağda ileriye ve geriye doğru hareket etme işlemleri tekrarlanmaktadır. İleriye doğru hareket etme aşamasında ağın çıktı değeri ve kayıp fonksiyonunun değeri hesaplanmaktadır. Geriye doğru hareket etme aşamasında ise kayıp fonksiyonun değerinin azaltılması için ağırlıkların ne şekilde

ayarlanması gerektiği *gradyan* iniş algoritması ile bulunmaktadır. Bu işlem doğrultusunda ağırlıklar yapay sinir ağının öğrenme oranı parametresi çerçevesinde küçük miktarda güncellenmektedir. Kayıp fonksiyonun değerinin azaltılması için ağırlıkların nasıl güncelleneceğini belirlemede kullanılan *gradyan* iniş algoritmasında, *gradyan,* bir ağırlık parametresi güncellendiğinde kayıp fonksiyonunun değerinde meydana gelecek değişiklik olarak tanımlanmaktadır. Bahse konu algoritma, zincir kuralını kullanarak her ağ parametresinin *gradyanını* hesaplamakta ve kayıp fonksiyonunun değerini ne şekilde azaltacağına karar vermek için *gradyan* yönünü kullanmaktadır. Ancak belirtmek gerekir ki, büyük veri setleri için her bir örneğin *gradyan* hesabı yapılması zaman alıcı olacaktır. Bu sebeple, süreci hızlandırmak için *stokastik gradyan* iniş algoritması da kullanılabilmektedir. Bu algoritmada ise girdi seti yığınlara ayrılarak yığınlar için *gradyan* hesaplaması yapılmaktadır. *Stokastik gradyan* iniş algoritması, *Adam, Adagrad, RMSprop* gibi varyantlarıyla birlikte eğitim için yaygın olarak kullanılan algoritmalardır (Ruder ve Plank, 2018).

Yukarıda adımları açıklanan kayıp fonksiyonunun değerinin küçültülmesi doğrultusundaki eğitim işlemine ilgili kayıp fonksiyonunun değeri artık azaltılamaz olana kadar devam edilmekte ve bu noktaya ulaşıldığında eğitim işlemi tamamlanmaktadır. Böylece, oluşturulan yapay sinir ağı yeni örneklemler için kullanılabilir hale gelmektedir.

2.3. Veri Ön İşleme Süreci

Sınıflandırma algoritmalarının uygulanmasında, kullanılacak veri setinin yapısı oldukça önemli olmaktadır. Bu sebeple, söz konusu algoritmaların uygulanmasına geçilmeden önce genellikle veri seti üzerinde bazı ön işlemler gerçekleştirilmekte olup, söz konusu ön işlemler ile veri setinin hazırlanması makine öğrenmesi sürecinde çok fazla zaman gerektiren aşamalardan biridir (Miksovsky ve ark., 2002). Bu ön işlemlerin amacı algoritmaların etkin bir şekilde uygulanabilmesi için ham veri setinin daha anlamlı, odaklı, okunabilir ve yorumlanabilir bir hale getirilerek kalitesinin artırılmasıdır. Söz konusu veri ön işleme sürecinde veri özetleme, veri indirgeme,

eksik değer doldurma ve veri dönüştürme gibi teknikler uygulanabilmektedir. Ancak bu süreçte uygulanacak işlemlere yönelik olarak belirlenmiş herhangi bir standart bulunmamakta olup, genellikle uygulanacak işlemler kullanılacak ham veri setine bağlı olarak değişiklik göstermektedir. Bu tez çalışmasında kullanılan veri ön işleme tekniklerine ise bu bölümde değinilmektedir.

2.3.1. Veri Anlama ve Temizleme

Makine öğrenmesi algoritmaları kullanılarak bir makine öğrenmesi modelinin oluşturulmasından önce, özellikle çok boyutlu ve karmaşık yapıdaki veri setleri açısından veri setinin iyi anlaşılması önem arz etmektedir. Çünkü verilerin analiz edilmesiyle veri seti hakkındaki önemli bilgiler elde edilebilmekte ve ön işleme sürecinde hangi yöntemlerin uygulanacağına karar verilmesi mümkün olmaktadır. Dolayısıyla, bu aşamada belirtilen amaç doğrultusunda veri seti hakkında fikir edinilebilmesi için istatistiksel hesaplamalar yapılabilmekte ve öznitelikler detaylı bir şekilde incelenebilmektedir.

Ayrıca, makine algoritması ile model geliştirilmesinin temelinde, kullanıcıların elde etmek istedikleri bilgilerin veya tahminlerin veri seti üzerinden gerçekleştirilmesi bulunduğundan kullanıcı hedeflerinin ve gereksinimlerinin veri seti üzerinden makine öğrenmesi hedefi haline getirilmesi gerekmektedir. Bu gerekliliğin yerine getirilebilmesi içinse veri setinin anlaşılması sonrasında hedeflenen amaca uygun olarak yanlış veya boş bilgi içeren özniteliklerin elimine edilmesi, geçersiz veri tipine sahip olan özniteliklerin kaldırılması, birbiri ile bağlantılı olduğu anlaşılan özniteliklerden hangisinin kullanılacağına karar verilmesi ve yararlı olmayacak örneklemlerin çıkarılması gibi veri temizleme uygulamaları yapılmaktadır. Örneğin, herhangi bir terörist eylem gerçekleştiğinde sorumlu grubun tahmin edilmesini amaçlayan bir makine öğrenmesi modeli oluşturulmak istenildiğinde, eğitim veri setinde terörist grubu temsil eden özniteliğin boş olması model açısından anlamsız olacaktır.

Bu kapsamda, gerek veri setinin anlaşılması gerekse de özniteliklerin ve örneklemlerin amaca uygun hale getirilmesi makine öğrenmesi algoritmalarının başarısının ve performansının artırılmasında ve sürecin uygun bir şekilde tamamlanabilmesinde önemli rol oynamaktadır.

2.3.2. Eksik Değer Doldurma

Makine öğrenmesi modellerinde kullanılacak veri setinde bazı örneklemler için bazı özniteliklere ait değerlerin bulunmaması söz konusu olabilmektedir. Bu durumda bulunmayan öznitelik değerleri eksik değer olarak ifade edilmekte olup, söz konusu eksik değerler veya eksik değerlerin yanlış kullanılması nedeniyle birtakım problemler ortaya çıkabilmektedir. Örneğin, bahse konu eksik değerler hesaplamada zorluk yaratacağından dolayı algoritmaların performansı önemli ölçüde etkileyebilmektedir (Suthar ve ark., 2012). Ayrıca, eksik değerler içeren veri setinde önyargı problemi olabileceğinden ilgili algoritmaların doğruluk değerlerinin istenilen düzeyde olmaması sonucu da ortaya çıkabilecektir. Bahse konu eksik değerlerin uygun değerlere atanması yani doldurulması önem arz etmekte olup, bu işlem için geliştirilen bazı yöntemler ise aşağıda verilmektedir (Maniraj ve ark., 2019).

- Hedef özniteliğin eksik olduğu örneklem veri setinden çıkarılabilmektedir.
- Çok fazla eksik değer içeren öznitelikler veri setinden çıkarılabilmektedir.
- Kayıp değerlerin tamamı sabit bir değere veya veri setinde özniteliğin alabileceği “bilinmiyor”, “diğer” gibi tanımlı değerlerden en uygun olanına ataması yapılabilmektedir.
- Kayıp değerin bulunduğu özniteliğin ortalama değeri kayıp değere atanabilmektedir.
- Kayıp değerin bulunduğu kategorik özniteliklerde en sık tekrar eden değer kayıp değere atanabilmektedir.
- Kayıp değerlerin makine öğrenmesi algoritmaları ile tahmin edilmesi sonucunda bulunan değerlerin kayıp değerlere atamasının yapılması.

2.3.3. Öznitelik Ölçekleme ve Normalleştirme

Bazı durumlarda, veri setinde bulunan nümerik özniteliklerin aldığı değerler arasında büyük farklar bulunması söz konusu olmaktadır. Veri setinde bulabilecek bu şekildeki bir dengesizlik ise büyük değere sahip özniteliklerin hedef değişken üzerinde daha fazla etkisinin olması sonucunu doğrulabilmektedir. Dolayısıyla, ham veri setinde bulunan özniteliklerin değerleri arasında büyük farklar bulunması durumunda bazı makine öğrenme algoritmalarının etkin bir şekilde çalışması mümkün olmamaktadır. Bu noktada ise veri setinde bulunan özniteliklerin aralığının standartlaştırılması yöntemi olan ölçeklendirme işlemi, bahse konu problemle mücadele etmek adına ön plana çıkmaktadır. Makine öğrenmesi alanında genellikle iki tür ölçeklendirme yöntemi kullanılmakta olup, bunlara ilişkin açıklamalar aşağıda sunulmaktadır (Kartal, 2015).

Standardizasyon: Bir veya daha fazla özniteliğin, ortalama değerleri 0 ve standart sapma değerleri 1 olacak şekilde ölçeklendirilmesi işlemidir. Standardizasyon işleminde verilerin bir *Gauss* dağılımına sahip olduğu varsayılmaktadır. Verinin *Gauss* dağılımına sahip olması yönünde bir zorunluluk bulunmamakla birlikte söz konusu dağılıma sahip verilerde standardizasyon işlemi daha etkili olmaktadır.

Normalizasyon: Minimum-maksimum normalizasyonu olarak da adlandırılmakta olup, bir veya daha fazla özniteliğin 0 ila 1 aralığına yeniden ölçeklendirilmesi işlemidir. Yani, normalizasyon işleminden sonra ilgili özniteliğin içerdiği verilerin maksimum değeri 1, minimum değeri ise 0 olacaktır. Normalizasyon işleminde normal dağılım esas alınmakta olup, yeni değerler her bir eski değerin ortalamaya olan uzaklığının standart sapmaya oranı ile elde edilmektedir. Dolayısıyla, özniteliklerin farklı ortalama ve standart sapmaya sahip olabilmeleri mümkündür.

2.3.4. Kategorik Değer Dönüştürme

Makine öğrenmesi algoritmaları yalnızca sayısal vektörlerin işlenmesi için uygundur. Dolayısıyla, söz koz konusu algoritmalar kullanılarak model geliştirilebilmesi için veri türü nümerik olmayan özniteliklerin veri türlerinin nümerik değerlere dönüştürülmesi gerekmektedir. Bu doğrultuda, özniteliklerin makine öğrenmesi algoritmaları için uygun bir forma dönüştürülebilmesi amacıyla "*label encoder*" ve "*one hot encoder*" kullanılmaktadır.

"*Label encoder*", veri setinde yer alana kategorik öznitelikleri doğrudan bir sayısal değere atayarak nümerik değere dönüştürme işlemini yapmaktadır. Yani, veri setindeki özniteliklerin her bir farklı değeri nümerik bir değer ile temsil edilmektedir. Aşağıda çizelge 2.3'de terör gruplarının "*label encoder*" kullanılarak nümerik değere dönüştürülmüş hali gösterilmektedir (Karabay, 2019).

Çizelge 2.3 Label Encoding Uygulanması.

Kategorik Değişken	**Nümerik Karşığı**
Grevciler	1
Öğrenci Radikalleri	2
İsyancılar	3

Diğer taraftan, kategorik özniteliklerin aldıkları değerler arasından sıralı bir ilişki bulunmaması durumunda nümerik değere dönüştürme işleminde verilerin doğrudan sayısal bir değere atanması ilgili değerlerin makine öğrenmesi algoritmaları tarafından yanlış yorumlanmasına sebebiyet verecektir. Bu noktada ise kategorik değişkenlerin nümerik dönüşümü "*one hot encoding*" kullanılarak yapılmaktadır. Bu yöntemde, kategorik değişkenlerin ikili olarak temsil edilmesi yapılmaktadır. Yani, bir öznitelik alanında m farklı değere sahip n örneklem bulunduğu varsayıldığında m farklı öznitelik değerinin temsil edilmesi için m-bit ikili kodlayıcı kullanılmakta ve her bir farklı değer için yalnızca bir bit aktif olarak işaretlenmektedir. Sonuç olarak da aşağıda çizelge 2.4'de gösterildiği gibi n*m matrisi elde edilmektedir.

Çizelge 2.4 One Hot Encoding Uygulanması.

	Grevciler	Öğrenci Radikalleri	İsyancılar
Örnek 1	1	0	0
Örnek 2	0	1	0
Örnek 3	0	0	1
Örnek 4	0	0	1
Örnek 5	0	1	0
Örnek 6	1	0	0

2.4. Veri Sızıntısı

Makine öğrenmesi alanındaki en sinsi problemlerden biri hedefe yani tahmin edilmesi planlanan alana ilişkin veri sızıntısının olmasıdır. Veri sızıntısı, en geniş tabiriyle modelde yer alması uygun olmayan özniteliklerin modele dahil edilmesi nedeniyle modelin performansının gerçek dünya uygulamalarında pek mümkün olmayacak bir seviyeye gelmesini ifade etmektedir. Veri sızıntısı daha dar bir şekilde ifade edilebilecek olmakla birlikte konuya geniş bir çerçeveden bakılması önemli konuların tartışılması ve değerlendirilmesine olanak sağlamaktadır. Örneğin, veri sızıntısının, gerçek bir olayın tahmin edilmesi sırasında mevcut olmayacak özniteliklerin modele dahil edilmesi ile test sonuçlarının olması gerekenden daha iyi görünmesi olarak dar bir şekilde ifade edilmesi de mümkündür. Herhangi bir özniteliğin model için uygun olup olmaması ise hem modelin kapsamına hem de performansına bağlıdır.

Hedefe ilişkinin veri sızıntısının en basit örneği, hedefi tahmin edebilmek için doğrudan hedefin kendisinin kullanılmasıdır. Örneğin, herhangi birinin yaşının tahmin edilmesi amacıyla basit bir model oluşturulduğu varsayıldığında doğum yılının modelin giriş veri setinde yer alması modelin etkinliği ve geçerliliğini ortadan kaldıracaktır. Ancak veri sızıntısının fark edilmesi her durumda bu kadar basit olmayabilir. Bu doğrultuda, veri sızıntısı oluşturabilecek bazı durumlar aşağıda açıklanmaktadır (Brownlee, 2021).

1) Modelin, tahmin esnasında modelde bulunmayacak bir veya birkaç öznitelik içermesi durumunda veri sızıntısı ortaya çıkacaktır. Örneğin, hastaneden taburcu olacak bir hastanın tekrar hastaneye gelip gelmemesine dair bir model oluştururken bir sonraki ziyaretindeki kan şekerinin girdi veri setine dahi edilmesi bahse konu veri sızıntısını ortaya çıkaracaktır. Çünkü hastaneye ikinci defa gelmeyecek bir hasta için böyle bir veri mevcut olmayacaktır.

2) Zaman değişkenleri, hedef değişkenin tahmin edilmesinde önemli bir öngörücüdür. Ancak, modelin değerlendirilmesinde eğitim setindeki durumlardan önce meydana gelen durumların kullanılması durumunda veri sızıntısı oluşacak olup, bu tür bir veri sızıntısı modeli oluşturmak için kullanılan tüm değişkenler tahmin anında da mevcut olduğundan özellikle daha sinsidir. Yani, bu durumda zaman değişkenleri sayesinde modelin, hedef değişkendeki daha büyük varyansın veya yalnızca gelecekte göze çarpan bir durumunun farkında olmasıyla geleceği tahmin etmesi söz konusu olmaktadır. Örneğin, belirli faktörlere bağlı olarak kişilerin egzersiz davranışlarının tahmin edilmesi için bir model geliştirilmek istenildiği ve elde 28 Şubatta sona eren günlük egzersiz verilerinin bulunduğu varsayıldığında, geleneksel yaklaşımla geliştirilen modelde yeni yılda alınan kararlar ve belirlenen hedefler doğrultusunda egzersiz davranışlarının 1 Ocak sonrasında artış eğiliminde olduğu kolaylıkla anlaşılabilecektir. Bu durum ise belirli faktörlere göre tahmin yapılması olayının önüne geçilerek zamana bağlı bir tahmin yapılması sonucunu ortaya çıkaracaktır. Esas itibariyle benzer durum belirli dönemlerde aktif olan terör grupları açısından da söz konusudur.

3) Modelin, tahmin anında mevcut olsa bile gerçek dünyada bazı durumlarda oluşmayan bir veya birkaç özniteliği içermesi durumunda da veri sızıntısı oluşacaktır. Örneğin bir kanser hastasının hastalık düzeyinin tahmin edilmesi amacıyla model oluşturulduğu varsayıldığında girdi veri setinde alınan kemoterapi sayısının olması problem olabilecektir. Çünkü ilk defa hastaneye gelen ve esas itibariyle modelin tahmin yürütmesi beklenilen hastalar için bu veri elde olmayacaktır. Dolayısıyla, bu durum bahse konu özniteliğe bağlı olarak tahminlerde eğilim bulunması sonucunu ortaya çıkaracaktır.

4) Modelin, kullanım durumları dikkate alındığında anlam ifade etmeyen bir veya birkaç özniteliği içermesi de istenilen bir durum olmayacaktır. Bu tür veri sızıntısının ortaya çıkarılabilmesi için modelin kullanım amaçları ile problemin detaylı bir şekilde anlaşılması gerekmektedir.

5) Modelin, hedef değişkenle doğrudan etkileşime giren öznitelikleri içermesi durumunda da veri sızıntısı meydana gelecektir. Bu durumun anlaşılabilmesi için hedef değişken ile diğer öznitelikler arasındaki korelasyonun incelenmesi gerekmektedir. Bu durum esasen, bir özniteliğin gerçek dünya uygulamasından uzaklaşarak hedef değişkeni tahmin edebilmesi durumunda ortaya çıkmaktadır. Şöyle ki, terör gruplarının tahmin edilmesi amacıyla geliştirilen bir modelde, olayda kullanılan silah türü doğrudan sonucu veriyorsa yalnızca bu değişkene bağlı olarak bir model geliştirilmesi gerçek dünya uygulamasında elde bulunacak diğer verilerin önüne geçecektir ve modelin etkinliği azalacaktır.

6) Modelin oluşturulması esnasında, dış kaynaklardan toplanan hedef değişkene ilişkin bilgilerden yararlanılması da veri sızıntısı meydana getirecektir.

7) Eğitim veri setinde hedef değişkene ilişkin ilave bilgi bulunması durumu veri sızıntısına neden olacak olup, bu alandaki en temel veri sızıntısı örneklerindendir. Bu durum, hedef değişkenin doğrudan girdi veri setinde yer alması durumu ile oldukça benzerdir.

2.5. Öznitelik Seçimi

Makine öğrenmesi algoritmaları kullanıcının doğrudan müdahalesi olmadan veriden öğrenme yeteneğine sahiptir. Dolayısıyla, söz konusu algoritmaların performansı kullanılan veri setine oldukça bağlıdır. Veri setinde, bilgilerin öznitelikler kullanılarak temsil edildiği dikkate alındığında amaca uygun doğru özniteliklerin

belirlenmesi önem arz etmektedir. Bu noktada ise en basit anlamda veri setindeki özniteliklerden sınıflandırma sonucu üzerinde etkisi olabilecek olanlarına karar verilmesine ve diğer özniteliklerin elimine edilmesine imkan tanıyan öznitelik seçiminin üzerinde durulması gerekmektedir. Öznitelik seçimi ile tahmin edilmesi veya sınıflandırılması planlanan değişkene bağlı olarak ilgisiz özniteliklerin veri setinden atılması sonucunda veri setinin boyutunun küçültülmesi, algoritmaların çalışma sürelerinin minimize edilmesi, algoritmaların ezber yapmasının önüne geçilmesi ve doğruluk oranlarının artırılması amaçlanmaktadır. Öznitelik seçimi sonrasında, özellikle karmaşık veri setlerinde söz konusu amaçlara ulaşılmasına ek olarak veri setinin yönetimi daha kolay hale gelmekte, öznitelikler arasındaki ilişkiler kolayca tanımlanabilmekte ve modellerin anlaşılabilirliği artmaktadır. Öznitelik seçimine ilişkin olarak çeşitli algoritmalar ve yöntemler bulunmakta olup, bunlardan aşağıda bahsedilmektedir.

Öznitelik seçiminde kullanılan yöntemlerin; filtre yöntemleri, sarmalayıcı yöntemler ve gömülü yöntemler olmak üzere 3 ana gruba ayrılması mümkündür (Guyon ve Elisseeff, 2003). Filtre yönteminde, hedef değişken ile öznitelikler arasındaki bağlantı dikkate alınarak özniteliklerin önem derecesi hesaplanmakta ve bu önem derecesi dikkate alınarak da öznitelikler filtrelenmektedir. Sonuç olarak, filtreleme işlemi neticesinde yeni bir öznitelik grubu elde edilmektedir. Bu yöntemin, sınıflandırma algoritmalarından bağımsız olarak işlem yapabilmesi olumlu tarafı iken diğer yöntemlerle kıyaslandığında daha az başarılı olması sıklıkla karşılaşılan bir durumdur. Sarmalayıcı yöntemde, alt öznitelik grupları belirlenerek kullanılacak algoritma ile sınıflandırma işlemi yapılmaktadır. Alt öznitelik seçimi ve sınıflandırma işlemine en yüksek doğruluk oranı elde edilene kadar devam edilmekte ve en yüksek doğruluk oranının elde edildiği öznitelikler üzerinde karar kılınmaktadır. Bu yöntemde, kolay uygulanabilme ve başarı oranın yüksek olması olumlu taraf olarak öne çıkmaktayken işlem maliyetinin fazla olması, söz konusu yöntemin olumsuz tarafını oluşturmaktadır (John, 1997). Gömülü yöntemde ise model oluşturma aşamasında rastgele orman gibi bazı makine öğrenmesi algoritmalarının sağladığı özellikler kullanılarak öznitelik seçim işlemi gerçekleştirilmektedir. Bu yöntemde işlem maliyetinin az ve başarı oranının yüksek olması olumlu taraf iken

algoritmalardan ayrı kullanılmasının mümkün olmaması olumsuz taraftır (Guyon ve Elisseeff, 2003).

Bu çerçevede, yukarıda açıklanan bilgiler ışığında öznitelik seçimi ile ilgili olarak bazı yöntemler ve algoritmalar hakkındaki detaylara aşağıda yer verilmektedir (Filiz, 2019).

Korelasyon öznitelik seçim algoritması: Bu yöntemde, hedef değişken ile her bir öznitelik arasındaki *Pearson* korelasyon değeri ölçülmektedir. Elde edilen değerler göstergesinde de filtreleme işlemi gerçekleştirilmektedir.

Ki-kare testi algoritması: Yalnızca kategorik öznitelikler için uygulanabilir olan bu algoritmada, ilgili öznitelikler ile hedef değişken arasındaki bağlantı istatistiksel olarak hesaplanmakta olan p değerine göre değerlendirilmekte ve özniteliklerin veri setini tanımlayabilip tanımlayamadığına karar verilmektedir.

***CFS subset* algoritması:** Bu algoritmada değişkenler korelasyon yardımı ile değerlendirilmekte olup, özniteliklerin birbirleri arasında düşük korelasyon, hedef değişken ile de yüksek korelasyon değerine sahip olması aranmaktadır.

***One - r* algoritması:** Bu algoritmada, öznitelik seçim işlemi özniteliklerin hata oranına göre sıralanması ile yapılmaktadır. Hata oranı ise her bir özniteliğin tek başına kullanılması durumunda oluşacak değeri ifade etmektedir.

Bilgi kazancı algoritması: *Entropi*, bir sistemin düzensizliğini ifade eden 0 ila 1 arasındaki bir değerdir. Herhangi bir veri için söz konusu değerin 1'e yakın olması daha fazla bilgi içerdiğini ifade etmektedir. *Entropi* temelli bir algoritma olan bilgi kazancı algoritmasında, her bir özniteliğe ait bilgi kazancı değeri hesaplanmaktadır. Yani, bir öznitelik kullanılarak veri setinin ayrıştırılması yapıldığında hedef değişkene ait belirsizliğinin değişimi ölçülmektedir. Özniteliklerin bilgi kazancı değerinin daha

önceden belirlenen eşik değerden düşük olması durumunda da ilgili öznitelik veri setinden elimine edilmektedir.

Kazanım oranı algoritması: Bu algoritma, bilgi kazancı algoritmasının bir alternatifini oluşturmaktadır. Yukarıda açıklanan bilgi kazancı algoritmasında, farklı değerlere sahip özniteliklerin seçilme ihtimali yüksektir. Bu durumun azaltılması amacıyla kazanç oranı algoritması kullanılmakta olup, bu algoritmada değişkenlerin sayısı ve boyutu da dikkate alınmaktadır. Bu doğrultuda, bilgi kazancının bölünmüş bilgiye oranı ile yeni bir kazanç oranı elde edilmektedir.

***Gini* katsayısı algoritması:** Bu algoritmada, yukarıda açıklanan bilgi kazancı ve kazanım oranı algoritmalarındakine benzer şekilde her bir öznitelik için kazanım değeri hesaplamakta ve bu hesaba göre seçim işlemini tamamlamaktadır. Ancak bu yöntem söz konusu diğer iki yöntemden farklı olarak *entropi* değerini kullanmamakta, dolayısıyla diğer iki yöntemin alternatifini oluşturmaktadır.

***ReliefF* algoritması:** Bu algoritmada temel olarak özniteliklerin değerleri ile aralarında bağımlılık olup olmadığı bilgileri üzerinden öznitelik seçim işlemi yapılmaktadır. Bunun için ilk olarak bir örnek seçilmekte ve söz konusu örneğin kendi sınıflarındaki diğer örneklere yakınlığı ve uzaklığı dikkate alınarak öznitelik seçim işleminin yapılabileceği bir model oluşturulmaktadır.

İleriye doğru arama yöntemi: Bu yöntemde, boş bir öznitelik seti ile işlemlere başlanmakta ve doğruluk oranının artırılması gibi belirlenen bir performans kriterine göre özniteliklerin test edilerek seçilmesi işlemi yapılmaktadır.

Geriye doğru arama yöntemi: İleriye doğru arama yöntemine benzer olmakla birlikte ondan farklı olarak geriye doğru öznitelik eleme işlemi yapılmaktadır. Yani, veri setinde bulunan tüm özniteliklerle işleme başlanmakta ve belirlenen performans kriterlerine göre kötü performans gösteren öznitelikler elimine edilmektedir. Bu işleme en iyi performans gösteren veri seti elde edilene kadar devam edilmektedir.

Diğer taraftan, açıklanan bu yöntemlere ek olarak yürütülen çalışmanın amacı ve elde bulunan veri seti dikkate alınarak farklı yaklaşımların uygulanması da mümkündür. Bu tez çalışmasında da permütasyon ve sütun kaldırma yöntemleri ile rastgele orman algoritmasının birlikte kullanıldığı bir yaklaşım ile öznitelikler incelenmiştir. Bu nedenle, aşağı söz konusu hususlara ilişkin bilgilerden kısaca bahsedilmektedir.

Öncelikle belirtmek gerekir ki sınıflandırma işleminin sonuçlarını doğru bir şekilde tahmin edebilen bir model geliştirilmesi oldukça önemlidir. Ancak bu tek başına yeterli olmayacaktır çünkü aynı zamanda geliştirilen modelin yorumlanabilir olması da önem arz etmektedir. Dolayısıyla, öznitelikler arasından seçimde bulunulması ve bu amaç ile özniteliklerin önem derecelerinin incelenmesi makine öğrenmesi alanındaki çalışmaların önemli bir kısmını oluşturmaktadır. Bu noktada, rastgele orman algoritması ile ilgili uygulamalarda özniteliklerin önem dereceleri hakkında bilgi sağlanabilmektedir. Dolayısıyla, esasen birçok modelde uygulanabilir olan permütasyon yöntemi veya sütun kaldırma yöntemi ile özniteliklerin önem derecesinin incelenmesi tekniği rastgele orman algoritması ile kolayca uygulanabilecektir. Esas itibariyle, rastgele orman algoritmasında varsayılan önem derecelerine ilişkin bilgi sağlanmaktadır. Ancak, söz konusu algoritma tarafından özniteliklerin önem derecesinin hesaplanması sırasında sürekli değişkenlerin veya eleman sayısı yüksek olan kategorik değişkenlerin önem derecelerinin şişirilmesi yönünde bir eğilimin olduğu bilinmekte olup, bahsedilen eğilim nedeniyle söz konusu algoritma tarafından varsayılan olarak sağlanan önem derecesi bilgisinin bazı durumlarda doğruluğuna ve güvenirliğine yönelik tereddütler bulunmaktadır. Bu nedenle, rastgele orman algoritmasının varsayılan önem dereceleri dikkate alınarak öznitelik seçilmesi yerine, özellikle yukarıda bahsedilen permütasyon ve sütun kaldırma gibi tekniklerin kullanılması yerinde olacaktır (Parr ve ark., 2021).

Permütasyon yöntemi ile bir özniteliğin öneminin hesaplanması için öncelikle bir doğrulama seti veya rastgele orman gibi algoritmaların sağladığı OOB örnekleri rastgele orman algoritmasından geçirilerek bir temel doğruluk oranı elde edilmektedir. Daha sonra tek bir özniteliğin değerleri değiştirilerek aynı şekilde yeni bir doğruluk

oranı hesaplanmaktadır. Bahse konu özniteliğin önemi ise temel doğruluk oranı ile öznitelik değerlerinin yer değiştirilmesiyle elde edilen doğruluk oranı arasındaki fark olarak bulunmaktadır. Son adımda ise özniteliklerin önem derecelerinin birbirine göre durumuna bakılarak seçim yapılabilmektedir. Permütasyon yönteminde modelin tekrar eğitilmesine ihtiyaç yoktur, yalnızca öznitelik değerlerinde yer değiştirme işlemi yapılarak daha önceden eğitilmiş model ile yeni doğruluk oranları hesaplanmaktadır. Ancak yine de bu yöntem rastgele orman algoritmasının varsayılan yöntemine kıyasla hesaplama açısından daha fazla maliyetlidir. Ancak, permütasyon yönteminin sonuçlarının doğruluk oranı varsayılan yöntemden daha yüksektir. Bahse konu yöntem herhangi bir makine öğrenmesi algoritması ile kullanılabilecek olmakla birlikte rastgele orman algoritması, OOB örneklerini sağlaması ve bu nedenle doğrulama setleri oluşturulması gerekliliğini ortadan kaldırması nedeniyle tercih edilmektedir.

Genel olarak öznitelik değerlerinin karıştırılmasının model doğruluğu üzerindeki etkisini ölçmek için kullanılan permütasyon yönteminde belirlenen modelin tekrar tekrar eğitilmesine gerek duyulmaması, modeli eğitmenin maliyeti dikkate alındığında oldukça önemli bir kazanç oluşturmakla birlikte birbirleriyle korelasyonu bulunan özniteliklere yönelik potansiyel bir önyargı oluşmasına da sebep olmaktadır. Bu noktada, modeli yeniden eğitmenin hesaplama maliyetini göz ardı edilirse, sütun kaldırma yöntemi ile en doğru öznitelik önem derecelerinin elde edilmesi mümkündür. Sütun kaldırma yönteminde temel hareket noktası permütasyon yöntemine benzemekte olup, bu yöntemde de öncelikle temel bir performans değeri elde edilmektedir. Ancak sonrasında bir sütun yani öznitelik veri setinden tamamen kaldırılmakta ve yeniden eğitilen model ile yeni bir performans değeri hesaplanmaktadır. Veri setinden çıkarılan özniteliğin önem derecesi ise yeni elde edilen performans değeri ile temel performans değeri arasındaki fark olarak bulunmaktadır. Dolayısıyla, bu modelde bir özniteliğin genel olarak modelin performansı üzerinde ne kadar etkili olduğu üzerine yoğunlaşılmaktadır. Sütun kaldırma yöntemi, permütasyon yöntemine kıyasla daha doğru sonuçlar vermekle birlikte daha maliyetlidir. Söz konusu maliyet tüm eğitim veri seti yerine bu eğitim veri setinden elde edilen alt kümelerin kullanılmasıyla bir nebze azaltılabilmektedir.

Sütun kaldırma yöntemi de permütasyon yöntemine benzer olarak herhangi bir algoritmayla kullanılabilmekle birlikte rastgele orman algoritmasında ayrıca bir doğrulama veri setine ihtiyaç olmaması ve performans ölçümü için doğrudan algoritma tarafından sağlanan OOB skorunun kullanılabilmesi nedeniyle çoğunlukla rastgele orman algoritması ile birlikte kullanılmaktadır.

Diğer taraftan, hem permütasyon yönteminde hem de sütun kaldırma yönteminde her öznitelik ayrı ayrı ele alınmaktadır. Bu, her özniteliğin birbirinden bağımsız olması ve aralarında korelasyon bulunmaması durumunda bir problem oluşturmayacaktır. Ancak, veri setinde aralarında korelasyon bulunan özniteliklerin bulunması halinde bahse konu yöntemler beklenmedik sonuçlar verebilecektir. Örneğin, sütun kaldırma yöntemi dikkate alındığında başka bir öznitelik ile korelasyon içeren bir öznitelik kaldırıldığında temel performansta çok büyük bir değişim olamayacağından dolayı söz konusu özniteliğin önemi oldukça düşük görünecektir. Dolayısıyla, birbiri ile korelasyon içeren öznitelikler önemleri paylaşıyor gibi olacağından önemli olsalar dahi önemsiz gibi görünmeleri söz konusu olacaktır. Bu kapsamda, gerek permütasyon yöntemi gerekse de sütun kaldırma yöntemi kullanılırken korelasyon içeren öznitelikler incelenmeli ve her iki yönteminde değer olarak farklı olsa dahi önem sırası olarak çok benzer öznitelikleri belirlemesi önem arz etmektedir.

2.6. Performans Ölçüm Kriterleri

Sınıflandırma algoritmaları kullanılarak oluşturulan modellerin başarı oranının incelenmesi yani hangi modelin daha doğru sonuçlar elde edebildiğine karar verilebilmesi amacıyla farklı performans kriterleri kullanılmaktadır. Bu kriterlerin bazılarının hesaplanmasının temelinde hata matrisi oluşturulması bulunmakta olup, bu bölümde hata matrisi ve bahse konu performans kriterleri hakkındaki bilgilere yer verilmektedir (Japkowicz, 2011).

2.6.1. Hata Matrisi

Makine öğrenmesinde, sınıflandırma algoritmalarının ne kadar başarılı olduğunun görselleştirilebilmesi için karmaşıklık matrisi olarak da bilinen hata matrisi kullanılmaktadır. Bu matriste gerçek durumlar ile sınıflandırma algoritmalarının tahminlerini yansıtan durumlar bulunmaktadır. Dolayısıyla, hata matrisi sınıflandırma algoritmalarının performanslarının değerlendirilebilmesine olanak tanıyan doğru ve yanlış tahmin edilen değerleri içermekte olup, söz konusu matrisin satırlarında gerçek sınıf değerleri, sütunlarında ise tahmin edilen sınıf değerleri bulunmaktadır. Satır ve sütunların temsil ettiği değerlerin tam tersi şeklinde uygulanması da mümkündür. Örnek olarak, aşağıda çizelge 2.5de ikili sınıflandırıcı için oluşturulan bir hata matrisi verilmektedir. Söz konusu ikili hata matrisi aşağıdaki değerlerden oluşmaktadır (Feng ve ark., 2020).

Doğru pozitif (DP): Algoritmanın doğru tahmin ettiği pozitif verileri ifade etmektedir.

Doğru negatif (DN): Algoritmanın doğru tahmin ettiği negatif verileri ifade etmektedir.

Yanlış pozitif (YP): Gerçekte negatif olan bir verinin algoritma tarafından pozitif olarak tahmin edildiğini ifade etmektedir. Yani, bu durumda negatif örneğin yanlış sınıflandırılması söz konusudur.

Yanlış negatif (YN): Gerçekte pozitif olan bir verinin algoritma tarafından negatif olarak tahmin edildiğini ifade etmektedir. Yani, bu durumda pozitif örneğin yanlış sınıflandırılması söz konusudur.

Çizelge 2.5 Hata Matrisi Örneği.

	Tahmin Değeri: Hayır	Tahmin Değeri: Evet
Gerçek Değeri: Hayır	DN=50	YP=10
Gerçek Değeri: Evet	YN=5	DP=100

İkiden fazla sınıfın bulunduğu durumlarda yani m ≥ 2 olacak şekilde m sınıf verildiğinde hata matrisi m x m boyutlu olmaktadır. Söz konusu matriste bir modelin doğru tahminde bulunduğu veriler matrisin diyagonalinde yer almaktadır. Böyle bir matriste DP, DN, YP, YN hesaplamalarının yapılabilmesi amacıyla öncelikle her bir sınıf için odaklanılan sınıf ve diğerleri şeklinde hata matrisinin yeniden yapılandırılması sonucunda ikili hata matrisinin elde edilmesi gerekmektedir. Elde edilen bu yeni matristeki odaklanılan sınıf ve diğerleri şeklindeki yapı yukarıda açıklanan ikili sınıflandırıcılar için oluşturulan hata matrisine benzemektedir. Dolayısıyla, DP, DN, YP, YN değerlerinin bu matris üzerinden elde edilmesi mümkündür. Ancak, bu durumda sınıf sayısı kadar ikili matris oluşturulmuş olacağından söz konusu değerlerin hesaplanması işlemi her bir ikili matristen elde edilen değerlerin toplamı olarak yapılmaktadır.

Bu doğrultuda, sınıflandırma algoritmalarının performansını hata matrisinin kullanılmasıyla ölçmek üzere oluşturulan doğruluk, belirginlik, hassasiyet, geri çağırma, f1 skoru, AUC ve ROC eğrisi metriklerine aşağıdaki bölümlerde yer verilmektedir (Feng ve ark., 2020).

2.6.2. Doğruluk

Doğruluk oranı, doğru şekilde sınıflandırılmış örneklerin toplam örnek sayısına bölünmesiyle elde edilmektedir. Doğruluk oranı hesaplanırken tüm hata türleri dikkate alınmakta ve tüm örnekler aynı derecede önemsenmektedir. Dolayısıyla, aşağıda şekil 2.9'da formülüne yer verilen doğruluk oranının sınıflandırma işleminde kullanılan algoritmanın toplam performansının değerlendirilmesinde kullanılması söz konusudur. Bununla birlikte veri setinin dengesiz dağıldığı durumlarda etkili olduğunu

söylemek mümkün değildir. Şöyle ki, bir kişinin hasta olup olmadığının tahmin edilmesine yönelik olarak geliştirilen modelde örneklem veri setinde hasta kişilerin sayısı veri setinin yalnızca %5 gibi küçük bir kısmını oluşturuyorsa rastgele bir tahminde bulunulması durumunda dahi kişinin hasta olmadığının %90 başarı ile tahmin edilmesi mümkün olacaktır. Böyle durumlarda da yüksek doğruluk oranı yüksek performans anlamına gelmeyecektir.

$$Doğruluk = \frac{DP + DN}{DP + DN + YP + YN}$$

Şekil 2.9 Doğruluk Oranı Hesaplama Formülü.

2.6.3. Belirginlik

Belirginlik, algoritmanın negatif örnekleri sınıflandırmadaki başarısını ölçmek için kullanılmaktadır. Dolayısıyla, belirginlik oranı doğru negatif değerinin doğru negatif değeri ile yanlış pozitif değerinin toplamına yani toplam negatif örnek sayısına bölünmesiyle elde edilmekte olup ilgili formül aşağıda şekil 2.10'da verilmektedir.

$$Belirginlik = \frac{DN}{DN + YP}$$

Şekil 2.10 Belirginlik Oranı Hesaplama Formülü.

2.6.4. Hassasiyet

Hassasiyet ile toplam pozitif olarak tahmin edilen örneklerin doğru tahmin edilme oranı ölçülmektedir. Dolayısıyla, hassasiyet oranı doğru sınıflandırılan pozitif örneklerin pozitif olarak tahmin edilen toplam örnek sayısına bölünmesiyle elde edilmektedir. Hassasiyet hesaplama formülüne aşağıda şekil 2.11'de yer verilmektedir.

$$Hassasiyet = \frac{DP}{DP + YP}$$

Şekil 2.11 Hassasiyet Oranı Hesaplama Formülü.

2.6.5. Duyarlılık

Duyarlılık oranı, doğru tahmin edilen pozitif örneklerin toplam pozitif örneklere bölünmesiyle hesaplanmakta olup, duyarlılık hesaplama formülüne aşağıda şekil 2.12'de yer verilmektedir. Yani, duyarlılık ile algoritmanın pozitif örnekleri tahmin etmedeki etkinliği ölçülmektedir.

$$Duyarlılık = \frac{DP}{DP + YN}$$

Şekil 2.12 Duyarlılık Oranı Hesaplama Formülü.

2.6.6. F1 Skoru

Genellikle hassasiyet oranı yükseldiğinde duyarlılık oranı azalmakta yani hassasiyet oranın artması için duyarlılık oranından feragat edilmesi söz konusu olmaktadır. Dolayısıyla, bahse iki oran arasında bir denge bulunmakta olup, bu dengenin bulunması amacıyla hassasiyet oranı ve duyarlılık oranının kapsamlı bir şekilde ele alındığı f1 skoru kullanılmaktadır (Pillai ve ark., 2017). Bu kapsamda, f1 skoru hassasiyet ve duyarlılık metriklerinin harmonik ortalaması olarak hesaplanmakta olup, f1 skorunun hesaplanmasına ilişkin formül aşağıda şekil 2.13'te verilmektedir.

$$F1\ Skoru = \ 2 * \frac{Hassasiyet * Duyarlılık}{Hassasiyet + Duyarlılık}$$

Şekil 2.13 F1 Skoru Hesaplama Formülü.

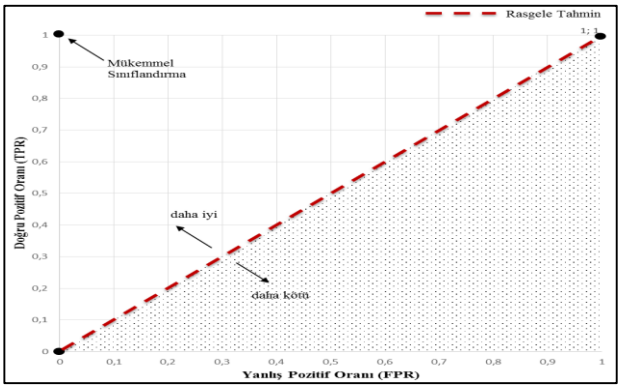
2.6.7. ROC /AUC

AUC-ROC eğrisi sınıflandırma algoritmalarının performanslarının değerlendirilmesinde sıklıkla kullanılmakta olup, formüllerine aşağıda şekil 2.14'de verilmekte olan yanlış pozitif oranına karşılık gerçek pozitif oranı için çizilen bir grafiktir.

$$Doğru\ Pozitif\ Oranı = \frac{DP}{DP + YN} \qquad Yanlış\ Pozitif\ Oranı = \frac{YP}{YP + DN}$$

Şekil 2.14 DP ve YP Oranı Hesaplama Formülü.

Bu kapsamda, aşağıda şekil 2.15'de örnek bir ROC eğrisi verilmektedir. Söz konusu şekilden de görüleceği üzere DP oranının yüksek, YP oranın düşük olduğu modellerin diğer modellerden daha iyi olduğu söylenebilecektir. Bu sebeple, bir modelin YP oranının 0, DP oranın ise 1 olması en istenilen durumdur. Dolayısıyla bu duruma ulaşan modellerin mükemmel sınıflandırma yapabilmesi mümkün olup, modellerin şekil 2.15'de sol üst köşeye yakın olması tercih edilmektedir. YP oranın DP oranına eşit olması ise rastgele sınıflandırmanın sonucunu temsil etmektedir ki bu durum şekil 2.15'de yer alan köşegendir. Söz konusu köşegenin altında kalan kısım rastgele sınıflandırmadan daha kötü bir sonucu temsil etmektedir.



Şekil 2.15 ROC Eğrisi Örneği (Fawcett, 2006).

AUC ise ROC eğrisi altında kalan alanı ifade etmekte olup, 0 ila 1 arasında değer almaktadır. AUC değerinin 0 olması yapılan tüm tahminlerin yanlış olduğunu ifade

etmektedir. Yani AUC, sınıflar arasında ne kadar iyi ayrım yapabildiği hakkında bilgi sağlamakta olup, değer ne kadar büyükse algoritmanın sınıfları ayırma konusunda ve genelleme performansında o kadar başarılı olduğu söylenebilecektir (Fawcett, 2006).

2.7. Boyut Azaltma Yöntemleri

Modellerin oluşturulması esnasında yüksek boyutlu yani öznitelik sayısı çok fazla olan veri setlerinin kullanılması durumunda girdi değişkinlerine ilişkin yeterli miktarda permütasyonun keşfedilebilmesi için aynı zamanda veri setindeki örneklerin sayısının da fazla olması gerekmektedir. Bu problemle mücadele edebilmek adına genellikle bir nebze doğruluk oranına karşılık basitliğin amaçlandığı boyut azaltma teknikleri ve onların odak noktası olan birbirleriyle korelasyon içeren özniteliklerin incelenmesi konusu gündeme gelmektedir. Boyut azaltma tekniklerinden biri olan öznitelik çıkarımı, yüksek boyutlu bir öznitelik uzayından daha küçük boyutlu yeni bir öznitelik uzayının elde edilmesini sağlamaktadır. Öznitelik çıkarımı gibi boyut azaltma tekniklerinin uygulanmasıyla algoritmaların performanslarının geliştirilmesi, hesaplama karmaşıklığının azaltılması, modeller için gerekli belleğin azaltılması, modellerin daha kolay genelleştirilmesi ve daha güvenilir hala gelmesi sağlanabilmektedir. Çünkü belirtilen şekilde veri setinin boyutunun azaltılması sonucunda oluşturulan yeni öznitelik uzayında en ilgili öznitelikler bulunmaktadır. Ayrıca daha düşük boyuta sahip veri setlerinin görselleştirilebilmesi ve makine öğrenmesi algoritmaları tarafından daha hızlı analiz edilebilmesi mümkündür. Bu tez çalışmasında, sıklıkla kullanılan boyut azaltma yöntemlerinden biri olan Temel Bileşen Analizi (PCA) tercih edilmiş olup, söz konusu yönteme ilişkin bilgilere aşağıda değinilmektedir.

PCA yöntemi ile orijinal veri setindeki bilgilerin mümkün olan en yüksek oranda korunarak veri setinin daha düşük boyutlu bir temsilinin bulunması amaçlanmaktadır. Bu amaç çerçevesinde PCA, orijinal veri setinde bulunan çok sayıdaki ilişkili değişkenin temsil etmekte olduğu varyansı, temel bileşenler adı verilen daha az sayıdaki temsili değişkenlerle toplu olarak açıklamakta ve özetlemektedir. PCA'deki

bilgi miktarı her bir temel bileşen için açıklanan varyans oranını ifade etmekte olup, söz konusu varyans oranı 0 ila 1 arasında değer almaktadır (Dang ve Nilsson, 2020).

PCA yönteminin uygulanmasının ilk adımı, baskın değişkenlere önyargılı olunmasının önüne geçilmesi amacıyla sürekli değişkenlerin standardizasyon işleminin gerçekleştirilmesidir. İkinci adım olaraksa kovaryans matrisinin hesaplanması yapılmakta olup, bu adım ile orijinal veri setindeki öznitelikler arasında korelasyon olup olmadığı incelenerek veri setinde gereksiz bilgi temsili olup olmadığının anlaşılması amaçlanmaktadır. Sonraki adımda da, temel bileşenlerin belirlenmesi için kovaryans matrisinin özdeğerleri ve özvektörleri hesaplanmaktadır. Söz konusu temel bileşenler, başlangıç değişkenlerinin doğrusal kombinasyonları veya karışımları olarak oluşturulan yeni değişkenlerdir. Bu temel bileşenler arasında korelasyon bulunmamakta olup, orijinal veri setindeki bilgilerin çoğu ise ilk temel bilişene sıkıştırılmaktadır. Bahsedilen durum geometrik olarak belirtilecek olursa temel bileşenler, maksimum varyans miktarını açıklayan verilerin yönlerini, yani verilerin çoğunun bilgisini yakalayan çizgileri temsil etmektedir. Dolayısıyla, PCA yönteminde düşük bilgiye sahip değişkenlerin atılması söz konusu olduğundan çok fazla bilgi kaybı olmadan boyut azaltılması yapılabilmektedir. Kovaryans matrisinin özvektörleri en fazla varyansı içeren eksenlerin yönlerini yani yukarıda açıklanan temel bileşenleri vermekte iken özdeğerler ise her bir temel bileşende taşınan varyans miktarını vermektedir. Öz vektörlerin, öz değerlerine göre büyükten küçüğe sıralanmasıyla temel bileşenler önem sırasına göre elde edilebilmektedir. Bu kapsamda son olarak, istenen boyut indirgeme miktarına bağlı olarak özvektörler seçilmektedir.

Uygulama adımlarına yukarıda yer verilen PCA yöntemindeki en önemli konulardan bir tanesi temel bilişen sayısına yani yeterli alt uzaya karar verilmesidir. Çünkü öznitelik boyutu küçültülürken aynı zamanda orijinal veri setindeki bilginin çok fazla kaybedilmemesi gerekmektedir. Bu gerekliliğin yerine getirilip getirilmediğinin ya da ne kadar bilgi kaybedildiğinin anlaşılabilmesi için öncelikle orijinal öznitelik uzayındaki varyansa göre her bir temel bileşendeki varyans gözlemlenmekte, sonrasında ise orijinal varyans ile yeni alt uzayda bulunan varyansın

karşılaştırılması yapılmaktadır. Mevcut durumda, PCA yönteminde kullanılacak temel bileşen sayısının belirlenmesine yönelik kabul edilmiş bir yöntem bulunmamaktadır. Ancak, bu konuda yaygın olarak kullanılan yöntemlerden bir tanesi *Elbow* yöntemidir. Bahse konu yöntem esas olarak denetimsiz kümeleme algoritmalarında uygun küme sayısına karar verilmesi amacıyla kullanılmakla birlikte PCA yönteminde bulunacak temel bileşen sayısının belirlenmesi amacıyla da kullanılabilmektedir. *Elbow* yönteminde, genel olarak varyans oranının düşmeye başladığı noktanın tespiti yapılmaktadır. Söz konusu tespitin yapılabilmesi için de varyansın temel bileşen sayısının fonksiyonu olduğu bir yamaç grafiği oluşturulmakta ve değişim oranının düşmeye başladığı nokta yani dirsek noktası gözlemlenmektedir. Dolayısıyla, *Elbow* yönteminin temel mantığı söz konusu grafiğin incelenerek dirsek noktasının bulunmasına dayanmaktadır (Dang ve Nilsson, 2020). Her bir temel bileşenin etkisinin yine söz konusu yamaç grafiğinden gözlemlenmesi mümkün olup, ilk temel bileşen en yüksek varyans miktarını içermektedir. Yani, bu temel bileşen herhangi bir temel bileşenden daha fazla değişkenliğe neden olmaktadır.

2.8. Hiper Parametre Seçim Yöntemleri

Yapılan birçok çalışmada, makine öğrenmesi algoritmalarının performansını etkileyen iki önemli faktörün, hiper parametrelerin değeri ve seçilen öznitelikler olduğu belirtilmektedir. Özniteliklerin belirlenmesi konusuna bölüm 2.5'de değinilmiş olup, bu bölümde hiper parametrelerin belirlenmesi konusu üzerinde durulacaktır.

Makine öğrenmesi algoritmalarında hiper parametre, ilgili modeli oluşturan kişi tarafından karar verilmesi gereken bir değerdir ve çözülmesi hedeflenen problem ile kullanılmakta olan veri seti gibi unsurlara bağlı olarak değeri değişiklik gösterebilmektedir. Örneğin, k-en yakın komşu algoritmasındaki k değeri bir hiper parametredir. Çoğunlukla oluşturulan modellerin başarı oranı hiper parametrelere de bağlı olabilmektedir. Dolayısıyla, başarı oranının artırılabilmesi ve nihayetinde modelin amaca uygun olabilmesi için en uygun hiper parametrelerin seçilmesi gerekmektedir. Hiper parametrelerin seçilmesi genel olarak modeli oluşturan kişilerin

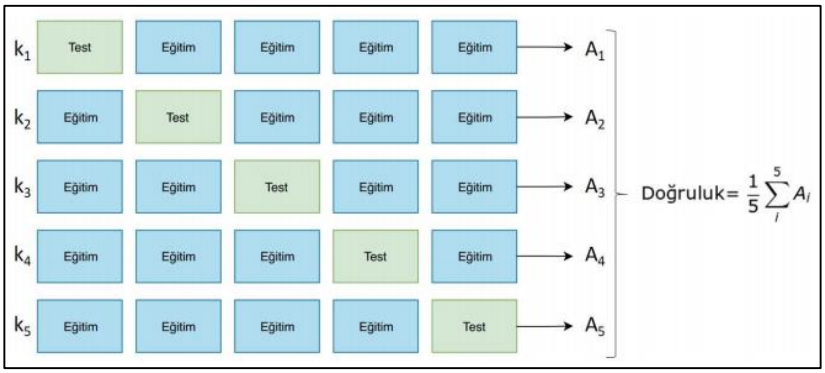
sezgisine, tecrübesine ve daha önceki uygulamaların etkisine bağlı olabilmektedir. Bununla birlikte son zamanlarda, hiper parametrelerin optimum değerinin belirlenmesi için sezgisel arama, ızgara araması ve *Bayes* yöntemi gibi çeşitli teknikler kullanılmaktadır. Sezgisel arama yönteminde, önceden sahip olunan bilgiler kullanılarak belirlenen hiper parametreler ile model oluşturulmakta ve sonuçları gözlemlenmektedir. Devamında ise modelin başarı oranını artıracağı değerlendirilen yeni hiper parametre tahminleri yapılmakta ve bu işlemlere beklenen sonuç elde edilene kadar devam edilmektedir. Izgara araması yöntemi ise önceden tanımlanmış bir dizi parametreden en iyi performans gösterenlerin belirlenmesi esasına dayanan bir kaba kuvvet uygulamasıdır (Adamsson, 2017). Yani, parametrelerin belirlenmesi için önceden tanımlanmış parametreler çerçevesinde modelin tüm olası varyasyonları değerlendirilmekte ve en başarılı varyasyon nihai model olarak seçilmektedir. Söz konusu yöntemin uygulanmasında, özellikle bazı hiper parametrelerin sonsuz değer alabileceği dikkate alındığında sahip olunan ön bilgiler kullanılarak hiper parametrelerin alabileceği değerler için aralık belirlenmesi önem arz etmekte olup, belirlenen bu aralıklardan ana noktalar seçilerek hareket edilmektedir. Sonuç olaraksa, söz konusu ana noktaların değerleri kullanılarak oluşturulan tüm kombinasyonlar ile modeller test edilmekte ve en iyi sonuca ulaşan kombinasyonlarda karar kılınmaktadır. *Bayes* yönteminde, bir önceki deneyim sonuçları bir sonraki hiper parametre seçiminde kullanılmakta ve bilinen *Bayes* teoremi üzerinden olasılık hesabı ile hiper parametreler belirlenmeye çalışılmaktadır.

Diğer taraftan, yukarıda belirtilen yöntemlere ek olarak derin öğrenme ve genetik algoritma gibi halen bu alanda gelişim aşamasında olan daha akıllı tekniklerin de hiper parametrelerin belirlenebilmesi amacıyla kullanılabilmesi söz konusudur. Ancak hangi yöntemin kullanılacağı genellikle geliştirilecek modelin ve belirlenecek hiper parametrelerin karmaşıklığı ile kullanılacak yöntemin yeterli olup olmayacağına bağlı olarak değişiklik gösterebilmekte olup, bu tez çalışmasında ızgara araması yöntemi uygulanmıştır.

2.9. Çapraz Doğrulama

Daha önceki bölümlerde de bahsedildiği üzere, makine öğrenmesi algoritmalarının uygulanmasında veri seti eğitim ve test olmak üzere iki gruba ayrılmaktadır. Eğitim veri seti kullanılarak makine öğrenmesi modelleri oluşturulmakta, test veri seti kullanılarak da modellerin performansları değerlendirilmektedir. Ancak performansın bu şekilde yalnızca bir eğitim ve test verisi üzerinden değerlendirilmesi durumunda eğitim ve test veri setlerinin tesadüfen iyi sonuçlar verecek şekilde denk gelip gelmediğinin tespit edilmesi mümkün olmamaktadır. Böyle bir durumda da modelin uygulanmaya alınması beklenmeyen sonuçlarla karşılaşılması sonucunu ortaya çıkarabilecektir. Bu noktada makine öğrenmesi modellerinin performanslarının daha etkin bir şekilde değerlendirilebilmesi amacıyla çeşitli yöntemler uygulanmakta olup, bu tez çalışmasında da bahse konu amaç için k-kat çapraz doğrulama yöntemi uygulanmıştır.

K-kat çapraz doğrulama yönteminde öncelikle veri seti rastgele karıştırılarak k eşit gruba ayrılmaktadır. Sonrasında bir grup test veri seti olarak seçilmekte ve diğer tüm gruplar modelin eğitilmesin kullanılmaktadır. Bahse konu eğitim veri seti ile oluşturulan modelin performansı ilgili test veri seti üzerinden ölçülmekte ve saklanmaktadır. Açıklanan bu işlemler her bir grup için tekrarlanmakta ve her bir adımda elde edilen performans kriterinin ortalaması alınarak modelin nihai performansı belirlenmektedir. Böylece, bahsedilen yöntemle makine öğrenmesi modellerinin geliştirme ortamında başarılı sonuçlar verirken uygulamada başarısız olması probleminin önüne geçilebilmekte ve oluşturulan modeller daha kararlı hale gelmektedir (Hastie ve ark., 2001). Aşağıda şekil 2.16’da 5-kat çapraz doğrulama işlemi örnek olarak gösterilmektedir.



Şekil 2.16 Çapraz Doğrulama Örneği (Ustuner ve Sanli, 2020).

K-kat çapraz doğrulama yönteminde veri setinin bölüneceği grup sayısını gösteren k parametresinin belirlenmesi gerekmektedir. Yapılan bazı çalışmalarda k değerinin 2,5 ve 10 olarak kullanılmakta olduğu görülmekle birlikte en sık kullanılan ve çoğunlukla önerilen k değeri 10'dur.

3. BULGULAR

Terörist saldırılar hemen hemen dünyanın her bir köşesinde insanlık için büyük bir problemdir. Terörist saldırılara yönelik yapılan incelemelerin yönlendirilebilmesi ve daha kısa sürede sonucu ulaşılabilmesi amacıyla bulguların elde edilebilmesine yönelik bu uygulamada, bir terör olayı meydana geldikten sonra olayın hangi terörist grup tarafından yapılmış olabileceğine ilişkin tahminde bulunulması amaçlanmıştır. Bu amaç doğrultusunda, öncelikle Küresel Terörizm Veritabanı'nı üzerinde veri ön işleme sürecinin gerçekleştirilmesiyle yeni bir veri seti elde edilmiştir. Daha sonra, bu veri seti üzerinde tezin gereç ve yöntem bölümünde detayları açıklanan makine öğrenmesi algoritmaları ile çeşitli modeller oluşturularak sınıflandırma işlemleri gerçekleştirilmiş ve söz konusu bölümde açıklanan yöntemlerle bu modellerin hız ve performans olarak geliştirilmesi üzerinde durulmuştur. Son olarak ise farklı algoritmalar kullanılarak oluşturulan modellerin yine tezin gereç ve yöntem bölümünde detayları açıklanmış olan performans ölçüm kriterlerine göre karşılaştırması yapılmıştır. Söz konusu uygulamanın geliştirilmesinde *Python* programlama dili kullanılmıştır.

Bu doğrultuda, bahse konu uygulama ile örnek bir veri üzerinden sorumlu grupları ve ilgili gerçekleri bulacak modellerin uygulanıp test edilmesi ve ham verilerin güvenlik uzmanlarının çalışmalarında kullanılabilecek yararlı bilgiye dönüştürülerek bu bilgiler doğrultusunda tahminlerde bulunulabilmesi aşamaları gösterilmiş olup, söz konusu uygulamanın terör saldırıları ile ilgili incelemelerin yanı sıra diğer adli incelemeler için de yol gösterici olacağı düşünülmektedir.

3.1. Deneysel Kurulum

Bu tez çalışmasının bulguların elde edilebilmesine ilişkin deneysel kurulumu; Küresel Terörizm Veri Tabanı'nda bulunan verilerin ham veri seti olarak ele alınması, ham veri setinin temizlenmesi, ham veri setinde öznitelik seçimi yapılarak yeni veri

setleri oluşturulması, oluşturulan yeni veri setlerinin eğitim veri seti ve test veri seti olarak 2 gruba ayrılması, terörist grupların tahmin edilebilmesi için eğitim veri seti ve makine öğrenmesi algoritmaları kullanılarak sınıflandırma modellerinin oluşturulması, veri setlerinde boyut azaltma ve algoritmalarda parametre ayarlama işlemleri yapılarak modellerin performansının artırılmaya çalışılması, modellerin 10-kat çapraz doğrulama yöntemi ile test veri setleri kullanılarak test edilmesi, farklı algoritmalar kullanılarak oluşturulan modellerin performans kriterlerine göre incelenmesi ve başarılı modellere karar verilmesi olarak sunulabilecek olup, çalışmada kullanılan programlama dili ve kütüphaneler ile geliştirme ortamına ilişkin detaylı bilgilere ilerleyen bölümlerde verilmektedir.

3.1.1. Geliştirme Ortamı

Makine öğrenmesi alanında bulguların elde edileceği uygulamalar geliştirirken karşılaşılan problemlere hızlı bir şekilde çözüm bulabilmek önem arz etmektedir. Bu nedenle, uygulamaların geliştirileceği ortamın ve programlama dilinin de bu husus dikkate alınarak belirlenmesi gerekmektedir. Bu doğrultuda, makine öğrenmesi alanında çok fazla araştırma ve uygulamada tercih edilmiş olan açık kaynak kodlu *Python* programlama dili uygulama geliştirme dili olarak belirlenmiştir. *Python* programlama dili *Anaconda* platformu üzerinden kullanılmıştır. Kaynak kodların test edilmesi ise *Google Colaboratory* platformunda yapılmış ve ayrıca bu platformda da yapılan ilave kodlamalarda *Python* programlama dili kullanılmıştır.

Anaconda, veri bilimi ve makine öğrenmesi gibi alanlarda uygulama geliştirilmesinde *Python* veya R programlama dillerini tercih edenler için hazırlanmış ve içerisinde birçok hazır paket içermekte olan tümleşik bir platformdur. Söz konusu platformun içerisinde veri bilimi, makine öğrenmesi ve yapay zeka çalışmalarında sıklıkla kullanılan kütüphanelere ek olarak *Spyder* ve *Jupiter Notebook* gibi geliştirme araçları da bulunmaktadır. Bu çalışmada geliştirme aracı olarak *Spyder* ve *Google Colaboratory* kullanılmıştır. Dolayısıyla, *Anaconda* ilgili paketlerin yönetimini sağlayarak uygulama için uygun bir geliştirme ortamı sağlamıştır.

Python programla dili makine öğrenmesinin yanında sistem programlama, bilimsel hesaplama ve ağ uygulamalarının geliştirilmesi gibi birçok alanda kullanılmakta olup, derleyiciye ihtiyaç duymaksızın çalışabilen, nesne tabanlı, açık kaynak kodlu ve yüksek seviyeli bir programlama dilidir. Bu programlama dilinin söz dizimi hem yazması hem de anlaşılması açısından oldukça kolaydır. Ayrıca, *Python* dinamik yazma ve bağlamaya imkan tanımaktadır. Dolayısıyla, *Python* programla dili ile hızlı bir şekilde program geliştirilmesi mümkün olmaktadır. Bu çalışmada kullanılan bazı *Python* kütüphanelerine ise aşağıda yer verilmektedir.

3.1.2. Kullanılan Kütüphaneler

Python programlama dili, makine öğrenmesi ile ilgili uygulamalarda verilerin işlenmesini ve sınıflandırma algoritmaları kullanılarak modellerin geliştirmesini sahip olduğu kütüphaneler sayesinde oldukça basit bir hale getirebilmektedir. Bu doğrultuda, genel olarak bu çalışmada kullanılan kütüphaneler hakkındaki bilgiler aşağıda sunulmaktadır.

Numpy, veri ön işleme sürecinde ve verilerin hazırlanması aşamasındaki matematiksel işlemlerin yapılması amacıyla kullanılmış olup, genel olaral diziler ve matrisler üzerinde hesaplamalar yapmayı sağlamaktadır.

Pandas kütüphanesi, *Numpy* kütüphanesi ile birlikte verilerin hazırlanması için kullanılmış olup, genel olarak veri setinin yüklenmesi, veri formatının ayarlanması ve veri setinden çerçeve oluşturarak işlemlerin yapılması amacıyla kullanılmaktadır.

Scikit-Learn kütüphanesi, sınıflandırma algoritmalarıyla model oluşturulması amacıyla kullanılmış olup, genel olarak sınıflandırma, kümeleme ve regresyon için kullanılmakta olan birçok algoritmayı içermektedir.

Tensorflow, ekran kartının özelliğine bağlı olarak cpu veya gpu kullanımına uygun olan açık kaynaklı bu kütüphane *Google* tarafından derin öğrenme ve makine öğrenmesi alanlarında kullanılmak üzere geliştirilmiştir.

Keras, TensorFlow'un üzerinde çalışabilen ve derin öğrenme modellerinin tanımlanması ve eğitilmesi için kullanılmakta olan bir üst düzey sinir ağları uygulama programlama arayüzü (API) 'dür. *Keras*, hızlı sonuç elde edilebilmesi için geliştirilmiş olup, *Python* programlama diliyle yazılmıştır.

3.2. Veri Setinin Toplanması

Herhangi bir terör olayı gerçekleştikten sonra sorumlu olabilecek grubun tahmin edilebilmesine yönelik makine öğrenmesi modellerinin oluşturulması sürecinde ilgili algoritmaların eğitilmesi ve test edilmesi için daha önce gerçekleşen terör olayları hakkındaki bilgilere ihtiyaç duyulmaktadır.

Bu tez çalışmasında, makine öğrenmesi algoritmalarının uygulanacağı veri setinin oluşturulması için *Global Terrorism Database* (GTD/Küresel Terörizm Veritabanı) isimli veritabanından faydalanılmıştır (Start, 2019). GTD projesi ilk olarak *Maryland* Üniversitesi ve ABD İç Güvenlik Bakanlığı tarafından başlatılmıştır. Söz konusu proje mevcut durumda, *Maryland* Üniversitesi'nden 15 araştırmacı tarafından sürdürülmekte olup terörizm araştırmaları alanında uzman kişilerden oluşan 9 kişilik bir danışma kurulu tarafından da desteklenmektedir. Bahsedilen bu ekip, halihazırda ABD İç Güvenlik Bakanlığı tarafından finanse edilmekte olan Ulusal Terörizm ve Terörizmle Mücadele Araştırma Birliği (*START*) bünyesine entegre edilmiştir. Dolayısıyla, GTD projesi kapsamında oluşturulan veritabanının güncelliği ve işlerliği *START* tarafından sağlanmaktadır.

Herkesin kullanımına açık bir veritabanı olan GTD içerisinde yer alan veriler; tamamen kitaplar, yasal belgeler, haberler ve diğer medya unsurları gibi açık bir

şekilde bulunan kaynaklardan elde edilmiş ve güvenilirlikleri konu hakkında eğitim almış bir ekip tarafından kontrol edilmiştir. Ancak her ne kadar güvenirlilik kontrolü yapılsa dahi medya kaynaklarına olan bu bağımlılığın doğal sonucu olarak sansür uygulanması veya yanlış bilgilendirme yapılmasından kaynaklanan yanlışlıkların veri setinde yer alması riski bulunmaktadır. Dolayısıyla, GTD'nin terör olaylarına ilişkin eksiksiz bir veri seti olduğunu söylemek mümkün olmamakla birlikte mevcut durumda terör olaylarına ilişkin en kapsamlı sınıflandırılmamış verileri içerdiğinin iddia edilmesi mümkündür. Öyle ki GTD veri setinde, 135 adet öznitelik kullanarak 1970 yılından 2018 yılına kadar gerçekleşmiş 190.000'den fazla küresel terör olayı tanımlanmakta, diğer bir deyişle söz konusu olaylara ilişkin açıklayıcı bilgilere yer verilmektedir. Herhangi bir olayın bahse konu öznitelikler kullanılarak veritabanına kaydedilmesi aşamasında ise GTD tarafından benimsenmiş olan terörist saldırı tanımından hareket edilmekte olup, bahsedilen tanımda terörist saldırının; korku, baskı veya gözdağı yoluyla siyasi, ekonomik, dini veya sosyal bir hedefe ulaşmak için devlet dışı aktörler tarafından yasadışı güç ve şiddet kullanılması veya kullanılacağına yönelik tehdidin bulunması olduğu belirtilmektedir. Dolayısıyla, tanım çerçevesinde bir olayın GTD'ye kaydedilebilmesi için bahse konu olayın; kasıtlı olma, şiddete veya doğrudan şiddet tehdidine neden olma ve failinin alt ulusal olması özelliklerini taşıması gerekmektedir.

Bu kapsamda, bazı araştırmacılar tarafından terör saldırılarının amacının şiddet eyleminin kendisinden ziyade belirli bir siyasi, dini veya diğer bir amaca ulaşmak olduğunun çokça vurgulanmakta olduğu dikkate alındığında GTD veritabanın kapsamın yeterli olmasına ek olarak bir olayın terör olayı olup olmadığının anlaşılmasına yönelik olarak GTD'de belirlenen kriterlerin de yerinde olduğu anlaşılmaktadır.

3.3. Veri Ön İşleme

Bu bölümde, analizlere hazır hale getirilmiş veri setleri oluşturulması amacıyla bölüm 2.3'de verilen bilgilerden faydalanılarak aşağıdaki işlemler uygulanmıştır.

3.3.1. Veri Setinin İncelenmesi ve Temizlenmesi Süreci

Makine öğrenmesi modelleri geliştirilmeden önce kullanılacak veri setinin tam olarak anlaşılması ve temizlenmesi önem arz etmektedir. Çünkü modellerin etkinliği ve genelleştirilebilirliği kullanılan veri setine doğrudan bağlıdır. Bu bağlamda, öncelikle GTD veri setine ilişkin bilgilerin verilmesi ve bu bilgiler ışığında veri setinin limitlerinden bahsedilmesi modellerin oluşturulma sürecinin daha kolay anlaşılması açısından yerinde olacaktır.

Kullanılan Veri Seti İlişkin Limitler: 1970 yılından 1997 yılına kadar olan olaylarla ilgili veriler, *Pinkerton* Küresel İstihbarat Teşkilatı (*Pinkerton Global Intelligence Services* /PGIS) tarafından gerçek zamanlı olarak sayılabilecek şekilde olayın meydana gelme tarihine yakın tarihlerde toplanmıştır. 1998 ve 2007 yılları arasındaki olaylara ilişkin veriler geriye dönük olarak, 2007 yılından sonra meydana gelen olaylara ilişkin veriler ise tekrardan gerçek zamanlı olarak sayılabilecek şekilde GTD ekibi tarafından toplanmıştır. Esas itibariyle bu durum tüm olaylar için tek tip bir veri toplama metodolojisinin uygulanmamış olduğunu ifade etmektedir. Ayrıca, kaynak çeşitliğinin gelişen teknoloji ile birlikte çoğalmış olması nedeniyle nispeten yakın tarihte meydana gelmiş olaylara ilişkin GTD'de daha fazla bilgi yer almasının yanında zamanla medya kaynaklarının erişilemez hale gelmesi gibi sebeplerle geriye dönük olarak veri toplamasının gerçekleştirildiği 1998 ve 2007 yılları arasında gerçekleşmiş bütün olaylarla ilgili detaylı bilgilerin GTD'ye kaydedilmiş olduğundan söz edilmesi de mümkün değildir (Barkhau, 2017). Ek olarak, GTD ekibi tarafından 1998 yılında PGIS'den veri toplama görevinin alınması sonrasında GTD veri setine yeni öznitelikler eklenmiş olup, bu özniteliklere ilişkin bilgiler yalnızca 1998 yılında ve sonrasında gerçekleşen olaylar için mevcuttur.

1993 yılında meydana gelen olaylara ilişkin veriler, meydana gelen veri kaybı nedeniyle GTD veri setinden tamamen çıkarılmıştır. Bu verilerin kurtarılmaya çalışması durumunda ise yalnızca %15'lik bir kısmının kurtarılmasının mümkün

olmasından dolayı olağandışı şekilde düşük olay sayısının yanlış yorumlanmasının önüne geçilmesi amacıyla bu veriler de GTD veri setine dahil edilmemiştir.

Genel olarak veri toplama yöntemi ve veri toplama süreci ile GTD'nin bakımında zaman içinde meydana gelmiş olan değişiklikler dikkate alındığında GTD'nin 1970 ve 2018 yılları arasında meydana gelmiş bütün terör olaylarına yönelik eksiksiz bir veritabanı olmadığı anlaşılmaktadır. Ancak yine de bu tez çalışmasında, veri karmaşıklığının azaltılarak makine öğrenmesi algoritmalarına yoğunlaşılabilmesi amacıyla GTD'nin, 1970 ve 2018 yılları arasında meydana gelmiş olan tüm terör olaylarına ilişkin verileri içerdiği varsayılmıştır. Bu varsayımın doğal sonucu olarak da bahsedilen metodolojik tutarsızlıkların GTD veritabanında bulunan öznitelikleri nasıl etkilediği ve bu doğrultuda hangi öznitelikler ile olay örneklerinin tez çalışmasında kullanılmasının uygun olacağı hususunun araştırılması gerekmiştir. Bu doğrultuda, ilk adım olarak veri setinin temizlenmesi işlemi yerine getirilmiş olup, bu sürece ilişkin bilgiler aşağıda verilmektedir.

Veri setinin temizlenmesi: Veri madenciliği ve makine öğrenmesi uygulamalarında, veri setinin kalitesi uygulamaların başarısına doğrudan etki etmektedir. Bu kaliteye ulaşmak amacıyla veri seti üzerinde yapılan ön işlemler ise iş yükünün yaklaşık %70'lik bir kısmını oluşturmaktadır (Feng ve ark., 2020). Dolayısıyla, makine öğrenmesi ile ilgili modellerde modelin çıktısı olacak sonuçların geçerliliğinin tam olarak sağlanabilmesi için model oluşturulmadan önce elde bulunan veri setine bağlı olarak bolca zaman ve çabanın verilerin işlenmesi ve anlamlı hale getirilmesi aşamasına harcanması söz konusu olabilmektedir.

Bu kapsamda, bu tez çalışmasında da makine öğrenmesi algoritmalarının uygulanmasına geçilmeden önce yukarıda açıklanan limitleri dikkate alınarak GTD veri setinde yer alan özniteliklerin tam olarak kavranabilmesi için söz konusu veri seti üzerinde keşfedici veri çözümlemesi yapılmıştır. Yapılan bu keşfedici çalışma sonrasında ise veri setinde yer alan bazı örneklerin ve bazı özniteliklerin makine

öğrenmesi algoritmalarının test edilmesi aşamasında katkı sağlamasının mümkün olmadığı anlaşılmıştır.

Bu doğrultuda, GTD veri setinde analiz süreçlerine önem katmayacak olmasından dolayı ilgisiz olarak nitelendirilebilecek, diğer bir deyişle analiz süreçlerine ilişkin kayda değer nitelikte bir özellik taşımayan alanların elimine edilmesi amacıyla öncelikle GTD'ye ilişkin kod çizelgesi (*codebook*) dokümanı incelenmiş ve saldırıyı tanımlayan farklı özelliklerin birleşiminden oluşmakta olan saldırı modelinin GTD'de öznitelikler kullanılarak temsil edilmekte olduğu anlaşılmıştır. GTD'de terör olaylarının temsil edilebilmesi amacıyla 10 ana öznitelik[1] altında çok sayıda alt öznitelik kullanılmaktadır. Mevcut durumda, GTD'de 135 adet öznitelik bulunmaktadır. Bahse konu ana öznitelikler ile alt özniteliklere ilişkin detaylı bilgilere, açıklamalara, tespitlere ve tespitler kapsamında veri setinden çıkarılma nedenlerine ilişkin çizelge Ek/1'de yer sunulmaktadır. Söz konusu çizelgeden de anlaşılabileceği üzere GTD veri setinde; çok fazla eksik değer veya dengesizlik içermekte olan, kategorik değişkenlerin metin karşılığını temsil eden, olaylardan bağımsız olarak veri tabanının tutarlılığının sağlanmasına yönelik olan, metin değişkeni olarak çok farklı değerler içeren ve niteleyici bir özelliğinden ziyade açıklayıcı özellikleri bulunan öznitelikler yer almakta olup, bu tez çalışmasında kullanılacak makine öğrenmesi algoritmaları açısından anlamlı ve önemli olabilecek öznitelikler korunurken veri setinin kontrolünün sağlamasında yaşanabilecek zorlukların bertaraf edilmesi ve verinin etkinliğinin sağlanması amacıyla bahse konu öznitelikler veri setinden çıkarılmıştır. Bu amaçla, ilk olarak GTD'de bulunan 135 adet öznitelikten aşağıda verilen 37 adedi, kategorik değişkenlerin metin karşılığını temsil etmesinden veya olaydan ziyade yalnızca veri tabanı ile ilgili bilgiler içermesinden dolayı makine öğrenmesi algoritmaları açısından gereksiz olması nedeniyle veri setinden çıkarılmıştır. Daha sonra ise kalan 98 adet öznitelikten; çok fazla eksik değer içermesi, eksik değer içeren veya niteleyici özelliği bulunmayan özniteliğe doğrudan bağlı olması ya da yalnızca metin olarak açıklayıcı özelliğinin bulunması sebebiyle

[1] ID and Date ii. Incident Information iii. Incident Location iv. Attack Information v. Weapon Information vi. Target/Victim Information vii. Perpetrator Information viii. Casualties and Consequences ix. Additional Information and Sources

aşağıda verilen 78 adet öznitelik de veri setinden çıkarılmıştır. Veri setinden çıkarılan öznitelikler ile çıkarılma gerekçelerine genel anlamda aşağıdaki çizelge 3.1'de yer verilmektedir.

Çizelge 3.1 Veri Setinden Çıkarılan Öznitelikler ve Çıkarılma Gerekçeleri.

I. Kategorik Değişkenlerin Metin Karşılığını Temsil Eden Öznitelikler (28 Adet)
alternative_txt, country_txt, region_txt, attacktype1_txt, attacktype2_txt, attacktype3_txt, weaptype1_txt, weapsubtype1_txt, weaptype2_txt, weapsubtype2_txt, weaptype3_txt, weapsubtype3_txt, weaptype4_txt, weapsubtype4_txt, targtype1_txt, targsubtype1_txt, natlty1_txt, targtype2_txt, targsubtype2_txt, natlty2_txt, targtype3_txt, targsubtype3_txt, natlty3_txt, claimmode_txt, claimmode2_txt, claimmode3_txt, propcomment, hostkidoutcome_txt
II. Olaydan Ziyade Veri Tabanı İle İlgili Bilgileri İçeren Öznitelikler (9 Adet)
eventid, crit1, crit2, crit3, doubtterr. scite1, scite2, scite3, dbsource
III.Çok Yüksek Oranda Eksik Değer İçeren Öznitelikler (60 Adet)
resolution, alternative, multiple, related, attacktype2, attacktype3, weaptype2, weapsubtype2, weaptype3, weapsubtype3, weaptype4, weapsubtype4, targtype2, targsubtype2, natlty2, targtype3, targsubtype3, natlty3, gname2, gsubname2, gname3, gsubname3, motive, guncertain1, guncertain2, guncertain3, nperps, nperpcap, claimmode, claim2, claimmode2, claim3, claimmode3, compclaim, nkillus, nkillter, nwound, nwoundus, nwoundte, propextent, propvalue, nhostkid, nhostkidus, nhours, ndays, divert, kidhijcountry, ransom, ransomamt, ransomamtus, ransompaid, ransompaidus, ransomnote, hostkidoutcome, nreleased, addnotes, INT_LOG, INT_IDEO, INT_MISC, INT_ANY
IV. Çok Farklı Değerler İçeren veya Yalnızca Açıklayıcı Nitelikte Olan Öznitelikler (18 Adet)
Approxdate, summary, provstate, city, latitude, longitude, specificity, vicinity, location, weapdetail, corp1, target1, corp2, target2, corp3, target3, gsubname[2], propextent_txt

Codebook dokümanının incelenmesi sonrasında keşfedici çalışmalara veri setinde yer alan olayların incelenmesi ile devam edilmiş olup, olayı gerçekleştiren örgütün kodlanmakta olduğu "*gname*" alanında 86261 adet olay için "bilinmiyor" ifadesinin bulunduğu tespit edilmiştir. Bu tez çalışmasında, gerçekleştirilen bir saldırının hangi terör örgütü tarafından gerçekleştirildiğinin tahmin edilmesi amaçlanmakta olduğundan tespit edilen bu 86261 adet olay makine öğrenmesi algoritmalarının eğitilmesinde katkı sağlamayacak olması nedeniyle veri setinden çıkarılmıştır. Ek olarak, "*gname*" alanında terör olaylarını gerçekleştiren örgüt olarak 3616 adet farklı örgüt isminin kaydedilmiş olduğu anlaşılmıştır. Bunun üzerine, bahse konu örgütlerin gerçekleştirdikleri olay sayılarına yönelik incelemeler yapılmış olup, örgütler tarafından gerçekleştirilen olay sayılarında büyük dengesizlikler bulunmakta olduğu anlaşılmıştır. Öyle ki, 3616 adet örgütten 3046 adedi[3] 1970-2018 yılları arasında yalnızca 10 veya daha az sayıda olay gerçekleştirmiştir. Aynı tarihlerde 100

[2] Ayrıca 184990 adet eksik değer içermektedir.

[3] Toplam gerçekleştirdikleri olay sayısı 6788.

veya daha az sayıda olay gerçekleştiren örgüt[4] sayısı 3490, 1000 veya daha az sayıda olay gerçekleştiren örgüt[5] sayısı ise 3597'dir. Buna karşılık, 1000 ve üzeri olay gerçekleştiren örgüt sayısı 19 ve toplam gerçekleştirdikleri olay sayısı da 54117'dir. Dolayısıyla, veri setinde bulunan bazı örgütlerin köklü ve uzun yıllar boyunca olay gerçekleştirmekte olan örgütler olduğu bazılarının ise muhtelif tarihlerde ortaya çıkan ve köklü olduğu değerlendirilen örgütlerle ilişki içinde bulunan küçük çaplı örgütler olduğu anlaşılmıştır. Bu kapsamda, veride bulunan bahse konu dengesizlik nedeniyle, yukarıda açıklanan olay gerçekleştirme ve örgüt sayıları da dikkate alınarak makine öğrenmesi algoritmaların verimliliğin artırılması ve veri setinde amaca uygun olarak tutarlılığın sağlanabilmesi için yalnızca 1000 ve üzeri sayıda olay gerçekleştiren örgütler tarafından gerçekleştirilen olaylar veri setinde korunmuştur. Ayrıca, korunan bu örgütlerin uygunluğunun test edilebilmesi amacıyla bölüm 3.2'de açıklanan veri toplama metodolojisindeki değişiklik göz önünde bulundurularak 1998 ve 2018 yılları arasında en fazla olay gerçekleştiren örgütler de incelenmiş olup, veri setinde korunan örgütlerin bu liste ile de uyumlu olduğu görülmüştür. Böylece, veri setinin güncelliği sağlanırken küçük örgütler, en fazla olay gerçekleştiren örgütlerle ilişkili gruplar ve istisnai olaylar ile algoritmalar açısından anlam ifade etmeyecek sayıdaki olayları gerçekleştiren örgütler elimine edilmiş ve makine öğrenmesi algoritmalarının çalışma süresi ve veri setinin karmaşıklığının azaltılması amaçlanmıştır.

Diğer taraftan, makine öğrenmesi algoritmalarının güncel olayların veya yeni gerçekleşen bir olayın hangi örgüt tarafından gerçekleştirilmiş olabileceğinin tahmin edilerek araştırmaların bu noktaya yönlendirilmesi amacıyla kullanılacak olması durumunda gerilla savaşları, iç karışıklıklar, toplu katliamlar gibi şüpheli olaylar ile sona ermiş terör örgütleri tarafından gerçekleştirilmiş olayların kapsam dışı tutulması amacıyla makine öğrenmesi algoritmalarının eğitim ve test verisinden çok eski tarihli olayların çıkarılmasının uygun olacağı değerlendirilmektedir. Ancak, bu tez çalışması açısından "bilinmiyor" olarak kayıtlı olayların eski tarihlerde de bulunduğu ve terör

[4] Toplam gerçekleştirdikleri olay sayısı 21409.
[5] Toplam gerçekleştirdikleri olay sayısı 51086.

örgütlerine ilişkin yukarıda belirtildiği şekilde filtreleme yapıldığı dikkate alınarak böyle bir eliminasyona ihtiyaç duyulmamıştır.

Bu çerçevede, makine öğrenmesi modellerinin tasarımında ilk adım olarak veri temizleme sürecinin uygulanmasıyla; modellerin eğitim süresinin kısaltılması, modellerin kolay açıklanabilir ve anlaşılabilir şekilde oluşturulmasına katkı sağlanması ve modellerin eğitim verilerinde aşırı uyma (*overfitting*) probleminin önüne geçilmesi amaçlanmış ve yukarıda açıklanan işlemlerin gerçekleştirilmesi sonucunda temizlenmiş bir veri seti elde edilmiştir. Söz konusu temizlenmiş veri setine ilişkin bilgilere bölüm 3.2.2.'de yer verilmektedir.

3.3.2. Temizlenen Veri Setine İlişkin Bilgiler

GTD veri seti üzerinde bölüm 3.3.1.'de açıklanan çalışmaların yapılması sonucunda bahse konu veri setinin bir alt kümesi elde edilmiş olup bu alt küme bundan sonraki bölümlerde Tahmin Veri Seti olarak kullanılacaktır. Bu kapsamda, gelinen nokta itibariyle Tahmin Veri Seti'nde 19 adet öznitelik ve 54117 adet örnek olay bulunmaktadır.

Tahmin Veri Seti'nde yer alan özniteliklere ilişkin bilgi ve açıklamalara aşağıdaki çizelge 3.2'de yer verilmektedir. Ayrıca, 54117 adet örnek olay bulunmakta olan Tahmin Veri Seti'nde yer alan özniteliklerin tekrar incelenmesi sonrasında "*individual*" özniteliğinin yalnızca 2 adet olay için "1" olarak kodlanmış olduğunun tespit edilmesi üzerine bu öznitelik ve bu özniteliğin "1" olarak kodlandığı 2 adet örnek olay da Tahmin Veri Seti'nden çıkarılmıştır. Dolayısıyla Tahmin Veri Seti'nde bu aşamada 18 adet öznitelik ve 54115 örnek olay bulunmaktadır. İleriki bölümlerde bahsedilecek olan veri sızıntısının incelenmesi neticesinde Tahmin Veri Seti makine öğrenme algoritmaları ve öznitelik seçim yöntemleri açısından eksiksiz hale gelecektir.

Çizelge 3.2 Tahmin Veri Seti Özniteliklerine İlişkin Açıklamalar.

Değişken İsmi	Değişken Türü	Alt Kırılım	Açıklama
İyear	Nümerik	-	Olayın meydana geldiği yılı göstermektedir. Uzun bir süreye yayılan olaylarda ise olayın başlangıç yılını göstermektedir.
İmonth	Nümerik	-	Olayın meydana geldiği ayı göstermektedir. Uzun bir süreye yayılan olaylarda ise olayın başlangıç ayını göstermektedir. 1970 ile 2011 yılları arasında gerçekleşen saldırılar için, olayın tam ayı bilinmiyorsa "0" olarak kaydedilir. 2011'den sonra gerçekleşen olaylar içinse approxdate alnından elde edilen orta nokta değeri kaydedilir.
iday	Nümerik	-	Olayın meydana geldiği günü göstermektedir. Uzun bir süreye yayılan olaylarda ise olayın başlangıç gününü göstermektedir. 1970 ile 2011 yılları arasında gerçekleşen saldırılar için, olayın tam günü bilinmiyora ise "0" olarak kaydedilir. 2011'den sonra gerçekleşen olaylar içinse approxdate alnından elde edilen orta nokta değeri kaydedilir.
extended	Kategorik	2 Adet (Evet/Hayır)	Bir olayın süresinin 24 saatten fazla olup olmadığını göstermek için kullanılmakta olup, 24 saatten fazla süren olaylar için 1 kısa süren olaylar için 0 olarak kaydedilmektedir.
country	Kategorik	Codebook dokümanında 1004 adet ülke kodu yer almaktadır. Ancak Tahmin Veri Seti'nde yer olaylara bakıldığında, olayların 79 farklı ülkede gerçekleşmiş olduğu anlaşılmaktadır.	Bu alan, olayın gerçekleştiği ülkeyi veya yeri tanımlamak için kullanılmaktadır.
region	Kategorik	12 Adet	Bu alan olayın gerçekleştiği bölgeyi tanımlar. Bölgeler, olay için kodlanan ülkeye bağlı olarak belirlenmektedir.
success	Kategorik	2 Adet (Evet/Hayır)	Kategorik değişkin olan bu alan olayın başarılı olması durumunda "1" olarak kodlanmaktadır. Başarısız olması durumunda ise "0" olarak kodlanmaktadır.

Çizelge 3.2.Devam Tahmin Veri Seti Özniteliklerine İlişkin Açıklamalar.

Değişken İsmi	Değişken Türü	Alt Kırılım	Açıklama
suicide	Kategorik	2 Adet (Evet/Hayır)	Kategorik değişken olan bu alan, failin saldırıdan canlı olarak kaçmak niyetinde olmadığına dair kanıt bulunan durumlarda "1", diğer durumlarda ise "0" olarak kodlanmaktadır.
attacktype1	Kategorik	9 Adet	Kategorik değişken olan bu alan saldırı tipini, 9 kategoride kodlamak için kullanılmaktadır
weaptype1	Kategorik	13 Adet[6]	Bir saldırıda kullanılan 4 silah türü ve 4 silah alt türü her olay için kaydedilmektedir.
weapsubtype1	Kategorik	31 Adet	Kategorik değişken bu alanda, "weaptype1" alanında kodlanan silah türüne ilişkin detaylar alt kırılım olarak kodlanmaktadır.
targtype1	Kategorik	22 Adet	Her olay için hedef/mağdur bilgisi kaydedilmektedir.
targsubtype1	Kategorik	113 Adet	Kategorik değişken bu alanda, "targtype1" alanında kodlanan hedef/mağdur türüne ilişkin detaylar alt kırılım olarak kodlanmaktadır.
natlty1	Kategorik	Codebook dokümanında 1004 adet ülke kodu yer almaktadır. Ancak Tahmin Veri Seti'nde yer olaylara bakıldığında, "natlty1" alanında 111 farklı ülke olduğu anlaşılmaktadır.	Kategorik değişken olarak saldırıya uğrayan hedefin uyruğu kaydedilmektedir
gname	Metin	19 farklı örgüt ismi yer almaktadır.	Metin değişkeni olan bu alan saldırıyı gerçekleştiren grubun adını kaydetmek için kullanılmaktadır. Bölüm 3.3.1.'de belirtildiği şekilde 1000 ve üzeri sayıda olay gerçekleştiren örgütler tarafından gerçekleştirilen olaylar veri setinde korunması sonucunda bu alanda 19 farklı örgüt ismi bulunmaktadır.
individual7	Kategorik	2 Adet (Evet/Hayır)	Saldırının bir grup ya da kuruluşa bağlı olduğu bilinmeyen kişi veya kişiler tarafından gerçekleştirilip gerçekleştirilmediğini göstermek amacıyla kullanılmaktadır.

[6] "Bilinmiyor" kategorisi dahil olmak üzere kategori sayısını göstermektedir.

[7] Yalnızca 2 adet olay için "1" olarak kodlandığının anlaşılması üzerine tahmin veri setinden çıkarılmıştır.

Çizelge 3.2.Devam Tahmin Veri Seti Özniteliklerine İlişkin Açıklamalar.

Değişken İsmi	Değişken Türü	Alt Kırılım	Açıklama
claimed[8]	Kategorik	2 Adet (Evet/Hayır)	Bir grubun veya kişinin / kişilerin saldırının sorumluluğunu üstlenip üstlenmediğini belirtmek için kullanılmaktadır.
nkill	Nümerik	-	Olay için saldırgan ölümleri de dahil olmak üzere teyit edilen toplam ölümlerin sayısı kaydedilmektedir
property	Kategorik	3Adet (Evet/Hayır/Bilinmiyor)	Olaydan kaynaklanan maddi hasar olup olmadığını göstermek amacıyla kullanılmaktadır.
ishostkid	Kategorik	3Adet (Evet/Hayır/Bilinmiyor)	Mağdurların rehin alınıp alınmadığı veya kaçırılıp kaçırılmadığı bilgisi kaydedilmektedir.

Yukarıda belirtilen kriterlere bağlı olarak oluşturulan Tahmin Veri Seti'nin tam olduğu söylemek mümkün değildir. Çünkü, bu veri setinde yer alan özniteliklerden az sayıda da olsa eksik değer içerenler bulunmaktadır ki bu durum makine öğrenmesi ile ilgili oluşturulacak modellerin eğitilmesi aşamasında oldukça önemli etkiye sahiptir. Ayrıca, bahse konu veri setinde veri sızıntısı olup olmadığının değerlendirilmesi ve özniteliklerden elenebilecek olanların olup olmadığının incelenmesi gerekmektedir. Bu doğrultuda, aşağıda yer alan bölüm 3.3.3.'de öncelikle Tahmin Veri Seti'nde bulunan eksik değerlerin yerine hangi değerlerin kaydedileceği hususuna değinilmektedir.

3.3.3. Eksik Değerlerin İncelenmesi ve Doldurulması Süreci

Veri setinde eksik değerler bulunması durumunda, eksik değerlerin ele alınması açısından uygulanabilecek üç tür yaklaşım bulunmaktadır. Bunlar eksik değer içiren özniteliklerin kaldırılması, eksik değerlerin doldurulması ve eksik değerlerin özel bir şekilde kodlanmasıdır. Bu çalışmada, veri setinin temizlenmesi sürecinde çok fazla eksik değer içeren özniteliklerin veri setinden çıkarılmış olduğu ve çok fazla eksik

[8] Veride dengesizlik olmasına ragmen örgüt yapısı ve karekteristiği hanginde bilgi verebilecek olması sebebiyle ilk aşamada Tahmin Veri Seti'nde korunmuştur.

değer içermeyen özniteliklerin veri setinden çıkarılmasının bazı bilgilerin kaybolmasına sebep olabileceği dikkate alınarak Tahmin Veri Setindeki eksik değerlerin doldurulması tercih edilmiştir. Dolayısıyla, bu eksik değerler yerine hangi değerlerin kaydedilebileceği hususu araştırılmış olup, yukarıdaki çizelge 3.2'de bulunan özniteliklere ilişkin açıklamalar ve özniteliklerin özellikleri dikkate alınarak aşağıdaki çizelge 3.3'de belirtilen değerlerin bahse konu alanlara kaydedilmesi kararlaştırılmıştır.[9]

Çizelge 3.3 Eksik Verilerin Doldurulma Değerleri.

Değişken Adı	Veri Seti İndeksi	Eksik Değerin Ne Olarak Kaydedileceği
İyear	1	-
imonth	2	-
İday	3	-
extended	5	-
country	7	-
region	9	-
success	26	Nan değerler 0 olarak kaydedilecek. (Hayır)
suicide	27	Nan değerler 0 olarak kaydedilecek. (Hayır)
attacktype1	28	Nan değerler 9 olarak kaydedilecek. (Bilinmiyor)
weaptype1	81	Nan değerler 13 olarak kaydedilecek. (Bilinmiyor)
weapsubtype1	83	Nan değerler 27 olarak kaydedilecek. (Bilinmiyor)
targtype1	34	Nan değerler 20 olarak kaydedilecek. (Bilinmiyor)
targsubtype1	36	Nan değerler 114 olarak kaydedilecek. (Bilinmiyor)
natlty1	40	Nan değerler 1005 olarak kaydedilecek. (Bilinmiyor)
gname	58	-
individual	68	Nan değerler 1 olarak kaydedilecek. (Bireysel)
claimed	71	Nan değerler -9 olarak kaydedilecek. (Bilinmiyor)
Nkill	98	Nan değerler ortallama değeri olarak kaydedilecek.
property	104	Nan değerler -9 olarak kaydedilecek. (Bilinmiyor)
ishostkid	109	Nan değerler -9 olarak kaydedilecek. (Bilinmiyor)

Bu doğrultuda, uygulamada eksik değerlerin doldurulması amacıyla "verisetini_yukle_hazirla" isimli bir fonksiyon yazılmıştır. Söz konusu fonksiyonda öncelikle "*pandas*" kütüphanesinin "read_csv" metodu ile ham veri seti *dataframe* olarak uygulamaya yüklenmektedir. Veri seti yüklendikten sonra, "*scikit-learn*"

[9] Modelin geliştirilmesi ve uygulama aşamasında bahsedilen değerlerin eksik değer içeren alanlara kaydedilmesi sağlanacaktır.

kütüphanesinde bulunan "*imputer*" sınıfından oluşturulan nesneler kullanılarak ilgili özniteliklerde bulunan eksik değerler yukarıda çizelge 3.3'de belirtilen şekilde doldurularak *dataframe* güncellenmektedir. Son olarak ise, bu *dataframe* ve bu *dataframe'den* elde edilen girdi değişkenleri ile hedef değişken ayrı ayrı geri döndürülmektedir.

Bu kapsamda, Tahmin Veri Seti'nde yer alan eksik değerlerin ne şekilde doldurulacağının belirlenmesiyle Tahmin Veri Seti makine öğrenmesi modellerinin oluşturulabilmesi açısından daha eksiksiz bir hale gelmiştir.

3.3.4. Ölçeklendirilecek Özniteliklerin Belirlenmesi

Tahmin Veri Seti'nde bulunan özniteliklerin içerdiği veriler farklı aralıktadır. Bu durum, özellikle mesafeye dayalı hesaplamalar yapmakta olan algoritmalar açısından bazı özniteliklerin öne çıkmasına ve problem oluşmasına sebep olacaktır. Bu bağlamda, nümerik özniteliklere, "*sklearn.preprocessing*" sınıfından oluşturulan "*StandardScaler*" nesneleri kullanılarak Bölüm 2.3.3.'de açıklanan standardizasyon işlemi uygulanmış olup, ilgili kod parçacığı aşağıda şekil 3.1'de yer almaktadır.

```
def standardizasyon_islemini_yap (X_train, X_test, indeks):
    from sklearn.preprocessing import StandardScaler
    sc= StandardScaler()
    X_train [:, indeks:] =sc.fit_transform (X_train [:, indeks:])
    X_test [:, indeks:] =sc.transform (X_test [:, indeks:])
    return X_train, X_test
```

Şekil 3.1 Standardizasyon İşlemine İlişkin Kod Parçacığı.

Söz konusu kod parçacığından da görülebileceği üzere herhangi bir ölçeklendirme işlemi uygulanmadan önce veri setinin eğitim ve test veri setlerine ayrılmış olması gerekmektedir. Aksi durumda, makine öğrenmesi algoritmaları tüm veri setinin minimum değeri, maksimum değeri, ortalaması ve standart sapması gibi özelliklerinden bilgi sahibi olacağından veri sızıntısı oluşacaktır. Yine kod parçacığından görülebileceği üzere test veri setinin ölçeklendirilmesinde eğitim veri

setinde kullanılan parametrelerin kullanılması gerekmektedir. Bu işlem, kod parçacığında eğitim veri seti için "*fit*" metodu kullanılması ancak test veri seti için yeniden "*fit*" metodunun kullanılmaması ile yerine getirilmektedir. Özetle, "standardizasyon_islemini_yap (X_train, X_test, indeks)" fonksiyonu girdi parametresi olarak veri setleri ile hangi indekslerin standardize edileceğini almakta ve geriye standardize işlemi gerçekleştirilmiş veri setlerini döndürmektedir.

3.3.5. Kategorik Değerlerin Dönüştürülmesi

Tahmin Veri Setinde bazı önziteliklerin metin ve kategorik formatta olduğu görülmektedir. Bu formattaki özniteliklerin makine öğrenmesi algoritmaları tarafından kullanılabilmesi mümkün değildir. Dolayısıyla, bahse konu özniteliklerin, veri dönüştürme işlemi ile gerekli formata yani sayısal versiyona dönüştürülmesi gerekmektedir. Bu noktada, girdi veri setinde bulunan kategorik özniteliklere, özellikleri dikkate alınarak bölüm 2.3.4.'de belirtilen "*one hot encoder*" dönüşümü, hedef veri setindeki metin özniteliğe ise "*label encoder*" dönüşümü uygulanmıştır. Söz konusu dönüşüm işlemleri için "girdi_degiskenlerini_kodla (X, degiskenler)" fonksiyonu ve "hedef_degiskeni_kodla(y)" fonksiyonu yazılmış olup, ilgili kod parçacığı aşağıda şekil 3.2'de yer almaktadır.

```
def hedef_degiskeni_kodla(y):
    le=LabelEncoder()
    y= le.fit_transform(y)
    return y

def girdi_degiskenlerini_kodla (X, degiskenler):
    ct= ColumnTransformer (transformers=[('encoder',
                                          OneHotEncoder(categories='auto'),
                                          degiskenler)],
                           remainder= 'passthrough')
    X = (ct.fit_transform(X)).toarray()
    return X
```

Şekil 3.2 Veri Dönüştürme İşlemine İlişkin Kod Parçacığı.

Yukarıda yer alan kod parçacığından da anlaşılabileceği üzere, girdi veri setindeki kategorik değişkenlerin dönüşüm işleminin yapılabilmesi için öncelikle girdi veri setinin ve hangi özniteliklerin dönüşüme tabi tutulacağının

"girdi_degiskenlerini_kodla (X, degiskenler)" fonksiyonuna parametre olarak verilmesi gerekmektedir. Sonrasında ilgili fonksiyon, "*sklearn.compose*" kütüphanesi kullanılarak oluşturulan "*ColumnTransformer*" nesnesi ve "*sklearn.preprocessing*" kütüphanesi kullanılarak oluşturulan "*OneHotEncoder*" nesnesi vasıtasıyla yalnızca dönüştürülmesi istenilen özniteliklerin dönüşüm işlemini gerçekleştirmekte ve bu şekilde güncellenen girdi veri setini geri döndürmektedir. Hedef değişkenin dönüşümü ise benzer bir yaklaşımla "hedef_degiskeni_kodla(y)" fonksiyonunda "*sklearn.preprocessing*" kütüphanesi kullanılarak oluşturulan "*LabelEncoder*"[10] nesnesini aracılığı ile yapılmaktadır.

3.4. Özniteliklerin Belirlenmesi

Makine öğrenmesi modellerinde kullanılan veri setinde bulunan öznitelikler algoritmaların performansını doğrudan etkilemektedir. Kullanılan öznitelik sayısının yeterli olmaması durumunda algoritmalar tarafından sınıfların birbirinden tam olarak ayırt edilememesi, kullanılan öznitelik sayısının gerekenden fazla olması durumunda ise gürültülü öznitelikler sebebiyle doğruluk oranının azalması ve algoritmaların eğitim süresinin uzaması gibi problemler söz konusu olabilmektedir. Ayrıca, veri örneklerine kıyasla öznitelik sayısı fazlaysa eğitim aşamasında elde edilen iyi performansa karşılık uygulama esnasında çok kötü performans elde edilmesine sebep olan aşırı öğrenme problemi de meydana gelebilmektedir. Dolayısıyla, her ne kadar rastgele orman algoritması gibi bazı algoritmalar aşırı öğrenme problemine karşı dayanıklı olsalar dahi modeller geliştirilmeden önce öznitelik seçimi yapılması ihtiyacının olup olmadığının değerlendirilmesi yerinde olacaktır.

Bu çerçevede, daha önceki bölümlerde açıklandığı üzere veri temizleme işlemlerinden sonra oluşturulan Tahmin Veri Seti'nde 18 adet öznitelik bulunmakta olup Tahmin Veri Setinde bulunan özniteliklerin hedef değişkenin tahmin edilmesine bulundukları katkı açısından değerlendirilmesi ve elimine edilmesi gereken öznitelik

[10] "OneHotEncoder" ve "LabelEncoder" arasındaki farklılıklardan bölüm 2.3.4.'de bahsedilmektedir.

bulunup bulunmadığına karar verilmesi için bu tez çalışmasında permütasyon ve sütun kaldırma yöntemleri ile rastgele orman algoritmasının birlikte kullanıldığı bir yaklaşım ile öznitelikler incelenmiştir. Ancak özniteliklerin seçim aşamasına geçilmeden önce Tahmin Veri Seti'nde yer alan özniteliklerden kaynaklı veri sızıntısı olup olmadığının incelenmesi gerekmiştir. Çünkü gerek permütasyon yöntemi gerekse de sütun kaldırma yöntemi ile özniteliklerin belirlenmesinde özniteliklerin önem derecesi dikkate alınmaktadır. Dolayısıyla, veri sızıntısı bulunması durumunda ilgili öznitelikler önemli görünecek ve öznitelik seçiminin hatalı olması söz konusu olabilecektir. Ayrıca, bölüm 2.5.'de de belirtildiği üzere her iki öznitelik seçim yönteminde de her öznitelik ayrı ayrı ele alınmaktadır. Bu, her özniteliğin birbirinden bağımsız olması ve aralarında korelasyon bulunmaması durumunda bir problem oluşturmayacaktır. Ancak, veri setinde aralarında korelasyon bulunan özniteliklerin bulunması halinde bahse konu yöntemler beklenmedik sonuçlar verebilecektir. Dolayısıyla, veri setinde bulunan özniteliklerin arasında korelasyon bulunup bulunmadığının da öznitelik seçim aşamasına gelmeden önce incelenmesi gerekmektedir. Esas itibariyle veri sızıntısının ve öznitelikler arasında korelasyon bulunup bulunmadığının incelenmesi de bir nevi öznitelik seçimidir. Bu çerçevede, bu tez çalışmasına konu uygulamada da gerek hedef değişkeni doğrudan tahmin etme niteliği olan özniteliklerin olup olmadığının belirlenmesi gerekse de birbirleri ile yüksek oranda korelasyon bulunmasından dolayı fazla olarak değerlendirilebilecek özniteliklerin belirlenmesi amacıyla özniteliklerin bağımlılıkları incelenmiştir.

Bir x özniteliğinin diğer özniteliklere bağımlı olup olmadığının belirlenmesinin bir yolu makine öğrenmesi algoritmaları ile bir model geliştirerek bu modelde x'in hedef değişken ve diğer tüm özniteliklerin girdi değişkeni olarak kullanılmasıdır. Bu şekilde öznitelikler arasındaki bağımlılıklar belirlenebilecek olup, geliştirilecek modelde algoritma olarak ise OOB örneklerini sağlaması, doğrulama setleri oluşturulması gerekliliğini ortadan kaldırması ve hata tahmini sağlamasından dolayı rastgele orman algoritmasının kullanılması uygun olacaktır. Bu tez çalışmasına konu uygulamada da özniteliklerin bağımlılıklarının belirlenmesi amacıyla bahsedilen şekilde bir süreç uygulanmış olup, bu amaç doğrultusunda yapılan işlemlerin temelinde yer alan kod parçacığı aşağıda şekil 3.3'de verilmektedir.

```
def bagimlilik_matrisi_hesapla_goruntule (X_indeksler, ornekSayi):
    dataset = veriyi_yukle_korelasyon()
    X= dataset.iloc[:, 1:]
    X_kalanlar = X.iloc[:, X_indeksler]
    rf_classifier = RandomForestClassifier (n_estimators =100,
                                            criterion='entropy',
                                            random_state= 999,
                                            oob_score=True,
                                            bootstrap= True, n_jobs=-1)
    dp= oznitelik_bagimlilik_matrisi_hesapla(X_kalanlar,
                                             model_rf=rf_classifier,
                                             n_ornek=ornekSayi)
    dp.sort_values(by='Dependence', ascending=False)

    bagimlilik_matrisi= ciz_bagimlilik_matrisi(dp)
    bagimlilik_matrisi.view()
    return dp
```

Şekil 3.3 Öznitelik Bağımlılığının İncelenmesine İlişkin Kod Parçacığı.

Yukarıda şekil 3.3'de yer alan kod parçacığından da görülebileceği üzere öznitelikler arasındaki bağımlılıkların belirlenebilmesi için "bagimlilik_matrisi_hesapla_goruntule(X_indeksler, ornekSayi)" fonksiyonu yazılmıştır. Söz konusu fonksiyonunun temelinde parametre olarak verilen eğitim veri setindeki her bir özniteliği hedef değişken olarak kullanılarak yine parametre olarak verilen rastgele orman algoritması üzerinden özniteliklerin öneminin "*scikit-learn*" kütüphanesinin rastgele orman algoritması için olan OOB skoruna göre hesaplanması bulunmaktadır. Bahse konu fonksiyon tarafından belirtilen bu işlem ise yazılmış olan "oznitelik_bagimlilik_matrisi_hesapla" fonksiyonu ile yerine getirilmektedir. Sonuç olarak, "bagimlilik_matrisi_hesapla_goruntule(X_indeksler, ornekSayi)" fonksiyonu geriye simetrik olmayan bir bağımlılık matrisi döndürmektedir. Bu matriste, her bir satırda hedef değişken olarak belirlenmiş özniteliğin diğer öznitelikler tarafından tahmin edilmesindeki öznitelik önem dereceleri yani bağımlılık sayısı gösterilmektedir. Dolayısıyla, bağımlılık sayısının bir olması durumunda ilgili özniteliğin diğer öznitelikler kullanılarak doğrudan öngörülebilir olduğu yani doğruluk oranını etkilemeden kaldırılabileceği sonucuna ulaşılmaktadır.

Bu kapsamda, özniteliklerin belirlenmesi ve ilgisiz özniteliklerin elimine edilmesi aşamasına geçilmeden önce bölüm 2.4.'de açıklandığı gibi modelin, hedef değişkenle doğrudan etkileşime giren öznitelikleri içermesi durumunda veri sızıntısı

oluşacağı dikkate alınarak yukarıda çalışma prensibi açıklanan "bagimlilik_matrisi_hesapla_goruntule(X_indeksler, ornekSayi)" fonksiyonu ile öncelikle hedef değişken ile diğer öznitelikler arasındaki korelasyonun incelenmiştir. Uygulamada, hedef değişken ile korelasyon içeren değişkenlerin bulunması işlemi "hedef_verisizintisi_degiskenleri_yazdir()" fonksiyonu ile başlamaktadır. Söz konusu fonksiyon tarafından hedef değişkenin de dahil olduğu tüm değişkenler ile "hedef_korelasyonu_degiskenleri_bul (X_indeksler)" fonksiyonunu çağırmaktadır. Bu fonksiyon ise "bagimlilik_matrisi_hesapla_goruntule(X_indeksler, ornekSayi)" fonksiyonunu döngü içerisinde 10 defa çağırarak her bir değişkenin hedef değişkene olan bağımlılığını hesaplamakta olan "hedef_korelasyonu_bul(X_indeksler=[])" fonksiyonunu çağırmaktadır. Sonuç olarak, işlemlerin başlangıç noktası olan "hedef_verisizintisi_degiskenleri_yazdir()" fonksiyonuna her bir değişken açısından hedef değişkenin ""bagimlilik_matrisi_hesapla_goruntule(X_indeksler, ornekSayi)" fonksiyon 10 kere çağırıldığından 10 üzerinden tahmin edilme oranı ve elenecek indeksler döndürülmektedir. Elenecek indeksler ise tahmin edilme oranının 3'ten büyük olması kriterine göre belirlenen "hedef_korelasyonu_indeksleri_bul" fonksiyonunda belirlenmektedir. Hedef değişkenle veri sızıntısı olabilecek şekilde korelasyon içeren özniteliklerin belirlenmesine yönelik kullanılan ve yukarıda açıklanmış olan fonksiyonlar aşağıda Şekil 3.4'de verilmektedir.

```
def hedef_korelasyonu_bul(X_indeksler=[]):
    sonuc=[0.0]*X_indeksler.__len__()
    for number in range(10):
        dp=bagimlilik_matrisi_hesapla_goruntule(X_indeksler=X_indeksler,
                                                ornekSayi = 5000)
        dp_gname=dp['gname']
        for i in range(X_indeksler.__len__()-1):
            sonuc[i] = sonuc[i] + dp_gname[i]
    return sonuc

def hedef_korelasyonu_indeksleri_bul(hedefTahminedilme):
    elenecekListe=[]
    for i in range(hedefTahminedilme.__len__()):
        if(hedefTahminedilme[i]>3):
            elenecekListe.append(i)
    return elenecekListe
```

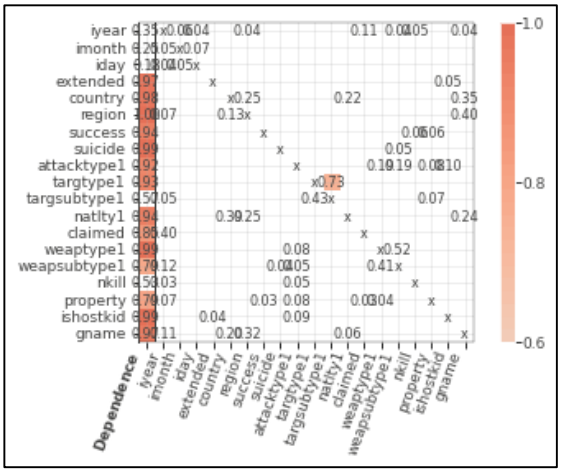
Şekil 3.4 Veri Sızıntısı Bulunması İşlemine İlişkin Kod Parçacığı.

```
def hedef_korelasyonu_degiskenleri_bul(X_indeksler):
    hedefTahminEdilme = hedef_korelasyonu_bul(X_indeksler)
    elenecekİndeksler = hedef_korelasyonu_indeksleri_bul(hedefTahminEdilme)
    elecenekX_indeksler=[]
    for i in range(elenecekİndeksler.__len__()  ):
        elecenekX_indeksler.append(X_indeksler[elenecekİndeksler[i]])

    return elecenekX_indeksler, hedefTahminEdilme
```

```
def hedef_verisizintisi_degiskenleri_yazdir():
    print("\n***ZAMAN DEĞİŞKENLERİ KALDIRILMADAN ÖNCE 18 DEĞİŞKEN HEDEF TAHMİN EDİLME ORANI HESAPLAMA BAŞLANGIÇ***\n")
    eleneceklerX_18Deg,hedefTahminEdilmeOranToplam = hedef_korelasyonu_degiskenleri_bul([0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18])
    print(eleneceklerX_18Deg)
    print("\n***18 DEĞİŞKEN- KALDIRILAN DEĞİŞKENLER İÇİN DOĞRULUK ORANI HESAPLAMA BAŞLANGIÇ***\n")
    dogrulukhesapla_ozniteliksecim(pcaDeger=0, cVal =10, ortalama = 'macro',öznitelikListe=eleneceklerX_18Deg)
    print("\n***18 DEĞİŞKEN- KALDIRILAN DEĞİŞKENLER İÇİN DOĞRULUK ORANI HESAPLAMA SON***\n")
    print("\n***ZAMAN DEĞİŞKENLERİ KALDIRILMADAN ÖNCE 18 DEĞİŞKEN HEDEF TAHMİN EDİLME ORANI HESAPLAMA SON***\n")

    print("\n***ZAMAN DEĞİŞKENLERİ KALDIRILDIKTAN SONRA 15 DEĞİŞKEN HEDEF TAHMİN EDİLME ORANI HESAPLAMA BAŞLANGIÇ***\n")
    eleneceklerX_15Deg, hedefTahminEdilmeOranToplam2 = hedef_korelasyonu_degiskenleri_bul([3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18])
    print(eleneceklerX_15Deg)
    print("\n***15 DEĞİŞKEN- KALDIRILAN DEĞİŞKENLER İÇİN DOĞRULUK ORANI HESAPLAMA BAŞLANGIÇ***\n")
    dogrulukhesapla_ozniteliksecim(pcaDeger=0, cVal =10, ortalama = 'macro',öznitelikListe=eleneceklerX_15Deg)
    print("\n***15 DEĞİŞKEN- KALDIRILAN DEĞİŞKENLER İÇİN DOĞRULUK ORANI HESAPLAMA SON***\n")
    print("\n***ZAMAN DEĞİŞKENLERİ KALDIRILDIKTAN SONRA 15 DEĞİŞKEN HEDEF TAHMİN EDİLME ORANI HESAPLAMA SON***\n")
```

Şekil 3.4.Devam. Veri Sızıntısı Bulunması İşlemine İlişkin Kod Parçacığı

Bu doğrultuda, terör gruplarının tahmin edilmesi amacıyla geliştirilen bir modelde, doğrudan bir değişkene bağlı olarak bir model geliştirilmesi gerçek dünya uygulamasında elde bulunacak diğer verilerin önüne geçecek ve modelin etkinliğini azaltacaktır. Dolayısıyla, uygulamada bir özniteliğin gerçek dünya uygulamasından uzaklaşarak hedef değişkeni tahmin edebilmesi durumunun önüne geçilmesi ve veri sızıntısının tespiti amacıyla "hedef_verisizintisi_degiskenleri_yazdir()" fonksiyonu çalıştırılmıştır. Sonuç olarak "*country*" ve "*region*" değişkenlerinin hedef değişken "*gname*" ile yüksek oranda korelasyon içerdiği tespit edilerek ilgili öznitelikler Tahmin Veri Seti'nden elenmiştir. Söz konusu yüksek korelasyonunu gösterir bir örnek aşağıda şekil 3.5'de sunulmaktadır.



Şekil 3.5 Hedef Değişken ile Yüksek Korelasyon İçeren Değişkenler.

Ayrıca, Tahmin Veri Seti'nde yer alan zaman özniteliklerinin (*iyear,imonth,iday*), özellikle olayın gerçekleştiği yılı göstermekte olan "*iyear*" özniteliğinin, diğer özniteliklere kıyasla baskın olduğu tespit edilmiştir. Ancak gerçekleşen bir terör olayından sonra incelemelerin etkin bir şekilde yürütülebilmesi adına sorumlu olabilecek grupların tahmin edilebilmesine yönelik bir makine öğrenmesi modelinin geliştirilmesinde zaman değişkenlerinden ziyade terör örgütlerinin özellikleri ve yöntemleri gibi unsurlara dayalı modeller tasarlanması daha uygun olacaktır. Bu doğrultuda, belirli faktörlere göre tahmin yapılması olayının önüne geçilerek zamana bağlı bir tahmin yapılması sonucunu ortaya çıkmasının önlenmesi açısından "*iyear*", "*imonth*" ve "*iday*" zaman değişkenleri de Tahmin Veri Seti'nden elimine edilmiştir. Bu durumun ortaya çıkması üzerine, 3 adet zaman değişkeni hariç tutulup Tahmin Veri Seti'nde bulunan 15 adet öznitelik ile hedef değişkenin doğrudan tahmin edilmesi hususu tekrardan incelenmiş ve "hedef_verisizintisi_degiskenleri_yazdir()" fonksiyonu yeniden çağrılmış olup, "*country*" ve "*region*" değişkenlerinin hedef değişkenle yüksek oranda korelasyon içermekte olduğu tespitinin değişmediği görülmüştür. Dolayısıyla, bahse konu özniteliklerin elimine edilmesiyle makine öğrenmesi algoritmaları açısından kullanılabilecek ve algoritmaların değerlendirilmesi ile öznitelik seçim aşamalarının başlangıç noktası olarak kabul edilebilecek 13 özniteliğe sahip Temel Veri Seti elde edilmiştir. Temel Veri Seti yukarıda çizelge 3.2'de verilen Tahmin Veri Seti'nden "*country*", "*region*" ,"*iyear*", "*imonth*" ve "*iday*" özniteliklerin elimine edilmesi sonrasında kalan 13 adet öznitelikten oluşmaktadır.

Makine öğrenmesi modellerinin geliştirilmesi ve test edilmesi aşamalarında kullanılabilecek olan Temel Veri Seti'nin elde edilmesi sonrasında, makine öğrenmesi modellerinin geliştirilmesi aşamasında önemli bir adım olan öznitelik seçim aşamasına geçilmiştir. Bu aşamada, çalışma mantığı bölüm 2.5.'de anlatılmış olan permütasyon ve sütun kaldırma yöntemleri uygulanmıştır. Öncelikle bahse konu yöntemler kullanılarak her bir özniteliğin önem derecesi hesaplanmıştır. Ancak, aşağıda şekil 3.6'da da görülebileceği üzere permütasyon ve sütun kaldırma yöntemlerinde özniteliklerin önem sırası birbirinden oldukça farklı çıkmıştır. Ek olarak, söz konusu yöntemler tekrar tekrar çalıştırıldığında tutarsız sonuçlar elde edilmiştir. Bu durumun,

yine bölüm 2.5.'de bahse konu yöntemlere ilişkin bilgiler verilirken açıklanmış olan öznitelikler arasında korelasyon bulunmasından kaynaklandığı anlaşılmıştır.

Sutun Kaldırma Onem		Permutasyon Onem	
natlty1	0.487896	natlty1	0.447704
claimed	0.012866	claimed	0.005082
targsubtype1	0.006837	targsubtype1	0.001825
weapsubtype1	0.004019	suicide	-0.000693
property	0.003349	weapsubtype1	-0.001155
nkill	0.003072	success	-0.002217
suicide	0.001432	ishostkid	-0.002472
attacktype1	0.001016	weaptype1	-0.002795
success	0.000346	property	-0.002980
targtype1	0.000139	targtype1	-0.003165
extended	-0.000624	attacktype1	-0.003465
ishostkid	-0.000901	extended	-0.003927
weaptype1	-0.002056	nkill	-0.004712

Şekil 3.6 Özniteliklerin Önem Dereceleri Arasındaki Tutarsızlıklar.

Bu doğrultuda, birbiriyle yüksek oranda korelasyon içeren öznitelikler arasında seçim yapılabilmesi adına "bagli_degiskenleri_kaldir(X_kalanlar)" fonksiyonu yazılmıştır. Söz konusu fonksiyona ilişkin kod parçacığı aşağıda şekil 3.7'de verilmektedir. İlgili kod parçacığından da görülebileceği üzere söz konusu fonksiyonda öncelikle yukarıda veri sızıntısının tespit edilmesi aşamasında da detayları açıklanan yaklaşım yani "hedef_verisizintisi_degiskenleri_yazdir()" fonksiyonu ile her bir değişken ile diğer öznitelikler arasındaki korelasyonun incelenmesi yapılmakta olup, buradaki farklılık veri setinde hedef değişkene yer verilmemesidir. Sonrasında ise bağımlılık yani diğer değişkenler tarafından tahmin edilme oranına göre öznitelikler sıralanmış ve en yüksek tahmin edilme oranına sahip özniteliğin oranının 1 üzerinden 0,5'ten büyük olup olmadığı kontrol edilmiştir. Söz konusu oranın 0,5'ten büyük olması durumunda ilgili öznitelik elimine edilerek kalan özniteliklerle "bagli_degiskenleri_kaldir(X_kalanlar)" fonksiyonu kendi kendine tekrardan çağırmaktadır. Bu döngüye tahmin edilme oranı 0,5'ten büyük öznitelik kalmayana kadar özyinelemeli şekilde devam edilmektedir. İşlemlerin tamamlanması sonrasında ise geriye, kalan öznitelikler döndürülmektedir. Bu şekilde aralarında korelasyon bulunan öznitelikler elendiğinden dolayı doğruluk oranında çok büyük bir azalma olması beklenmemektedir.

```
def bagli_degiskenleri_kaldir(X_kalanlar):
    dp=bagimlilik_matrisi_hesapla_goruntule(X_indeksler=X_kalanlar,
                                            ornekSayi = 5000)
    max_satir=dp[dp['Dependence']==dp['Dependence'].max()]
    max_öznitelik = str(max_satir.index.values[0])
    max_deger=max_satir['Dependence'].values[0]

    if max_deger < 0.5:
        return X_kalanlar
    else:

        if max_öznitelik == 'iyear':
            silinecek = 0
            print ("iyear Kaldırılıyor")

        if max_öznitelik == 'imonth':
            silinecek = 1
            print ("imonth Kaldırılıyor")
            ...
        ...
        if max_öznitelik == 'ishostkid':
            silinecek = 17
            print ("ishostkid Kaldırılıyor")

        X_kalanlar.remove(silinecek)

        return bagli_degiskenleri_kaldir(X_kalanlar)
```

Şekil 3.7 Bağlı Değişkenlerin Elenmesine İlişkin Kod Parçacığı.

Son olarak, öznitelik seçimi işleminin tamamlanması amacıyla "oznitelikbelirle_korelasyonelimine_onem()" fonksiyonu yazılmış olup, bu fonksiyon aralarında korelasyon bulunan öznitelikleri yukarıda açıklanan şekilde elimine etmekte ve özniteliklerin permütasyon ve sütun kaldırma yöntemleri ile hesaplanan önem derecelerini konsola yazdırmaktadır. Bahse konu fonksiyon öncelikle 18 adet öznitelik bulunan Tahmin Veri Seti için önem dereceleri ekrana yazdırılmakta daha sonra 13 adet öznitelik bulunan Temel Veri Seti için önem dereceleri ekrana yazdırılmaktadır. Sonrasında ise her iki veri seti için yukarıda detaylandırılan yöntemle eliminasyon işlemi gerçekleştirilmekte ve son olarak kalan öznitelikler ve önem dereceleri ekrana yazdırılmaktadır.

Söz konusu "oznitelikbelirle_korelasyonelimine_onem()" fonksiyonunun çalıştırılması sonucunda elde edilen bulgular değerlendirildiğinde, eliminasyon işlemi öncesinde her iki veri setin için de permütasyon ve sütün kaldırma yöntemlerinin tutarsız sonuçlar verdiği anlaşılmıştır. Ayrıca, her iki veri setinin eliminasyona tabi

tutulması sonucunda; Tahmin Veri Seti'nden geriye 3 adedi zaman değişkenleri olmak üzere 7 adet özniteliği kaldığı, Temel Veri Seti'nden geriye ise zaman değişkenleri haricindeki 4 adet aynı özniteliğin kaldığı görülmüştür. Dolayısıyla, zaman değişkenlerinin veri sızıntısı nedeniyle Temel Veri Seti yer almadığı dikkate alındığında her iki veri setinde aynı özniteliklerin önemli olduğu sonucuna ulaşılmış olup, veri sızıntısı işleminin yerinde olduğu anlaşılmıştır. Bahse konu 4 adet öznitelikten ise önem derecelerine göre 1 adet özniteliğin ("*nkill*") terör olaylarına ilişkin veri seti hakkında çok az bilgiyi içermekte olduğu görülmüştür. Dolayısıyla, öznitelik seçimi işlemi sonucunda, geriye kalan 3 adet özniteliğin seçilmesine karar verilmiş ve önem derecelerine ilişkin tutarsızlığın ortadan kalktığı gözlemlenmiştir. Bu kapsamda, 3 adet öznitelik bulunan veri seti (Nihai Veri Seti[11]) ve 13 adet öznitelik bulunan Temel Veri Seti karşılaştırmalı olarak makine öğrenmesi modellerinde kullanılacak ve sonuçları gözlemlenecektir. Diğer taraftan, 7 adet özniteliğe sahip veri seti (Zaman Değişkenli Veri Seti[12]) zaman değişkenlerine ilişkin veri sızıntısının gösterilmesi, 4 adet özniteliğe sahip veri seti (Belirlenen Veri Seti[13]) ise öznitelik önemi kapsamında yapılan eliminasyonun etkisinin değerlendirilmesi amacıyla kullanılacaktır.

3.5. Modellerin Eğitilmesi ve Test Edilmesi

Makine öğrenmesi modellerin oluşturulması sürecinde veri seti olarak bölüm 3.5'de açıklanan Temel Veri Seti, Zaman Değişkenli Veri Seti, Belirlenen Veri Seti ve Nihai Veri Seti kullanılmıştır. Dolayısıyla, algoritmaların performansının değerlendirilmesinin yanında veri sızıntısı ve öznitelik seçim aşamalarının etkinliğinin gözlemlenmesi de mümkün olmuştur.

Bu doğrultuluda, her bir algoritmanın eğitim ve test aşamasından aşağıda bahsedilmekle birlikte genel olarak eğitim ve test işlemi "*sklearn*" kütüphanesinin her

[11] Bulunan öznitelikler: *natlty1,weapsubtype1, targsubtype1,*
[12] Bulunan öznitelikler: *iyear, imonth, iday, natlty1,weapsubtype1, targsubtype1, nkill*
[13] Bulunan öznitelikler: *natlty1,weapsubtype1, targsubtype1, nkill*

bir algoritma ile ilgili modülü üzerinden gerçekleştirilmiştir. Söz konusu modüller üzerinden makine öğrenmesi algoritmalarının eğitim işleminin gerçekleştirilebilmesi için öncelikle aşağıda şekil 3.8'de gösterilen "algoritma_hazirla_getir(algoritmaAdi, X_train, y_train)" fonksiyonu yazılmıştır.

```
def algoritma_hazirla_getir(algoritmaAdi, X_train, y_train):

    if algoritmaAdi == 'RF':
        classifier = RandomForestClassifier (n_estimators =200, criterion='gini', random_state=999)
        classifier.fit(X_train,y_train)
    elif algoritmaAdi == 'KNN':
        classifier = KNeighborsClassifier(n_neighbors=10, metric ='minkowski', p =2)
        classifier.fit(X_train,y_train)
    elif algoritmaAdi == 'LR':
        classifier = LogisticRegression(C= 1, random_state=999)
        classifier.fit(X_train,y_train)
    elif algoritmaAdi == 'SVM':
        classifier = SVC(kernel='linear', C=1, probability= True, random_state=999)
        classifier.fit(X_train,y_train)
    elif algoritmaAdi == 'KSVM':
        classifier = SVC(kernel='rbf', C=1, gamma=0.5, probability= True, random_state=999)
        classifier.fit(X_train,y_train)
    elif algoritmaAdi == 'NB':
        classifier = GaussianNB()
        classifier.fit(X_train,y_train)
    elif algoritmaAdi == 'DT':
        classifier = DecisionTreeClassifier(criterion='entropy', random_state=999)
        classifier.fit(X_train,y_train)
    elif algoritmaAdi == 'ANN':
        classifier = KerasClassifier(build_fn=model_getir_ANN, epochs=50, batch_size=128, verbose=0)
        classifier.fit(X_train, y_train)
    else:
        classifier = RandomForestClassifier (n_estimators =200, criterion='entropy', random_state=999)
        classifier.fit(X_train,y_train)

    return classifier
```

Şekil 3.8 Algoritmaların Eğitilmesine İlişkin Kod Parçacığı.

Yukarıda yer alan kod parçacığından da görülebileceği üzere söz konusu fonksiyon parametre olarak metin değişkeni olan algoritma adını ve eğitim veri setini almaktadır. Eğitim veri setinin girdi değişkenleri "X_train", hedef değişken ise "y_train" şeklinde fonksiyona parametre olarak iletilmektedir. Bahse konu fonksiyon geriye ise algoritma adına göre oluşturduğu ve eğitim veri seti ile eğitim işlemini gerçekleştirdiği ilgili algoritmaya ait sınıflandırıcı nesnesini dönmektedir.

3.5.1. Modellerin Eğitilmesine İlişkin Genel Yapı

Bu tez çalışmasına konu uygulamada, modellerin geliştirilmesinde kullanılan ve karşılaştırmaları yapılacak olan algoritmaların eğitilmesi sürecinde benzer işlemler yapılmış olduğundan öncelikle bu işlemlere ilişkin genel yapıdan bahsedilmesi uygun olacaktır. Modellerin eğitilme sürecinin temelinde "algoritma_hazirla_getir (algoritmaAdi,X_train, y_train)" fonksiyonu bulunmakta olup, söz konusu fonksiyon yukarıda da bahsedildiği üzere parametre olarak verilen algoritma adına göre oluşturduğu ve eğitim veri seti ile eğitim işlemini gerçekleştirdiği ilgili algoritmaya ait sınıflandırıcının nesnesini geriye döndürmektedir. Modellerin eğitilmesine sürecinde bahse konu fonksiyon, her bir algoritma için ayrı olacak şekilde yazılmış olan fonksiyonlardan çağrılmaktadır. Her bir algoritma için yazılan bu fonksiyonların bir örneğine aşağıda şekil 3.9'da yer verilmektedir.

```
def modeli_calistir_sonuclari_getir_KNN_13Deg(pcaDeger=0, cVal =0, ortalama = 'macro'):
    print("\n***13 DEĞİŞKEN İÇİN KNN ALGORİTMASI ÇALIŞTIRILIYOR.***\n")
    print("\n *** PARAMETRE AYARLI***\n")
    print("\n *** PCA DEĞERİ",pcaDeger, "***\n")
    ilk =sure_olcmeye_basla()
    dataset, X_train, X_test, y_train, y_test = islemleri_tamamla_verisetlerini_getir_13Deg()
    if pcaDeger>0:
        X_train, X_test = boyut_azalt_PCA(X_train, X_test,pcaDeger)
    classifier = algoritma_hazirla_getir('KNN', X_train, y_train)
    y_pred = classifier.predict(X_test)
    y_pred_proba = classifier.predict_proba(X_test)
    sure = calisma_suresi_getir(ilk)
    if ortalama == 'macro':
        performans_degerlendirme_getir_macro(y_test, y_pred,y_pred_proba,sure)
    elif ortalama == 'micro':
        performans_degerlendirme_getir_micro(y_test, y_pred,y_pred_proba,sure)
    if cVal>0:
        caprazdogrulama_skor_hesapla_yazdir_10(classifier, X_train, y_train)
    print("\n***13 DEĞİŞKEN İÇİN KNN ALGORİTMASI TAMAMLANDI.***\n")
```

Şekil 3.9 Eğitim İşleminin Uygulanması İlişkin Kod Parçacığı.

Yukarıda yer alan kod parçacığından da görülebileceği üzere k-en yakın komşu algoritmasının eğitilmesi ve test edilerek sonuçlarının performans kriterlerine göre değerlendirilmesi amacıyla yazılan "modeli_calistir_sonuclari_getir_KNN_13Deg (pcaDeger=0, cVal =0, ortalama = 'macro')" fonksiyonu öncelikle algoritmaların eğitim ve test süresinin ölçülmesi için yazılmış olan "sure_olcmeye_basla()" fonksiyonu aracılığı ile süre sayacını başlatmakta; "islemleri_tamamla_verisetlerini _getir_13Deg()" fonksiyonunu kullanarak da algoritmaların eğitileceği eğitim veri

setini ve algoritmaların elde edeceği sonuçların kontrol edileceği test veri setini oluşturmaktadır. Veri setlerini oluşturmakta olan "islemleri_tamamla_verisetlerini _getir_13Deg()" fonksiyonu ise veri setini dosyadan yükleyerek 5'e 1 olacak şekilde eğitim ve test veri setlerine ayırmakta ve söz konusu veri setlerini veri ön işleme süreçlerini tamamlayarak geriye döndürmektedir. Bu şekilde hazır hale getirilmiş veri setlerinin elde edilmesinden sonra "modeli_calistir_sonuclari_getir_KNN_13Deg" fonksiyonu kendisine parametre olarak verilen "pcaDeger" sayısını kontrol etmekte ve eğer bu sayı 0'dan büyükse söz konusu veri setlerine "pcaDeger" değerine göre boyut azaltma (PCA) işlemini "boyut_azalt_PCA(X_train,X_test,pcaDeger)" fonksiyonu vasıtasıyla uygulamaktadır. Böylece algoritmalar tarafından kullanılabilecek formatta veri seti elde edilmesi ile ilgili işlemler tamamlanmakta ve ilgili algoritma adı ile birlikte söz konusu veri setleri parametre olarak verilerek yukarıda detayları açıklanmış olan "algoritma_hazirla_getir(algoritmaAdi,X_train, y_train)" fonksiyonu çağırılmaktadır. Bunun sonu olarak ise sınıflandırıcı model elde edilmekte olup, ilgili modelin "*predict*" metodunun test verisi ile çağrılmasıyla da tahmin sonuçları elde edilmektedir. Bu işlem sonrasında "calisma_suresi_getir(ilk)" fonksiyonu ile çalışma süresi elde edilmektedir ki bu fonksiyona parametre olarak yukarıda açıklanan ve işlemlerin başında başlatılan sayaç verilmektedir. Söz konusu ilk sayaç değeri ile "calisma_suresi_getir(ilk)" fonksiyonunun çağırıldığı an arasındaki fark ise modelin eğitim ve test işlemlerine harcanan süreyi göstermektedir.

Yukarıda belirtildiği şekilde test işleminin tamamlanıp tahminlerin elde edilmesinden sonra söz konusu tahminlere göre algoritmaların performanslarının değerlendirilmesi aşamasına geçilmektedir. Performans değerlendirme işleminin yerine getirilmesi içinse "modeli_calistir_sonuclari_getir_KNN_13Deg" fonksiyonu tarafından kendisine iletilen "ortalama" parametresinin değeri kontrol edilerek "performans_degerlendirme_getir_macro" veya "performans_degerlendirme_getir _micro" fonksiyonlarından bir tanesi çağrılmaktadır. Esas itibariyle bu fonksiyonların her ikisi de öncelikle hata matrisini yazdırmakta ve "*sklearn*" kütüphanesini kullanarak bölüm 2.6'da belirtilen performans metriklerini hesaplayıp konsola yazdırmaktadır. Ancak hemen belirtmek gerekir ki "*sklearn*" kütüphanesinin performans metriklerine bakıldığında, mikro ortalamanın toplam doğru pozitifleri, yanlış negatifleri ve yanlış

pozitifleri sayarak hesaplama yaptığı görülmektedir. Dolayısıyla, çok sınıflı problemlerde, yalnızca bir sınıfa yönelik tahminde bulunulduğundan, yanlış negatif ve yanlış pozitif sayısı eşit olacaktır ki bu durum da hassasiyet, duyarlılık ve f1 skorunun eşit olmasına neden olacaktır. Bu durum, uygulamada da test edilerek gözlemlenmiştir. Bu sebeple, her bir sınıf için ölçütlerin hesaplanması ve elde edilen değerlerin herhangi bir ağırlıklandırma olmadan ortalamasının alınması şeklinde çalışmakta olan makro ortalama yöntemi performans metriklerinin hesaplanmasında tercih edilmiş olup anlatılan performans değerlendirme sürecine ilişkin örnek kod parçacığı aşağıda şekil 3.10'da verilmektedir.

```
def performans_skorlarini_yazdir_micro(y_test, y_pred,y_pred_proba,sure):
    print("\n***Skorlar Yazdırılıyor. (Micro)***\n")
    print(" ")
    ac=accuracy_score(y_test, y_pred)
    pre=precision_score(y_test, y_pred, average='micro')
    rec=recall_score(y_test, y_pred,average='micro')
    f1=f1_score(y_test, y_pred,average='micro')
    print("Doğruluk: {:.6f} %".format(ac))
    print("Hassasiyet: {:.6f} %".format(pre))
    print("Duyarlılık: {:.6f} %".format(rec))
    print("F1 Skoru: {:.6f} %".format(f1))
    print("Süre: " + sure)
    print("\n***Skorlar Yazdırıldı. (Micro)***\n")

    metrikraporu_auc = skor_hesapla(
    y_dogru=y_test,
    y_tahmin= y_pred,
    y_olasilik= y_pred_proba, ortalama='micro')
    print(metrikraporu_auc)

def performans_degerlendirme_getir_macro(y_test, y_pred,y_pred_proba,sure):
    hata_matrisini_yazdir(y_test, y_pred)
    performans_skorlarini_yazdir_macro(y_test, y_pred,y_pred_proba,sure)
```

Şekil 3.10 Performans Değerlendirilmesine İlişkin Kod Parçacığı.

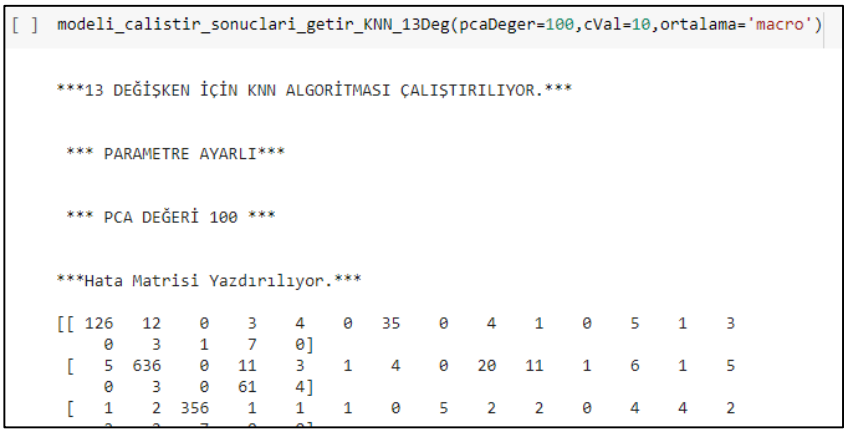
Son olarak ise "modeli_calistir_sonuclari_getir_KNN_13Deg" fonksiyonunda, "algoritma_hazirla_getir" fonksiyonu aracılığıyla yukarıda açıklanan şekilde elde edilen sınıflandırıcı nesnesi ve veri setleri kullanılarak aşağıda şekil 3.11'de gösterilen "caprazdogrulama_skor_hesapla_yazdir_10(classifier, X_train, y_train)" fonksiyonu çağırılmaktadır.

```
def caprazdogrulama_skor_hesapla_yazdir_10(classifier, X_train, y_train):
    print("\n***Çapraz Doğrulama Doğruluk Oranı Yazdırılıyor.***\n")
    ilk = sure_olcmeye_basla()
    accuracies = cross_val_score (estimator=classifier, X = X_train,
                                  y = y_train,cv=10)
    sure = calisma_suresi_getir(ilk)
    print("Doğruluk: {:.2f} %".format(accuracies.mean()*100))
    print("Standart Sapma: {:.2f} %".format(accuracies.std()*100))
    print("Süre: " + sure)
    print("\n***Çapraz Doğrulama Doğruluk Oranı Yazdırıldı.***\n")
```

Şekil 3.11 Çapraz Doğrulama Uygulanması İlişkin Kod Parçacığı.

Yukarıda yer alan kod parçacığından da görüleceği üzere "caprazdogrulama_skor_hesapla_yazdir_10(classifier, X_train, y_train)" fonksiyonu 10 kat çapraz doğrulama işlemini gerçekleştirmekte ve doğruluk oranını standart sapması ile birlikte ekrana yazdırmaktadır. Dolayısıyla, tek bir örnek için elde edilecek olası iyi bir sonucun şans eseri olup olmadığının doğrulaması yapılabilmekte ve algoritmaların karşılaştırması daha etkin bir şekilde yerine getirilebilmektedir.

Bu kapsamda, algoritmaların eğitilmesi ve test edilmesine ilişkin genel yapı yukarıda açıklanmış olup, aynı yapı tüm algoritmalar için tasarlanmıştır. Bahse konu yapıda, veri setlerinin yüklenmesi ve ön işlemlerin tamamlanması, boyut azaltma uygulanıp uygulanmaması, çapraz doğrulama yapılıp yapılmaması, tahminde bulunulması ve performans metriklerinin hesaplanması tüm algoritmalar için genelleştirilmiş olup farklılık algoritmalara ilişkin nesnelerin oluşturulmasından kaynaklanmaktadır. Bu doğrultuda, 13 değişkenli Temel Veri Seti için k-en yakın komşu algoritmasının test edilmesi amacıyla yazılmış olan ve yukarıda sürecin açıklanmasında örnek olarak ele alınan "modeli_calistir_sonuclari_getir _KNN_13Deg" fonksiyonunun ile bahsedilen diğer fonksiyonların benzerleri tüm algoritmalar ve veri setleri (Zaman Değişkenli Veri Seti, Belirlenen Veri Seti ve Nihai Veri Seti) için yazılmıştır. Sonrasında ise bahse konu fonksiyonlar aşağıda şekil 3.12'de gösterildiği şekilde çağırılmış ve sonuçları gözlemlenmiştir.



Şekil 3.12 Test Fonksiyonlarının Çağrılmasına İlişkin Kod Parçacığı.

3.5.2. Algoritmalara İlişkin Nesnelerin Oluşturulması

Yukarıda açıklanan yapıda farklılığın modellerde kullanılan algoritmaların oluşturulması sürecinde olduğu belirtilmiş olup, fonksiyonların çalıştırılması sonucunda elde edilecek bulguların değerlendirmesine geçilmeden önce yukarıda bahsedilen "algoritma_hazirla_getir(algoritmaAdi,X_train, y_train)" fonksiyonunda her bir algoritmaya ilişkin nesnelerin oluşturulmasına yönelik bilgilere aşağıda verilmektedir.

Naive Bayes sınıflandırıcısına ilişkin nesne "*sklearn.naive_bayes*" kütüphanesinin "*GaussianNB*" sınıfı kullanılarak oluşturulmuş olup herhangi bir parametre kullanılmamıştır.

Lojistik regresyon algoritmasına ilişkin sınıflandırıcı nesnesi "*sklearn.linear_model*" kütüphanesinin "*LogisticRegression*" sınıfı kullanılarak oluşturulmuş olup, kodun her çalıştığında verilerin aynı şekilde karıştırılmasını sağlayan "*random_state*" parametresi ve aşırı uyum probleminin önüne geçilmesi amacıyla uygulanacak cezayı ifade etmekte olan "C" parametresi kullanılmıştır.

K-en yakın komşu algoritmasına ilişkin sınıflandırıcı nesnesi "*sklearn.neighbors*" kütüphanesinin "*KNeighborsClassifier*" sınıfı kullanılarak oluşturulmuştur. Söz konusu sınıflandırıcı nesnesi oluşturulurken kullanılacak mesafe metriğinin gösteren "*metric*" parametresi "*minkowski*" ve bu metriğinin kuvvet değerini gösteren "p" parametresi 2 olarak kullanılmıştır. Böylece, "*metric*" ve "p" parametreleri için bu değerler seçilerek "*oklid*" uzaklığı kullanılmıştır. Kullanılacak komşu sayısını gösteren "*n_neighbors*" parametresi ise ilk aşamada varsayılan değer olan 5 olarak seçilmiş olup, hiper parametrelerin ayarlanması aşamasında bu parametre için farklı değerler test edilecektir.

Karar ağacı algoritmasına ilişkin sınıflandırıcı nesnesi "*sklearn.tree*" kütüphanesinin "*DecisionTreeClassifier*" sınıfı kullanılarak oluşturulmuş olup, parametre olarak "*criterion*" ve "*random_state*" kullanılmıştır. "*criterion*" parametresi bir bölünmenin kalitesini ölçmek için kullanılacak olan fonksiyonu belirlemek için kullanılmaktadır.

Rastgele orman algoritmasına ilişkin sınıflandırıcı nesnesi "*sklearn.ensemble*" kütüphanesinin "*RandomForestClassifier*" sınıfı kullanılarak oluşturulmuş olup, parametre olarak "*n_estimators*", "*criterion*" ve "*random_state*" kullanılmıştır. "criterion" ve "random_state" parametrelerine ilişkin bilgiler yukarıdaki algoritmalarda açıklanmıştır. "*n_estimators*" parametresi ise ormandaki ağaç sayısını göstermekte olup ilk aşamada varsayılan değer olan 100 olarak seçilmiş olup, hiper parametrelerin ayarlanması aşamasında bu parametre için farklı değerler test edilecektir.

Destek vektör makinelerine ilişkin sınıflandırıcı nesnesi "*sklearn.svm*" kütüphanesinin "SVC" sınıfı kullanılarak oluşturulmuş olup, parametre olarak "*kernel*", "*probability*", "*gamma*" ve "random_state" kullanılmıştır. "*kernel*" parametresi kullanılacak çekirdek fonksiyonunu göstermekte olup doğrusal olan ve olmayan destek vektör makinelerinin oluşturulabilmesi amacıyla kullanılmıştır. Bu amaç doğrultusunda söz konusu parametrenin "*linear*" ve "rbf" olarak seçildiği iki

farklı model oluşturulmuştur. Bu her iki modelde de "*probability*" parametresi "*true*" olarak seçilerek olasılık tahminleri yapılması etkinleştirilmiştir. Çekirdek fonksiyonunun katsayısını ise "*gamma*" parametresi göstermektedir.

3.6. Sonuçların Artırılmasına Yönelik Yöntemlerin Uygulanması

Bu bölümde, modellerin çalışma süresi ve performanslarının iyileştirilebilmesi amacıyla uygulanılan yöntemlerden bahsedilmektedir.

3.6.1. Boyut Azaltma Yönteminin Uygulanması

Öznitelik belirleme işlemi ile gereksiz öznitelikler elimine edilerek veri setinin boyutu 3 değişkene kadar düşürülmüştür. Ayrıca, oluşturulan modellerin ve izlenen yöntemlerin etkin bir şekilde değerlendirilebilmesi amacıyla alternatif olarak farklı sayıda öznitelik içeren veri setleri de korunmuştur. Bununla birlikte minimum özniteliğe sahip olan veri setinin dahi içerdiği kategorik özniteliklerin kategorilerinin fazla olması nedeniyle modellerin oluşturulması aşamasında kategorik değişkenlerin dönüşüm işlemi yapıldığında veri setinin boyutunun oldukça artmakta olduğu görülmüştür. Bu doğrultuda, özniteliklerin belirlenmesine ek olarak yapay sinir ağı ve diğer makine öğrenmesi algoritmaları ile oluşturulan modellerde PCA yöntemi uygulanarak boyut azaltma işlemi gerçekleştirilmiştir.

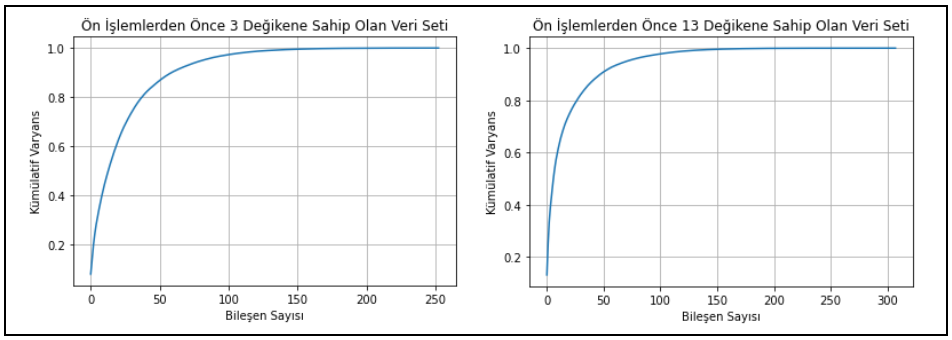
Bölüm 2.7'de detaylı olarak bahsedildiği üzere PCA yönteminde orijinal veri setindeki bilgilerin mümkün olan en yüksek oranda korunarak veri setinin daha düşük boyutlu bir temsilinin bulunması amaçlanmaktadır. Bu amaca ulaşabilmek içinse orijinal veri setinde bulunan değişkenlerin temsil ettiği varyansın, temel bileşenler adı verilen daha az sayıdaki temsili değişkenlerle toplu olarak özetlenmesi yöntemi uygulanmaktadır.

Bu kapsamda, tez çalışmasında, bahse konu yöntemin uygulanabilmesi için "boyut_azalt_PCA (X_train, X_test,component)" isimli fonksiyon yazılmış ve bu fonksiyonun, modellerin test edilmesi aşamasında çağırılmasıyla boyut azaltma işlemi gerçekleştirilmiştir. Bahse konu fonksiyona ilişkin kod parçacığı aşağıda şekil 3.13'de verilmekte olup, söz konusu şekilden de görülebileceği üzere boyut azaltma işlemi, parametre olarak verilen bileşen sayısı dikkate alınarak oluşturulan "*sklearn.decomposition*" kütüphanesinin "PCA" sınıfına ait nesnenin eğitim veri seti ile ayarlanması ve sonrasında bu nesne kullanılarak eğitim ve test veri setlerinin dönüştürülmesi suretiyle gerçekleştirilmektedir.

```
def boyut_azalt_PCA (X_train, X_test,component):
    pca = PCA(n_components = component)
    X_train = pca.fit_transform (X_train)
    X_test = pca.transform(X_test)
    return X_train, X_test
```

Şekil 3.13 Boyut Azaltma İşlemine İlişkin Kod Parçacığı.

Dolayısıyla, modeller içerisinden çağırılacak olan "boyut_azalt_PCA" fonksiyonuna parametre olarak verilecek bileşen sayısına karar verilmesi gerekmektedir. Bu bağlamda, bileşen sayısının bulunabilmesi amacıyla "PCA" sınıfından oluşturulan nesnenin "*explained_variance_ratio_"* özelliği kullanılarak bileşen sayısına göre temsil edilebilen kümülatif varyansın grafiği çizilmiştir. Bu işlem 3 değişkenli Nihai Veri Seti ve 13 değişkenli Temel Veri Seti için "bilesen_sayisi_bul_PCA_3Deg()" ve "bilesen_sayisi_bul_PCA_3Deg()" fonksiyonları aracılığı ile yapılmıştır. Bu fonksiyonların çalıştırılması sonucunda elde edilen grafiklere aşağıda şekil 3.14'de yer verilmektedir.



Şekil 3.14 Bileşen Sayılarına İlişkin Kümülatif Varyans Grafiği.

Yukarıda yer alan şekil 3.14'den de anlaşılabileceği üzere 100 temel bileşen seçilmesi durumunda her iki veri seti içinde toplan varyansın yaklaşık %95'inin korunabileceği sonucuna ulaşılmış ve 13 değişkenli Temel Veri Seti ile veri ön işleme sonucunda karar verilen 3 değişkenli Nihai Veri Seti için boyut azaltma işlemlerinde 100 temel bileşen kullanılmasına karar verilmiştir.

3.6.2. Hiper Parametrelerin Ayarlanması

Modellerin geliştirilmesinde kullanılan algoritmalar tarafından doğrudan öğrenilmeyen parametreler olan hiper parametrelerin ayarlanması modellerle elde edilecek sonuçların arttırılması için önerilmektedir. Bu çalışmada da, sonuçların artırılabilmesi amacıyla bölüm 2.8.'de detayları açıklanmış olan ızgara araması yöntemi uygulanmış olup, söz konusu yöntem önceden tanımlanmış bir dizi parametreden en iyi performansı gösterenlerin belirlenmesi esasına dayanan bir kaba kuvvet uygulamasıdır. Bu doğrultuda, ızgara araması yönteminin uygulanabilmesi için "*sklearn.model_selection*" kütüphanesinin "*GridSearchCV*" sınıfı kullanılmıştır. Aşağıda, şekil 3.15'de bahse konu yöntemin uygulanmasına ilişkin örnek bir kod parçacığı gösterilmektedir.

```
def izgara_aramasi_yap_RF_3Deg():

    dataset, X_train, X_test, y_train, y_test =
    islemleri_tamamla_verisetlerini_getir_parametre_ayarlama_3Deg(isANN=0)
    classifier = RandomForestClassifier (n_estimators =100,
                                         criterion='entropy', random_state=999)
    classifier.fit(X_train,y_train)


    print("\n***RASTGELE ORMAN ALGORİTMASI İÇİN IZGARA ARAMASI BAŞLADI***\n")
    parametreler = [{'n_estimators':[50,75,100,125,150,175,200],
                     'criterion':['entropy', 'gini']}]
    izgara_arama= GridSearchCV (estimator=classifier,
                                param_grid= parametreler,
                                scoring='accuracy',
                                cv=10,
                                n_jobs=-1)
    izgara_arama.fit(X_train,y_train)
    en_iyi_dogruluk=izgara_arama.best_score_
    en_iyi_parametreler = izgara_arama.best_params_


    print("En iyi doğruluk: {:.2f} %".format(en_iyi_dogruluk*100))
    print("En iyi parametreler:", en_iyi_parametreler)
    print("\n***RASTGELE ORMAN ALGORİTMASI İÇİN IZGARA ARAMASI TAMAMLANDI***\n")
```

Şekil 3.15 Izgara Araması Yöntemine İlişkin Kod Parçacığı.

Yukarıda yer alan kod parçacığından da görülebileceği üzere ızgara araması yönteminin uygulanabilmesi için öncelikle bazı hiper parametrelerin sonsuz değer alabileceği dikkate alınarak sahip olunan ön bilgilerin kullanılması suretiyle parametreler için aralıklar belirlenmiştir. Daha sonra ise bu parametreler ve parametre ayarlaması yapılacak algoritmaya ait nesne kullanılarak "*GridSearchCV*" sınıfının bir nesnesi oluşturulmuştur. Yeni oluşturulan bu nesne ile eğitim veri seti üzerinde 10-kat çapraz doğrulama yöntemi kullanılarak doğruluk oranlarının karşılaştırılması gerçekleştirilmiştir. Son olarak ise ızgara aramasının gerçekleştirildiği nesnenin "*best_params_*" özelliği kullanılarak en iyi doğruluğun elde edilmesini sağlayan parametrelere ulaşılmış ve ızgara araması yöntemine ilişkin süreç tamamlanmıştır. Ayrıca, parametre kombinasyonunun ve parametrelerin alabileceği değerlerin çok fazla olduğu durumlarda hesaplama maliyetinin fazla olacağı dikkate alınarak parametre kombinasyonlarının ve parametrelerin alabileceği değerlerin grup grup ele alınması yöntemi uygulanmıştır. Örneğin, yapay sinir ağlarında karar verilmesi ihtiyacı olan hiper parametrelerin sayısı göz önüne alınarak öncelikle kullanılacak gizli katman sayısının bulunmasına yönelik işlemler, daha sonra ise "*epoch*" ve "*batch_size*" parametleri ayarlanmasına yönelik işlemler ve son olarak da "*units*" ve "*dropout_rate*" parametrelerinin ayarlanmasına yönelik işlemler gerçekleştirilmiştir. Ayrıca, yapay sinir ağlarında parametrelerin alacağı değerler açısından da benzer bir

yaklaşımla hareket edilmiştir. Bu yaklaşıma örnek vermek gerekirse yapay sinir ağlarında "*units*" parametresi için öncelikle [20, 25, 30] örneklemleri test edilmiş ve bu örneklemden elde edilen sonucun 30 olması ile bir sonraki işlemde [25,30,35,40,45] değerleri kullanılarak sonucun değişip değişmediği gözlemlenmiştir. Böylece, ilk aşamada elde edilen sonucun aşağısı ve yukarısı değerler kullanılarak en iyi parametrenin yönünün değişip değişmediğine göre karar verilerek hareket edilmiştir. Dolayısıyla, hesaplama maliyetinin söz konusu olabileceği durumlarda en iyi parametre kombinasyonuna ulaşabilmek için ana noktalar seçilerek hareket edilmesi yöntemi benimsenmiştir ki burada veri setinden küçük bir örneklemin kullanılması veya çapraz doğrulama sayısının küçük tutulması gibi farklı yaklaşımlar uygulanabilmesi de mümkündür. Diğer taraftan, parametre kombinasyonunun ve parametrelerin alabileceği durumların çok fazla olmaması durumunda hesaplama maliyetinin az olacağı dikkate alınarak doğrudan ızgara araması yönteminin uygulanması tercih edilmiştir. Bahsedilen yaklaşımlarla parametrelerin ayarlanması işlemine 10-kat çapraz doğrulama yöntemi uygulanarak en iyi parametrelere ulaşılana kadar devam edilmiş olup, ayarlanan parametrelere ilişkin bilgiler aşağıda verilmektedir. Parametre ayarlamasının etkisi ise bulgular bölümünde değerlendirilmektedir.

Lojistik regresyon algoritması için aşırı uyum probleminin önüne geçilmesi amacıyla uygulanacak cezayı ifade etmekte olan "C" parametresine ilişkin olarak "0,25; 0,50; 0,75; 1" değerleri denenmiş ve bu parametrenin değeri 1 olarak belirlenmiştir.

K-En yakın komşu algoritması için kullanılacak komşu sayısını göstermekte olan "*n_neighbors*" parametresine ilişkin olarak "5,6,7,8,9,10" değerleri denenmiş ve bu parametrenin değeri 10 olarak belirlenmiştir.

Karar ağacı algoritması için bölünmenin etkinliğini ölçme fonksiyonunu ifade etmekte olan "*criterion*" parametresine ilişkin olarak "*entropy, gini*" değerleri denenmiş ve bu parametrenin değeri "*gini*" olarak belirlenmiştir.

Rastgele orman algoritması için kullanılacak ağaç sayısını göstermekte olan '*n_estimators*' parametresine ilişkin olarak "50,75,100,125,150,175,200" değerleri ve "*criterion'* parametresine ilişkin olaraksa karar ağacı algoritmasındaki benzer şekilde "*entropy, gini*" değerleri denenmiş ve '*n_estimators*" parametresinin değeri "200" , "*criterion*" parametresinin değeri "*entropy*" olarak belirlenmiştir.

Doğrusal destek vektör makineleri için lojistik regresyon algoritmasına benzer şekilde 'C' parametresine ilişkin olarak "0,25; 0,50; 0,75; 1" değerleri denenmiş ve bu parametrenin değeri 1 olarak belirlenmiştir.

Doğrusal olmayan destek vektör makineleri için çekirdek fonksiyonunun katsayısını göstermekte olan "*gamma*" parametresine ilişkin olarak "0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0.9" değerleri denenmiş ve bu parametrenin değeri 0,5 olarak belirlenmiştir. Ayrıca, doğrusal destek vektör makinelerine benzer şekilde "C" parametresine ilişkin olarak "0,25; 0,50; 0,75; 1" değerleri denenmiş ve bu parametrenin değeri olarak da 1 belirlenmiştir.

Yapay sinir ağları için öncelikle "1,2,3,4,5,6,7,8,16" değerleri denenerek gizli katman sayısının kaç olması gerektiği incelenmiş gizli katman sayısı 1 olarak belirlenmiştir. Daha sonra ise yukarıda bahsedilen yaklaşımla; eğitim veri seti üzerinden kaç defa geçileceğini belirtmekte olan "*epoch*" parametresi 50, her bir geçişte ağ üzerinde yayılacak örneklerin sayısını göstermekte olan "*batch_size*" parametresi 128, gizli katmanları boyutunu göstermekte olan "*units*" parametresi 30 ve aşırı öğrenme probleminin önüne geçilmesi için hangi oranda devre dışı bırakma yapılacağını göstermekte olan "*dropout_rate*" parametresi 0,3 olarak belirlenmiştir.

4. TARTIŞMA

Tez çalışmasının bu bölümünde makine öğrenmesi algoritmaları kullanılarak geliştirilen modellerin veri setlerine uygulanması sonrasında elde edilen performans bulgularına yönelik tartışmalara yer verilmektedir. Bu doğrultuda, bulgular üzerinden analiz işlemleri gerçekleştirilmesi için kullanılan veri setlerine ilişkin detaylı bilgiler bölüm 3.3.'de yer almakta olup, söz konusu veri setlerinin kullanım amaçları ve bulgular üzerinde değerlendirme yapılırken dikkat edilmesi gerekli hususlar kısaca şu şekilde özetlenebilecektir:

Temel Veri Seti: Ham veri seti üzerinde veri temizleme ve veri sızıntısı olan özniteliklerin kaldırılması işlemi sonrasında elde edilmiş olan 13 adet özniteliğe sahip bu veri seti; öznitelik belirleme, algoritmaların test edilmesi ve uygulanan diğer işlemlerin etkinliğinin ölçülmesi bakımından başlangıç noktası olarak dikkate alınabilecek veri setidir.

Zaman Değişkenli Veri Seti: Temel Veri Seti'nden farklı olarak veri temizleme aşamasında zaman değişkenleri kaldırılmamış veri seti üzerinden öznitelik belirleme işlemi yapılması sonucunda elde edilmiş olan 3 adedi zaman değişkeni olmak üzere toplam 7 adet özniteliğe sahip veri setidir. Bu veri seti, veri sızıntısı bulunan ve gelecek olaylarda bir anlam ifade etmeyecek olması sebebiyle modellerin genelleştirilmesinde engel olan zaman değişkenlerinin etkisinin gösterilmesi amacıyla kullanılmıştır.

Belirlenmiş Veri Seti: Temel Veri Seti'nden aralarında korelasyon bulunan özniteliklerin elenmesiyle belirlenen 4 adet özniteliğe sahip veri setidir.

Nihai Veri Seti: Belirlenmiş Veri Seti'nde bulunan özniteliklerinin önem derecelerinin incelenmesi sonrasında diğerlerine kıyasla çok düşük önem derecesine sahip "*nkill*" özniteliğinin elenmesiyle elde edilmiş ve 3 adet özniteliğe sahip veri

setidir. Esas itibariyle bu veri setinin modellerin değerlendirilmesi ve uygulanan yöntemlerin etkinliği açısından Temel Veri Seti ile karşılaştırılması uygun olacaktır.

Diğer taraftan, bulgular üzerinden değerlendirme yapılırken kullanılacak performans ölçüm kriterleri hakkındaki detaylı bilgilere ve hesaplanma yöntemlerine bölüm 2.6.'da yer verilmiş olup, söz konusu performans ölçüm kriterlerinin ne ifade ettiği şu şekilde özetlenebilecektir:

Doğruluk: Doğru şekilde sınıflandırılmış örneklerin toplam örnek sayısına bölünmesiyle elde edilmektedir.

Hassasiyet: Olayı gerçekleştirdiği tahmin edilen terör örgütlerinden kaç adedinin doğru tahmin edilmiş olduğunu göstermektedir. Dolayısıyla, bu ölçütün yüksek olması algoritmaların başarısı açısından olumlu bir göstergedir. Diğer bir deyişle, terör olayını gerçekleştirdi olarak tahmin edilen bir örgütün aslında olayı gerçekleştirmemiş olmasının maliyetini temsil etmektedir.

Duyarlılık: Modellerin doğru değerleri tahmin etmedeki başarısını göstermektedir. Yani olayı gerçekleştiren terör örgütlerinin tahmin edilmesindeki etkinliği ifade etmektedir. Diğer bir değişle olayı gerçekleştiren bir terör örgütünün olayı gerçekleştirmemiş olarak tahmin edilmesinin maliyetini belirtmektedir.

F1 Skoru: Hassasiyet oranının artması için duyarlılık oranından feragat edilmesi söz konusu olabildiğinden f1 skoru hassasiyet ve duyarlılık metriklerinin harmonik ortalaması olarak hesaplanmaktadır. Dolayısıyla, f1 skoru bahse iki oran arasında dengenin bulunması amacıyla hassasiyet oranı ve duyarlılık oranının kapsamlı bir şekilde ele alındığı ölçüttür.

AUC: YP oranına karşılık DP oranı için çizilen ROC grafiği altında kalan alanı göstermekte olup, sınıflar arasında ne kadar iyi ayrım yapabildiği hakkında bilgi

sağlamaktadır. Dolayısıyla, değer ne kadar büyükse algoritmanın sınıfları ayırma konusunda ve genelleme performansında o kadar başarılı olduğu söylenebilecektir.

Son olarak, parametre seçimi hakkında değerlendirme yapılabilmesi amacıyla parametre ayarlaması yapılmamış modellerin 3 değişkenli Nihai Veri Seti'ne uygulanması ile elde edilen bulgulara da bu bölümde yer verilmektedir.

4.1. Naive Bayes Sınıflandırıcısı İçin Bulguların Değerlendirilmesi

Naive Bayes sınıflandırıcısı kullanılarak oluşturulan modellerin test edilmesi için uygulamanın çalıştırılması sonucunda elde edilen ve karmaşıklık matrisi ile bu karmaşıklık matrisi kullanılarak hesaplanan performans ölçüm kriterlerini de göstermekte olan uygulama çıktısında bulunan bilgiler aşağıdaki çizelgelerde özetlenmektedir.

Çizelge 4.1'de *Naive Bayes* sınıflandırıcısı kullanılarak oluşturulan modelin veri setlerine ve bu veri setlerine boyut azaltma işlemi uygulanmasıyla elde edilen veri setlerine uygulanması sonucunda elde edilen performans kriterleri gösterilmektedir. Söz konusu çizelgeden de görülebileceği üzere boyut azaltama işlemi uygulanmadan önce en yüksek doğruluk oranına 13 değişkenli veri seti ile ulaşılmış olup, bu veri setini sırasıyla 7,6,4 ve 3 değişkenli veri setleri izlemiştir. Keza diğer performans metriklerinde de aynı sıralama elde edilmiş ve doğruluk oranı ile ilgili bir tutarsızlık gözlemlenmemiştir. Diğer taraftan, 3 değişkenli veri setine boyut azaltma işleminin uygulanması sonucunda doğruluk oranının yaklaşık %65 seviyesinden %80 seviyesine çıktığı ve hassasiyet, duyarlılık ve f1 skoru ölçütlerinde de benzer artışın olduğu görülmektedir. Bununla birlikte, boyut azaltma işlemi sonucunda 13 değişkenli veri setine ilişkin doğruluk oranı %73 seviyesinden %81,5 seviyesine yükselmiştir. Dolayısıyla, 3 değişkenli veri seti ile 13 değişkenli veri seti arasında doğruluk oranı ve diğer ölçütler bakımından boyut azaltma işlemi uygulanması ile iyileştirilmiş olarak elde edilen sonuçlar arasında kayda değer bir farklılık bulunmamakta olup, boyut azaltma işlemi sonrasında 3 değişkenli veri seti için AUC değeri 0,957621 olarak elde

edilmiştir. Ayrıca, hemen hemen tüm veri setleri için modellerin çalışma sürelerinin 10 saniyenin altında olacak şekilde oldukça hızlı olduğu görülmüştür. Dolayısıyla, öznitelik seçimi ve boyut azaltma işleminin etkin bir sonuç verdiği değerlendirmesine ulaşılmıştır.

Çizelge 4.1 NB-Çapraz Doğrulama Uygulanmadan Elde Edilen Sonuçlar.

Boyut Azaltma	Değişken Sayısı	Doğruluk	Hassasiyet	Duyarlılık	F1 Skoru	Süre (s:d:s)
Hayır	13	0,727617	0,776669	0,797815	0,732481	0:00:07.131
Hayır	7	0,710893	0,758889	0,786801	0,716403	0:00:04.770
Hayır	4	0,705719	0,753498	0,782496	0,711517	0:00:03.788
Hayır	3	0,648711	0,718323	0,753616	0,665326	0:00:03.337
Evet	13	0,815486	0,790003	0,805238	0,793612	0:00:09.709
Evet	3	0,806154	0,764593	0,780758	0,767612	0:00:05.847

Yukarıda yer alan sonuçlar dikkate alındığında ulaşılan durumun tesadüf olup olmadığı hakkında değerlendirme yapılabilmesi amacıyla aynı veri setlerine aynı model 10-kat çapraz doğrulama yöntemi ile uygulanmış olup, elde edilen doğruluk oranlarına ve bu doğruluk oranlarına ilişkin standart sapma değerlerine aşağıdaki çizelge 4.2'de yer verilmektedir. Aşağıdaki çizelgeden de görülebileceği üzere beklendiği şekilde tüm veri setlerine ilişkin doğruluk oranında küçük bir azalma meydana gelmekte ancak bu durum modelin genelleştirilebilir olduğunu ifade etmektedir. Sonuç olarak 13 değişkenli Temel Veri Seti ve 3 değişkenli Nihai Veri Seti arasında boyut azaltma işlemi sonucunda elde edilen doğruluk oranında önemli bir farklılık bulunmadığı ve her iki veri setinde de boyut azaltma işlemi uygulanması durumunda sonuçlarda iyileşme meydana geldiği anlaşılmıştır. Çapraz doğrulama uygulanması durumunda da modellerin çalışma süresinin verimli olduğu görülmektedir.

Çizelge 4.2 NB-Çapraz Doğrulama Uygulanarak Elde Edilen Sonuçlar.

Boyut Azaltma	Değişken Sayısı	Doğruluk Oranı (%)	Standart Sapma (%)	Süre (s:d:s)
Hayır	13	71,48	0,96	0:00:07.131
Hayır	7	68,86	2,70	0:00:02.949
Hayır	4	67,57	3,07	0:00:02.906
Hayır	3	64,00	3,11	0:00:02.930
Evet	13	80,81	0,74	0:00:01.170
Evet	3	79,99	0,59	0:00:01.168

Naive Bayes sınıflandırıcı ile oluşturulan modele ilişkin herhangi bir hiper parametre ayarlaması yapılmasına ihtiyaç duyulmamış olduğundan parametre ayarlamasının etkisinin değerlendirilmesi bu model için mümkün olmamıştır.

4.2. Lojistik Regresyon Algoritması İçin Bulguların Değerlendirilmesi

Lojistik regresyon algoritması kullanılarak oluşturulan modellerin test edilmesi için uygulamanın çalıştırılması sonucunda elde edilen ve karmaşıklık matrisi ile bu karmaşıklık matrisi kullanılarak hesaplanan performans ölçüm kriterlerini de göstermekte olan uygulama çıktısında bulunan bilgiler aşağıdaki çizelgelerde özetlenmektedir.

Çizelge 4.3'te Lojistik regresyon algoritması kullanılarak oluşturulan modelin veri setlerine ve bu veri setlerine boyut azaltma işlemi uygulanmasıyla elde edilen veri setlerine uygulanması sonucunda elde edilen performans kriterleri gösterilmektedir. Söz konusu çizelgeden de görülebileceği üzere en yüksek doğruluk oranına 7 adet özniteliğe sahip Zaman Değişkenli Veri Seti ile ulaşılmıştır. Ancak hemen belirtmek gerekir ki bu veri seti gerçekleşen yeni bir olayın tahmin edilmesi açısından gerçek dünya uygulamalarında kullanılması konusunda verimli bir veri seti değildir. Bununla birlikte boyut azaltma işlemi uygulanmadan önce Temel Veri Seti'nde %93,5 civarında, nihai veri setinde ise %92,8 civarında doğruluk oranına ulaşılmış olup diğer

performans ölçüm kriterleri bakımından da doğruluk oranı ile uyumlu sonuçlar elde edilmiştir. Boyut azaltma işleminden sonra ise bu iki veri setinden Temel Veri Seti'nin doğruluk oranı %92,9 civarına, Nihai Veri Seti'nin doğruluk oranı ise %92,3 civarına gerilemiş, performans kriterlerinde kayda değer bir azalma olmadığı görülmüştür. Boyut azaltma işlemi sonrasında, çalışma süreleri ise Temel Veri Seti için 3 saniyelik bir iyileşme göstererek 20 saniye civarına, Nihai Veri Seti içinse 2 saniyelik bir iyileşme göstererek 16 saniye civarına gelmiştir. Ayrıca, boyut azaltma işlemi sonrasında Nihai Veri Seti için AUC değeri 0,996443 olarak elde edilmiştir. Dolayısıyla, Temel Veri Seti ve Nihai Veri Seti üzerinde çalışan modelde performans ölçütlerinin birbirinden çok farklı olmadığı da dikkate alınarak öznitelik seçimi ve boyut azaltma işlemlerinin etkin bir sonuç verdiği değerlendirmesine ulaşılmıştır.

Çizelge 4.3 Çapraz Doğrulama Uygulanmadan Elde Edilen Sonuçlar.

Boyut Azaltma	Değişken Sayısı	Doğruluk	Hassasiyet	Duyarlılık	F1 Skoru	Süre (s:d:s)
Hayır	13	0.935877	0.908137	0.903060	0.904168	0:00:23.055
Hayır	7	0.939111	0.914606	0.908460	0.908697	0:00:19.621
Hayır	4	0.929872	0.900468	0.892294	0.893319	0:00:17.851
Hayır	3	0.928393	0.897408	0.889993	0.890478	0:00:18.858
Evet	13	0.929687	0.900819	0.895179	0.896240	0:00:20.836
Evet	3	0.922480	0.890503	0.883199	0.883430	0:00:16.649

Yukarıda yer alan sonuçlar dikkate alındığında ulaşılan durumun tesadüf olup olmadığı hakkında değerlendirme yapılabilmesi amacıyla aynı veri setlerine aynı model 10-kat çapraz doğrulama yöntemi ile uygulanmış olup, elde edilen doğruluk oranlarına ve bu doğruluk oranlarına ilişkin standart sapma değerlerine aşağıdaki çizelge 4.4'de yer verilmektedir. Aşağıdaki çizelgeden de görülebileceği üzere beklendiği şekilde tüm veri setlerine ilişkin doğruluk oranında küçük bir azalma meydana gelmekte ancak bu durum modelin genelleştirilebilir olduğunu ifade etmektedir. Sonuç olarak 13 değişkenli Temel Veri Seti ve 3 değişkenli Nihai Veri Seti arasında boyut azaltma işlemi sonucunda elde edilen doğruluk oranında önemli bir farklılık bulunmadığı ve her iki veri setinde de boyut azaltma işlemi uygulanması

durumunda sonuçlarda önemli ölçüde kötüleşme meydana gelmediği anlaşılmıştır. Çapraz doğrulama uygulanması durumunda da modellerin çalışma süresinin verimli olduğu görülmektedir.

Çizelge 4.4 LR-Çapraz Doğrulama Uygulanarak Elde Edilen Sonuçlar.

Boyut Azaltma	Değişken Sayısı	Doğruluk Oranı (%)	Standart Sapma (%)	Süre (s:d:s)
Hayır	13	93.07	0.42	0:02:36.391
Hayır	7	93.44	0.40	0:02:20.784
Hayır	4	92.41	0.40	0:02:18.823
Hayır	3	92.33	0.40	0:02:27.465
Evet	13	92.48	0.41	0:01:40.719
Evet	3	91.87	0.39	0:01:45.242

Son olarak, parametre ayarlaması yapılmamış algoritma ile oluşturulan modele ait doğruluk oranlarına ve ilgili standart sapmalara aşağıda yer verilmekte olup doğruluk oranında herhangi bir kötüleşme meydana gelmemiş ve çok küçük bir artış gözlemlenmiştir.

Çizelge 4.5 LR-Nihai Veri Seti Parametre Ayarlamasız Elde Edilen Sonuçlar.

Boyut Azaltma	Çapraz Doğrulama	Doğruluk Oranı (%)	Standart Sapma (%)	Süre (s:d:s)
Hayır	Hayır	92,84	-	0:00:18.836
Hayır	Evet	92,33	0,40	0:02:27.272
Evet	Hayır	92,25	-	0:00:16.856
Evet	Evet	91,88	0,38	0:01:44.860

4.3. K-en Yakın Komşu Algoritması İçin Bulguların Değerlendirilmesi

K-en yakın komşu algoritması kullanılarak oluşturulan modellerin test edilmesi için uygulamanın çalıştırılması sonucunda elde edilen ve karmaşıklık matrisi ile bu karmaşıklık matrisi kullanılarak hesaplanan performans ölçüm kriterlerini de

göstermekte olan uygulama çıktısında bulunan bilgiler aşağıdaki çizelgelerde özetlenmektedir.

Çizelge 4.6'da k-en yakın komşu algoritması kullanılarak oluşturulan modelin veri setlerine ve bu veri setlerine boyut azaltma işlemi uygulanmasıyla elde edilen veri setlerine uygulanması sonucunda elde edilen performans kriterleri gösterilmektedir. Söz konusu çizelgeden de görülebileceği üzere doğruluk oranı ve diğer performans ölçüm metrikleri birbiri ile uyumlu ve tutarlı sonuçlar vermişlerdir. Bu doğrultuda, doğruluk oranları incelendiğinde özellikle Temel Veri Seti'nde %85 civarında doğruluk oranı elde edilirken Nihai Veri Seti'nde %91 civarında doğruluk oranı elde edilmiştir ki bu da öznitelik seçim işleminin uygun olduğunu göstermektedir. Ek olarak Nihai Veri Seti 4 değişkenli Belirlenmiş Veri Seti'nden de daha iyi performans sonuçlarına ulaşmıştır. Ayrıca, gerek Temel Veri Seti gerekse de Nihai Veri Seti üzerinde boyut azaltma işlemi uygulandığında 7 dakika bandında olan çalışma sürelerinde yaklaşık 6 dakika civarında bir iyileşme gözlemlenmiş ve doğruluk oranında önemli bir azalma olmadığı görülmüştür. Böylece boyut azaltma işleminin yerinde olduğu anlaşılmıştır. Ayrıca, boyut azaltma işlemi sonrasında Nihai Veri Seti için AUC değeri 0,986580 olarak elde edilmiştir.

Çizelge 4.6 KNN-Çapraz Doğrulama Uygulanmadan Elde Edilen Sonuçlar.

Boyut Azaltma	Değişken Sayısı	Doğruluk	Hassasiyet	Duyarlılık	F1 Skoru	Süre (s:d:s)
Hayır	13	0,849764	0,829498	0,796849	0,809578	0:06:32.773
Hayır	7	0,914719	0,891489	0,877973	0,883657	0:02:24.314
Hayır	4	0,911670	0,881179	0,873058	0,876487	0:05:22.213
Hayır	3	0,910006	0,878356	0,868166	0,871974	0:07:30.350
Evet	13	0,836182	0,809175	0,784147	0,793783	0:01:20.783
Evet	3	0,904093	0,863428	0,861218	0,861733	0:00:45.007

Yukarıda yer alan sonuçlar dikkate alındığında ulaşılan durumun tesadüf olup olmadığı hakkında değerlendirme yapılabilmesi amacıyla aynı veri setlerine aynı model 10-kat çapraz doğrulama yöntemi ile uygulanmış olup, elde edilen doğruluk oranlarına ve bu doğruluk oranlarına ilişkin standart sapma değerlerine aşağıdaki

çizelge 4.7'de yer verilmektedir. Aşağıdaki çizelgeden de görülebileceği üzere beklendiği şekilde tüm veri setlerine ilişkin doğruluk oranında küçük bir azalma meydana gelmekle birlikte dikkat çekici bir azalma gözlemlenmemiştir. Nihai Veri Seti ile daha yüksek doğruluk oranı ve daha iyi çalışma süresi elde edilmiştir.

Çizelge 4.7 KNN-Çapraz Doğrulama Uygulanarak Elde Edilen Sonuçlar.

Boyut Azaltma	Değişken Sayısı	Doğruluk Oranı (%)	Standart Sapma (%)	Süre (s:d:s)
Hayır	13	84,14	0,48	0:12:06.151
Hayır	7	91,09	0,61	0:04:44.489
Hayır	4	90,50	0,51	0:10:04.446
Hayır	3	90,52	0,44	0:14:19.252
Evet	13	82,68	0,56	0:02:17.954
Evet	3	89,84	0,37	0:01:25.967

Son olarak, parametre ayarlaması yapılmamış algoritma ile oluşturulan modele ait doğruluk oranlarına ve ilgili standart sapmalara aşağıda yer verilmekte olup doğruluk oranında herhangi bir kötüleşme meydana gelmemiş ve küçük bir artış gözlemlenmiştir.

Çizelge 4.8 KNN-Nihai Veri Seti Parametre Ayarlamasız Elde Edilen Sonuçlar.

Boyut Azaltma	Çapraz Doğrulama	Doğruluk Oranı (%)	Standart Sapma (%)	Süre (s:d:s)
Hayır	Hayır	0,912409	-	0:06:59.684
Hayır	Evet	90,37	0,47	0:13:17.163
Evet	Hayır	0,901598	-	0:00:28.215
Evet	Evet	89,76	0,54	0:00:51.454

4.4. Karar Ağacı Algoritması İçin Bulguların Değerlendirilmesi

Karar ağacı algoritması kullanılarak oluşturulan modellerin test edilmesi için uygulamanın çalıştırılması sonucunda elde edilen ve karmaşıklık matrisi ile bu karmaşıklık matrisi kullanılarak hesaplanan performans ölçüm kriterlerini de

göstermekte olan uygulama çıktısında bulunan bilgiler aşağıdaki çizelgelerde özetlenmektedir.

Çizelge 4.9'da karar ağacı algoritması kullanılarak oluşturulan modelin veri setlerine ve bu veri setlerine boyut azaltma işlemi uygulanmasıyla elde edilen veri setlerine uygulanması sonucunda elde edilen performans kriterleri gösterilmektedir.

Söz konusu çizelgeden; Nihai Veri Seti'nin, Temel Veri Seti ve Belirlenmiş Veri Seti'ne nazaran daha iyi doğruluk oranına ulaşıldığı ve doğruluk oranı ile diğer performans kriterleri arasında bir uyumsuzluk bulunmadığı görülmektedir. Ayrıca, boyut azaltma işlemi neticesinde Temel Veri Seti'nin doğruluk oranı %92 civarından %88 civarına düşerek önemli ölçüde kötüleşirken Nihai Veri Seti'nde yalnız yaklaşık %0,08 civarında bir azalma ile %91,7 civarında bir doğruluk oranı elde edilmiştir. Bununla birlikte boyut azalma işlemi neticesinde çalışma süresinde Temel Veri Seti için 28 saniyelik, Nihai Veri Seti'nde ise 10 saniyelik bir kötüleşme görülmüş olmakla birlikte tüm çalışma sürelerinin 40 saniye gibi kısa bir süreden daha az olduğu görülmüştür. Ayrıca, boyut azaltma işlemi sonrasında Nihai Veri Seti için AUC değeri 0,972454 olarak elde edilmiştir.

Çizelge 4.9 DT-Çapraz Doğrulama Uygulanmadan Elde Edilen Sonuçlar.

Boyut Azaltma	Değişken Sayısı	Doğruluk	Hassasiyet	Duyarlılık	F1 Skoru	Süre (s:d:s)
Hayır	13	0,924697	0,892628	0,890691	0,891366	0:00:07.771
Hayır	7	0,945117	0,922309	0,921656	0,921840	0:00:05.145
Hayır	4	0,922665	0,891175	0,885896	0,888230	0:00:04.410
Hayır	3	0,925991	0,896846	0,889239	0,891537	0:00:04.056
Evet	13	0,880809	0,840541	0,835576	0,837672	0:00:35.743
Evet	3	0,917398	0,884895	0,877906	0,879837	0:00:14.218

Yukarıda yer alan sonuçlar dikkate alındığında ulaşılan durumun tesadüf olup olmadığı hakkında değerlendirme yapılabilmesi amacıyla aynı veri setlerine aynı model 10-kat çapraz doğrulama yöntemi ile uygulanmış olup, elde edilen doğruluk oranlarına ve bu doğruluk oranlarına ilişkin standart sapma değerine aşağıdaki çizelge

4.10'da yer verilmektedir. Aşağıdaki çizelgeden de görülebileceği üzere beklendiği şekilde tüm veri setlerine ilişkin doğruluk oranında küçük bir azalma meydana gelmekle birlikte dikkat çekici bir azalma gözlemlenmemiştir. Nihai Veri Seti ile daha yüksek doğruluk oranı ve daha iyi çalışma süresi elde edilmiştir. Ancak boyut azaltma işlemi sonrasında çalışma süresinde kötüleşme meydana geldiği gözlemlenmiştir.

Çizelge 4.10 DT-Çapraz Doğrulama Uygulanarak Elde Edilen Sonuçlar.

Boyut Azaltma	Değişken Sayısı	Doğruluk Oranı (%)	Standart Sapma (%)	Süre (s:d:s)
Hayır	13	92,13	0,39	0:00:12.974
Hayır	7	94,01	0,31	0:00:09.380
Hayır	4	91,81	0,38	0:00:09.058
Hayır	3	92,07	0,43	0:00:08.985
Evet	13	87,97	0,34	0:03:58.348
Evet	3	91,16	0,40	0:01:20.609

Son olarak, parametre ayarlaması yapılmamış algoritma ile oluşturulan modele ait doğruluk oranlarına ve ilgili standart sapmalara aşağıda yer verilmekte olup doğruluk oranında herhangi bir kötüleşme meydana gelmemiş ve küçük bir artış gözlemlenmiştir.

Çizelge 4.11 DT-Nihai Veri Seti Parametre Ayarlamasız Elde Edilen Sonuçlar.

Boyut Azaltma	Çapraz Doğrulama	Doğruluk Oranı (%)	Standart Sapma (%)	Süre (s:d:s)
Hayır	Hayır	92,60	-	0:00:04.021
Hayır	Evet	92,07	0,43	0:00:09.094
Evet	Hayır	91,40	-	0:00:14.243
Evet	Evet	91,13	0,45	0:01:19.897

4.5. Rastgele Orman Algoritması İçin Bulguların Değerlendirilmesi

Rastgele orman algoritması kullanılarak oluşturulan modellerin test edilmesi için uygulamanın çalıştırılması sonucunda elde edilen ve karmaşıklık matrisi ile bu karmaşıklık matrisi kullanılarak hesaplanan performans ölçüm kriterlerini de göstermekte olan uygulama çıktısında bulunan bilgiler aşağıdaki çizelgelerde özetlenmektedir.

Çizelge 4.12'de rastgele orman algoritması kullanılarak oluşturulan modelin veri setlerine ve bu veri setlerine boyut azaltma işlemi uygulanmasıyla elde edilen veri setlerine uygulanması sonucunda elde edilen performans kriterleri gösterilmektedir.

Söz konusu tablodan da görülebileceği üzere Nihai Veri Seti ve Belirlenmiş Veri Seti için hemen hemen aynı doğruluk değerleri elde edilmiş ve bu doğruluk değerleri ile Temel Veri Seti için elde edilen doğruluk değeri arasında önemli bir farklılık gözlemlenmemiştir. Ayrıca, boyut azaltma işlemi uygulanması durumunda da Nihai Veri Seti ve Temel Veri Seti için %92,55 civarında hemen hemen aynı doğruluk oranı elde edilmiştir. Bununla birlikte, karar ağacı algoritması ile geliştirilen modele benzer şekilde boyut azaltma işlemi neticesinde doğruluk oranında önemli bir azalma oluşmazken çalışma sürelerinde yaklaşık 30 saniye ile 1 dakika arasında bir kötüleşme meydana geldiği gözlemlenmiştir. Ayrıca, boyut azaltma işlemi sonrasında Nihai Veri Seti için AUC değeri 0, 994605 olarak elde edilmiş ve yukarıda değerlendirilen doğruluk oranı ile diğer performans metriklerinin uyumlu olduğu görülmüştür.

Çizelge 4.12 RF-Çapraz Doğrulama Uygulanmadan Elde Edilen Sonuçlar.

Boyut Azaltma	Değişken Sayısı	Doğruluk	Hassasiyet	Duyarlılık	F1 Skoru	Süre (s:d:s)
Hayır	13	0,932089	0,902416	0,899785	0,900835	0:00:32.585
Hayır	7	0,956759	0,941990	0,937604	0,939098	0:00:30.152
Hayır	4	0,927746	0,896378	0,891565	0,893502	0:00:24.098
Hayır	3	0,927284	0,897991	0,890436	0,892448	0:00:30.049
Evet	13	0,926638	0,897004	0,891139	0,893550	0:02:03.721
Evet	3	0,925437	0,895273	0,889246	0,890697	0:01:02.488

Yukarıda yer alan sonuçlar dikkate alındığında ulaşılan durumun tesadüf olup olmadığı hakkında değerlendirme yapılabilmesi amacıyla aynı veri setlerine aynı model 10-kat çapraz doğrulama yöntemi ile uygulanmış olup, elde edilen doğruluk oranlarına ve bu doğruluk oranlarına ilişkin standart sapma değerlerine aşağıdaki çizelge 4.13'te yer verilmektedir. Aşağıdaki çizelgeden de görülebileceği üzere beklendiği şekilde tüm veri setlerine ilişkin doğruluk oranında küçük bir azalma meydana gelmekle birlikte dikkat çekici bir azalma gözlemlenmemiştir. Nihai Veri Seti ve Temel Veri Seti için hemen hemen aynı doğruluk oranı elde edilmiştir. Ancak boyut azaltma işlemi sonrasında her iki veri seti içinde çalışma süresinde kötüleşme meydana geldiği gözlemlenmiştir

Çizelge 4.13 RF-Çapraz Doğrulama Uygulanarak Elde Edilen Sonuçlar.

Boyut Azaltma	Değişken Sayısı	Doğruluk Oranı (%)	Standart Sapma (%)	Süre (s:d:s)
Hayır	13	92,88	0,34	0:03:41.875
Hayır	7	94,98	0,36	0:02:52.023
Hayır	4	92,25	0,31	0:03:49.836
Hayır	3	92,29	0,41	0:03:52.142
Evet	13	92,29	0,33	0:16:43.841
Evet	3	92,00	0,43	0:08:29.807

Son olarak, parametre ayarlaması yapılmamış algoritma ile oluşturulan modele ait doğruluk oranlarına ve ilgili standart sapmalara aşağıda yer verilmekte olup doğruluk oranında herhangi bir kötüleşme meydana gelmemiş ve küçük bir artış gözlemlenmiştir.

Çizelge 4.14 RF-Nihai Veri Seti Parametre Ayarlamasız Elde Edilen Sonuçlar.

Boyut Azaltma	Çapraz Doğrulama	Doğruluk Oranı (%)	Standart Sapma (%)	Süre (s:d:s)
Hayır	Hayır	92,67	-	0:00:16.452
Hayır	Evet	92,23	0,41	0:01:30.442
Evet	Hayır	92,45	-	0:01:09.605
Evet	Evet	91,94	0,41	0:09:58.407

4.6. Destek Vektör Makineleri İçin Bulguların Değerlendirilmesi

Destek vektör makineleri kullanılarak oluşturulan modellerin test edilmesi için uygulamanın çalıştırılması sonucunda elde edilen ve karmaşıklık matrisi ile bu karmaşıklık matrisi kullanılarak hesaplanan performans ölçüm kriterlerini de göstermekte olan uygulama çıktısında bulunan bilgiler aşağıdaki çizelgelerde özetlenmektedir.

Çizelge 4.15'te destek vektör makineleri kullanılarak oluşturulan modelin veri setlerine ve bu veri setlerine boyut azaltma işlemi uygulanmasıyla elde edilen veri setlerine uygulanması sonucunda elde edilen performans kriterleri gösterilmektedir.

Söz konusu çizelgeden de görülebileceği üzere, doğruluk oranı ile diğer performans ölçüm kriterleri birbirleriyle uyumludur ve en yüksek doğruluk oranına veri sızıntısı olduğundan daha önce bahsedilmiş olan Zaman Değişkenli Veri Seti ile ulaşılmıştır ki aslında bu beklenilen bir durumdur ve esas itibariyle Zaman Değişkenli Veri Seti'nin test işlemlerine dahil edilmesinin sebebidir. Diğer taraftan, Temel Veri Seti ve Nihai Veri Seti için elde edilen doğruluk oranlarına bakıldığında yalnızca yaklaşık %0,06'lık gibi küçük bir farkla Temel Veri Seti için daha iyi doğruluk oranı elde edilmiştir. Bu küçük fark algoritmaların basitliği ve yorumlanabilirliği dikkate alındığında öznitelik belirleme aşamasının etkin olduğunu göstermektedir. Ayrıca, boyut azaltma işlemleri sonrasında Temel Veri Seti ve Nihai Veri Seti için doğruluk oranlarında önemli bir azalma görülmezken ilk aşamada yaklaşık 6 dakika olan çalışma süresinde yaklaşık 4 dakikalık bir iyileşme gözlemlenmiştir. Ayrıca, boyut azaltma işlemi sonrasında 3 değişkenli veri seti için AUC değeri 0,995222 olarak elde edilmiştir Dolayısıyla, öznitelik seçimi ve boyut azaltma işlemlerinin etkin bir sonuç verdiği değerlendirmesine ulaşılmıştır.

Çizelge 4.15 SVM-Çapraz Doğrulama Uygulanmadan Elde Edilen Sonuçlar.

Boyut Azaltma	Değişken Sayısı	Doğruluk	Hassasiyet	Duyarlılık	F1 Skoru	Süre (s:d:s)
Hayır	13	0,933383	0,904721	0,897938	0,898291	0:07:08.850
Hayır	7	0,939204	0,913707	0,908177	0,907677	0:05:23.130
Hayır	4	0,926268	0,894922	0,886295	0,886425	0:06:29.200
Hayır	3	0,926638	0,894921	0,886809	0,886718	0:06:03.504
Evet	13	0,926545	0,897205	0,889164	0,889547	0:02:52.429
Evet	3	0,921741	0,890542	0,881034	0,881376	0:02:38.460

Ek olarak, aynı veri setlerine aynı model 10-kat çapraz doğrulama yöntemi ile uygulanmış olup, elde edilen doğruluk oranlarına ve bu doğruluk oranlarına ilişkin standart sapma değerlerine aşağıdaki çizelge 4.16'da yer verilmektedir. Aşağıdaki çizelgeden de görülebileceği üzere beklendiği şekilde tüm veri setlerine ilişkin doğruluk oranında küçük bir azalma meydana gelmekte ancak bu durum modelin genelleştirilebilir olduğunu ifade etmektedir. Sonuç olarak, Temel Veri Seti ve Nihai Veri Seti arasında boyut azaltma işlemi sonucunda elde edilen doğruluk oranında önemli bir farklılık bulunmadığı ve her iki veri setinde de boyut azaltma işlemi uygulanması durumunda sonuçlarda önemli ölçüde bir kötüleşme meydana gelmediği, bununla birlikte çalışma süresi anlamında önemli kazanımlar elde edilebildiği anlaşılmıştır.

Çizelge 4.16 SVM-Çapraz Doğrulama Uygulanarak Elde Edilen Sonuçlar.

Boyut Azaltma	Değişken Sayısı	Doğruluk Oranı (%)	Standart Sapma (%)	Süre (s:d:s)
Hayır	13	92,92	0,46	0:49:13.243
Hayır	7	93,54	0,48	0:38:05.891
Hayır	4	92,32	0,47	0:44:51.702
Hayır	3	92,30	0,46	0:41:52.838
Evet	13	92,51	0,45	0:17:56.287
Evet	3	91,77	0,43	0:17:39.803

Diğer taraftan, parametre ayarlaması ile algoritmaların parametre ayarı yapılmadan oluşturulan modellerde kullanılan parametreler elde edilmiş olduğundan parametre ayarlarının bu model açısından herhangi bir etkisi bulunmamaktadır. Dolayısıyla, parametre ayarlaması ile kullanılan ilk parametre değerlerinin uygun olduğu teyit edilmiştir.

Son olarak, "rbf" çekirdek fonksiyonu kullanılarak geliştirilen doğrusal olmayan modelde Nihai Veri Seti'nde boyut azaltma işlemi uygulanması sonrasında %91,94 oranında doğruluk oranı elde edilmesine rağmen çalışma süresinin yaklaşık 45 dakika olduğu görülmüştür. Yani doğrusal olmayan model ile doğrusal olan modelde elde edilen doğruluk oranının (%91,77) yalnızca %0,17 fazlasına yaklaşık 3 kat sürede ulaşıldığı tespit edilmiştir.

4.7. Yapay Sinir Ağları İçin Bulguların Değerlendirilmesi

Yapay sinir ağı kullanılarak oluşturulan modellerin test edilmesi için uygulamanın çalıştırılması sonucunda elde edilen ve karmaşıklık matrisi ile bu karmaşıklık matrisi kullanılarak hesaplanan performans ölçüm kriterlerini de göstermekte olan uygulama çıktısında bulunan bilgiler aşağıdaki çizelgelerde özetlenmektedir.

Çizelge 4.17'de yapay sinir ağı kullanılarak oluşturulan modelin veri setlerine ve bu veri setlerine boyut azaltma işlemi uygulanmasıyla elde edilen veri setlerine uygulanması sonucunda elde edilen performans kriterleri gösterilmektedir.

Söz konusu çizelgeden de görülebileceği üzere doğruluk oranı ve diğer performans kriterleri arasında uyum bulunmakta olup, Temel Veri Seti, Belirlenmiş Veri Seti ve Nihai Veri Seti için birbirlerine yakın doğruluk oranları elde edilmiştir. Ayrıca, boyut azaltma işlemi neticesinde doğruluk oranlarında kayda değer bir azalma görülmezken çalışma süresinin azda olsa arttığı gözlemlenmiştir. Böylece, öznitelik

belirleme ve boyut azaltma işlemlerinin yerinde olduğu değerlendirmesine ulaşılmıştır.

Çizelge 4.17 ANN-Çapraz Doğrulama Uygulanmadan Elde Edilen Sonuçlar.

Boyut Azaltma	Değişken Sayısı	Doğruluk	Hassasiyet	Duyarlılık	F1 Skoru	Süre (s:d:s)
Hayır	13	0,937633	0,912035	0,904750	0,906439	0:00:27.418
Hayır	7	0,944563	0,923547	0,917206	0,917896	0:00:23.938
Hayır	4	0,931904	0,904126	0,897272	0,899080	0:00:22.976
Hayır	3	0,930518	0,902572	0,894170	0,895163	0:00:23.185
Evet	13	0,931165	0,905658	0,896693	0,897535	0:00:26.978
Evet	3	0,924235	0,894511	0,884544	0,885241	0:00:23.144

Yukarıda yer alan sonuçlar dikkate alındığında ulaşılan durumun tesadüf olup olmadığı hakkında değerlendirme yapılabilmesi amacıyla aynı veri setlerine aynı model 10-kat çapraz doğrulama yöntemi ile uygulanmıştır. Bu işlem sonucunda elde edilen doğruluk oranlarına ve bu doğruluk oranlarına ilişkin standart sapma değerlerine aşağıdaki çizelge 4.18'de yer verilmekte olup, yukarıda bahsedilen sonuçlarla çelişecek bir bulguya rastlanmamıştır.

Çizelge 4.18 ANN-Çapraz Doğrulama Uygulanarak Elde Edilen Sonuçlar.

Boyut Azaltma	Değişken Sayısı	Doğruluk Oranı (%)	Standart Sapma (%)	Süre (s:d:s)
Hayır	13	93,20	0,21	0:03:12.971
Hayır	7	93,95	0,37	0:03:03.660
Hayır	4	92,44	0,41	0:03:01.354
Hayır	3	92,40	0,22	0:02:59.224
Evet	13	92,65	0,30	0:02:41.386
Evet	3	91,97	0,34	0:02:42.524

Son olarak, parametre ayarlaması yapılmamış yapay sinir ağı ile oluşturulan modele ait doğruluk oranlarına ve ilgili standart sapmalara aşağıda yer verilmekte olup doğruluk oranında ve özellikle çalışma süresinde iyileşmelerin meydana geldiği

gözlemlenmiştir. Örnek olarak, boyut azaltma işlemi uygulanmış Nihai Veri Seti için 10-kat çapraz doğrulama ile çalıştırılan parametre ayarlı ve parametre ayarsız modeller karşılaştırıldığında parametrelerin ayarlanması sonucunda doğruluk oranında %0,44'lük bir artış, çalışma süresinde ise yaklaşık 12 dakikalık bir iyileşme olduğu gözlemlenmiştir.

Çizelge 4.19 ANN-Nihai Veri Seti Parametre Ayarlamasız Elde Edilen Sonuçlar.

Boyut Azaltma	**Çapraz Doğrulama**	**Doğruluk Oranı (%)**	**Standart Sapma (%)**	**Süre (s:d:s)**
Hayır	Hayır	92,70	-	0:01:40.933
Hayır	Evet	92,16	0,32	0:14:52.116
Evet	Hayır	92,12	-	0:01:40.215
Evet	Evet	91,53	0,27	0:14:17.233

5. SONUÇ ve ÖNERİLER

Terörizm günümüz dünyasının en önemli sorunlarından biridir ve terörizme müdahale, birçok ülke için ulusal güvenliğin ana gündemlerinden biri haline gelmiştir. Terörizmle mücadelede farklı tedbirler uygulanmakla birlikte gerçekleşen saldırıların faillerinin tespit edilmesi ve gelecekteki saldırıları önlemek için teknikler ve taktikler geliştirilmesi en somut tedbirlerdendir. Dolayısıyla, ülkelerin terörle mücadeleden sorumlu kurumları tarafından, fail grupları bastırmak için gerekli önlemlerin alınabilmesi ve önlemler sayesinde terörizm riskinin azaltılabilmesi için terör saldırılarından sorumlu terörist grupların etkin bir şekilde belirlenmesi önem arz etmektedir. Bahse konu durum dikkate alınarak bu çalışmada, Küresel Terörizm Veritabanı ve çeşitli makine öğrenmesi algoritmaları kullanılarak oluşturulan modeller aracılığı ile söz konusu verilerin sınıflandırılması suretiyle herhangi bir saldırıdan sorumlu olabilecek terörist grubun mümkün olan en yüksek doğrulukla tahmin edilebilmesine yönelik bir uygulama geliştirilmiş ve bu uygulamanın çıktıları doğruluk oranını, hassasiyet, duyarlılık, f1 skoru, çalışma süresi gibi performans ölçüm kriterlerine göre analize edilmiştir. Uygulamanın geliştirilmesi için *Python* programlama dili tercih edilmiş ve geliştirilen uygulama kapsamında;

- Veri setinin analiz edilmesi ve veri temizle işleminin uygulanması,
- Veri sızıntısı bulunup bulunmadığının incelenmesi,
- Özniteliklerin belirlenmesi,
- Test işlemlerinin gerçekleştirilebilmesi için Temel Veri Seti, Zaman Değişkenli Veri Seti, Belirlenmiş Veri Seti ve Nihai Veri Seti'nin oluşturulması,
- Modellerin yorumlanabilirliğinin artırılması için boyut azaltma yöntemi olarak temel bilişen analizinin uygulanması,
- Modellerin performansının artırılabilmesi için ızgara araması yöntemi ile parametrelerin ayarlanması,
- Zaman Değişkenli Veri Seti ile zaman değişkenlerinde bulunan veri sızıntısı ve genelleştirme probleminin gösterilmesi,

- Temel Veri Seti, Belirlenmiş Veri Seti ve Nihai Veri Seti ile öznitelik belirleme ve boyut azaltma gibi tekniklerin etkinliğinin gösterilmesi

işlemleri gerçekleştirilmiş olup, analizler sonucunda elde edilen bulgular ile ilgili tartışmalara bir önceki bölümde yer verilmiştir. Bu bulgular çerçevesinde, doğruluk oranı ve diğer performans ölçüm kriterlerinin birbirleriyle tutarlı olduğu ve çalışma kapsamında uygulanan yöntemlerin etkin sonuç verdiği anlaşılmış olup, modellerin geliştirilmesinde kullanılan algoritmaların karşılaştırılması Nihai Veri Seti üzerinden 10-kat çapraz doğrulama yöntemi ile elde edilen doğruluk oranına göre yapılmıştır. Söz konusu karşılaştırma sonuçlarına aşağıda çizelge 5.1'de yer verilmektedir.

Çizelge 5.1 Algoritmaların Karşılaştırılması.

Algoritma Adı	Doğruluk Oranı (%)	Standart Sapma (%)	Süre (s:d:s)
Naive Bayes	79.99	0.59	0:00:01.168
Lojistik Regresyon	91.87	0.39	0:01:45.242
K-En Yakın Komşu	89.84	0.37	0:01:25.967
Karar Ağacı	91.16	0.40	0:01:20.609
Rastgele Orman	92.00	0.43	0:08:29.807
Destek Vektör Makineleri Doğrusal	91.77	0.43	0:17:39.803
Destek Vektör Makineleri Doğrusal Olmayan	91.94	0.34	0:46:51.997
Yapay Sinir Ağı	91.97	0.34	0:02:42.524

Yukarıda yer alan çizelgede de görülebildiği üzere; rastgele orman algoritması ile en yüksek doğruluk oranı elde edilmiştir. Bu algoritmanın çalışma süresi her ne kadar yaklaşık 8 dakika görünse de boyut azaltma işlemi uygulanmadığı durumda doğruluk oranında kötüleşmeye sebep olmadan bu sürenin yaklaşık 4 dakikaya indiği görülmüştür. Rastgele orman algoritmasından sonra ise en yüksek doğruluk oranına yapay sinir ağları ile ulaşılmış olup yapay sinir ağlarının çalışma süresi bakımından rastgele orman algoritmasına kıyasla daha başarılı olduğu görülmüştür. Bahse konu bu algoritmalara yakın doğruluk oranı (%91,87) yaklaşık 1 dakika 45 saniye gibi bir

çalışma süresinde lojistik regresyon algoritması ile elde edilmiştir. Karar ağacı algoritması ile de %91,16 doğruluk oranı elde edilmiş ve bu doğruluk oranının boyut azaltma uygulanmadığı durumda %92,07'ye çıktığı görülmüştür ki rastgele orman algoritması ile birlikte karar ağacı algoritmasında boyut azaltmanın diğer algoritmaların aksine çalışma süresinin iyileştirilmesine katkı sağlamamış olduğu dikkate alındığında bu doğruluk oranın değerlendirilmesi yerinde olacaktır. *Naive Bayes* algoritmasının kullanılması durumunda ise diğer algoritmalara kıyasla oldukça düşük bir doğruluk oranı (%79,99) elde edilmiştir. Bununla birlikte destek vektör makinelerinin iyi performans gösteren algoritmalara yakın doğruluk oranı elde edebilmesine karşın oldukça kötü çalışma süresine sahip olduğu anlaşılmıştır.

Bu kapsamda, yaklaşık 1 ila 4 dakika arasında %91,87 ve 92,29 bandında doğruluk oranı elde edilebilmiş olup bu anlamda rastgele orman algoritması, yapay sinir ağları, karar ağacı algoritması ve lojistik regresyon algoritmasının bu tez çalışmasında açıklanan yöntemler yerine getirilerek herhangi bir terör olayı gerçekleştikten sonra sorumlu olabilecek grupların tahmin edilmesi konusunda uygulanabilir olduğu sonucuna ulaşılmıştır.

5.1. Değerlendirme

Bu çalışma kapsamında, bir terör olayı meydana geldikten sonra olayın hangi terörist grup tarafından yapıldığına ilişkin tahminde bulunulmasına yönelik bir uygulama geliştirilmiş olup, söz konusu uygulamada yapay sinir ağları ile rastgele orman, karar ağacı ve lojistik regresyon algoritmaları gerek performans ölçüm kriterleri gerekse de çalışma süresi dikkate alındığında öne çıkmıştır. Ayrıca, çalışma kapsamında uygulanan veri seti temizleme, veri sızıntısı inceleme, öznitelik belirleme, boyut azaltma ve parametre ayarlama işlemlerinin gerçekleştirilmesi sonucunda modellerin daha tutarlı, genelleştirilebilir ve uygulanabilir hale gelmesi sağlanmış olup bu durum 10-kat çapraz doğrulama yönteminin uygulanması ile gözlemlenmiştir.

Bu kapsamda, örnek bir veri seti üzerinden sorumlu grupları ve ilgili gerçekleri bulacak modellerin uygulanıp test edilmesiyle ham verilerin güvenlik uzmanlarının çalışmalarında kullanılabilecek yararlı bilgiye dönüştürülebileceği ve bu bilgiler doğrultusunda tahminlerde bulunulmasıyla incelemelerin yönlendirilebileceği değerlendirilmektedir. Bu bağlamda, devlet kurumları veya ilgili kuruluşlar tarafından Küresel Terörizm Veritabanı'na benzer veri tabanlarının oluşturulması durumunda bu tez çalışmasında açıklanan yöntemlere benzer yöntemler uygulanarak makine öğrenmesi algoritmalarının kullanılmasıyla oluşturulacak modeller aracılığı ile gerçekleştirilecek veri sınıflandırma işleminin;

- Güvenlik güçlerine ve ilgili kurumlara terörizmle mücadelede ve adli olaylarda incelemelerin yönlendirilebilmesi açısından katkı sağlanması,
- İstihbaratçılar tarafından toplanan bilgilerin hızlı ve etkin bir şekilde analiz edilmesi ve ham verilerin yasal kovuşturma açısından uygun bir istihbarat olarak sunulması,
- Kritik bilgi sistemlerinin maruz kaldığı saldırıların tespit edilmesi ve etkisinin azaltılması

gibi amaçlarla kullanılabileceği değerlendirilmektedir.

ÖZET

Adli Bilişimde Makine Öğrenmesi: Makine Öğrenmesi Algoritmaları İle Terör Olaylarının Tahmin Edilmesi Çalışması

Günümüz medeniyetine yönelik en önemli tehditlerden ve insanoğlunun karşı karşıya kaldığı en önemli sorunlardan biri terörizmdir. Terörizm, can kaybına sebep olmakta, ayrıca ülkelerin iş gücü, altyapı ve güvenlik harcamalarının artması sonucunu doğurmaktadır. Bu sebeple, terörizme müdahale, birçok ülke için ulusal güvenliğin ana gündemlerinden biri haline gelmiştir. Aynı zamanda, bilgi teknolojilerinin gelişmesiyle birlikte öngörülerde bulunarak stratejik kararlar alabilecek uygulamaların sağlık, ekonomi, güvenlik ve istihbarat gibi çok çeşitli alanlarda karar alınması amacıyla kullanılması mümkün hale gelmiştir.

Bu çalışmada, bir terör olayı meydana geldikten sonra olayın hangi terörist grup tarafından yapıldığına ilişkin tahminde bulunulması ve incelemelerin bu doğrultuda detaylandırılabilmesi amacıyla Küresel Terörizm Veritabanı üzerinde çeşitli makine öğrenmesi algoritmaları ile sınıflandırma işlemi gerçekleştirebilen modeller geliştirilmiştir. Çalışma kapsamında, modellerin tutarlı ve uygulanabilir hale getirilmesi amacıyla veri seti temizleme, veri sızıntısı inceleme, öznitelik belirleme, boyut azaltma ve parametre ayarlama işlemleri uygulanmış ve böylelikle örnek bir veri seti üzerinden ham verilerin güvenlik uzmanlarının çalışmalarında kullanılabilecek yararlı bilgiye dönüştürülmesi ve bu bilgiler doğrultusunda tahminlerde bulunulabilmesi aşamaları gösterilmiştir.

Modellerle elde edilen sonuçların değerlendirilmesi neticesinde, yapay sinir ağları ile rastgele orman, karar ağacı ve lojistik regresyon algoritmalarının doğruluk oranları ve çalışma süreleri dikkate alındığında bahse konu sınıflandırma işlemi için uygun olduğu anlaşılmıştır. Bununla birlikte, Naive Bayes algoritmasının diğer algoritmalara kıyasla düşük bir doğruluk oranı elde ettiği, destek vektör makinelerinin ise iyi performans gösteren algoritmalara yakın bir doğruluk oranı elde etmesine rağmen çalışma süresinin çok fazla olduğu görülmüştür. Dolayısıyla, bu algoritmaların bahse konu sınıflandırma işlemi için uygun olmadığı anlaşılmıştır.

Anahtar Sözcükler: Makine Öğrenmesi, Sınıflandırma, Terör Olayları, Veri Ön İşleme, Yapay Sinir Ağları.

SUMMARY

Machine Learning in Computer Forensics: Prediction of Terrorist Events with Machine Learning Algorithms

Terrorism is one of the most important threats to today's civilization and one of the most important problems facing humanity. Terrorism causes loss of life and also results in an increase in labor, infrastructure and security expenditures of countries. For this reason, combating against to terrorism has become one of the main agendas of national security for many countries. At the same time, with the development of information technologies, it has become possible to use applications that can make strategic decisions by making predictions for decision-making in various fields such as health, economy, security and intelligence.

In this study, models that can classify the Global Terrorism Database with various machine-learning algorithms have been developed in order to predict by which terrorist group the incident was committed after a terrorist incident occurred and to detail the investigations accordingly. Within the scope of the study, dataset cleaning, data-leakage examination, attribute determination, dimension reduction and parameter adjustment processes were applied in order to make the models consistent and applicable. In this way, the stages of converting raw data to useful information that can be used in the work of security experts and making predictions in line with this information have been shown over a sample data set.

As a result of the evaluation of the results obtained with the models, it has been understood that artificial neural networks and random forest, decision tree and logistic regression algorithms are suitable for the aforementioned classification process, considering the accuracy rates and executaion times. On the other hand, it has been observed that the Naive Bayes algorithm achieves a low accuracy rate compared to other algorithms and support vector machines have a very long executaion time despite achieving an accuracy rate close to well-performing algorithms. Therefore, it has been understood that these algorithms are not suitable for the mentioned classification process.

Keywords: Artificial Neural Networks, Classification, Data Preprocessing, Machine Learning, Terrorist Incidents.

KAYNAKLAR

ADAMSSON H (2017). Classification of Illegal Advertisement. Uppsala Universitet.

AGARWAL P, SHARMA M, CHANDRA S (2019). Comparison of Machine Learning Approaches in the Prediction of Terrorist Attacks. *Proceedings of the 2019 Twelfth International Conference on Contemporary Computing*, 1-7.

ALPAYDIN E (2004). Introduction to Machine Learning, second edition. The MIT Press.

BARKHAU J (2017). *Trend Detection among Terrorist Incidents in the Global Terrorism Database.* Tilburg: Tilburg University.

BLUM A (2021). Machine learning theory. Erişim Adresi: [www.cs.cmu.edu: http://www.cs.cmu.edu/afs/cs/user/avrim/www/Talks/mlt.pdf]. Erişim Tarihi: 07/01/2021.

BROWNLEE J (2021). Erişim Adresi: [https://machinelearningmastery.com/. Data Leakage in Machine Learning: https://machinelearningmastery.com/data-leakage-machine-learning/]. Erişim Tarihi: 07/01/2021.

CHRISTIE E (2019). Understanding the dynamics of unclaimed terrorism events in Pakistan: a machine learning approach. Orono: The University of Maine.

DANDIL E, CEVIK K K (2012). Yapay sinir ağları için net platformunda görsel bir eğitim yazılımının geliştirilmesi. *Bilişim Teknolojileri Dergisi,* **5:** 19-28.

DANG R, NILSSON A (2020). Evaluation of Machine Learning classifiers for Breast Cancer Classification. Royal Institute of Technology KTH.

FAWCETT T (2006). An introduction to ROC analysis. Pattern Recognition Letters, 861-874.

FENG, Y, WANG D, YIN Y, LI Z, HU Z (2020). An XGBoost-based casualty prediction method for terrorist attacks. *Complex and Intelligent Systems* , 721–740.

FİLİZ E (2019). Makine Öğrenmesi Yöntemleri ve Eğitim Verisi Üzerine Bir Uygulama: Uluslararası Matematik ve Fen Eğilimleri Araştırması 2015 Türkiye Örneği. Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü.

GOHAR F, HAİDER W, QAMAR U (2014). Terrorist group prediction using data classification . *e International Conference on Artificial Intelligence and Pattern Recognition*, 199-208.

GUYON I, ELISSEEFF A (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research 3*, 1157-1182.

H.JOHN R K (1997). Wrappers for feature subset selection. Artificial Intelligence, 273-324.

HARRINGTON P (2012). Machine Learning in Action. Manning Publications Shelter Island.

HASTIE T, TIBSHIRANI R, FRIEDMAN J (2001). The Elements of Statistical Learning. Springer.

IQBAL R, MURAD M A, MUSTAPHA A, PANAHY P H, KHANAHMADLIRAVI N (2013). An Experimental Study of Classification Algorithms for Crime Prediction. *Indian Journal of Science and Technology*, 1-7.

JAPKOWICZ N (2011). Performance Evaluation for Learning Algorithms. Cambridge University Press.

KALAIARASI S, MEHTA A, BORDIA D, SANSKAR (2019). Using Global Terrorism Database (GTD) and Machine Learning Algorithms to Predict Terrorism and Threat. *International Journal of Engineering and Advanced Technology*, 5995-6000.

KARABAY B (2019). Yaşanan Terör Olaylarını İçeren Büyük Verinin Makine Öğrenmesi Teknikleri ile Analizi. Fırat Üniveristesi Fen Bilimleri Enstitüsü.

KARTAL E (2015). Sınıflandırmaya Dayalı Makine Öğrenmesi Teknikleri ve Kardiyolojik Risk Değerlendirmesine İlişkin Bir Uygulama. İstanbul Üniversitesi Fen Bilimleri Enstitüsü.

KHORSHID M M, ABOU-EL-ENIEN T H, SOLIMAN G M (2015). A Comparison among Support Vector Machine and other Machine Learning Classification Algorithms. *IPASJ International Journal of Computer Science*, 25-35.

KRIEGER T, MEIERRIEKS D (2011). What causes terrorism? *Public Choice*, 3-27.

KUMAR V, MAZZARA M, GEN M, MESSINA A, YOUNG J (2019). A Conjoint Application of Data Mining Techniques for Analysis of Global Terrorist Attacks. *Advances in Intelligent Systems and Computing* (s. 146-158). içinde Springer Nature.

LECUN Y, BENGIO Y, Hinton G (2015). Deep Learning. *Nature*, 436-444.

LI Z, SUN D, LI B, LI Z, LI A (2018). Terrorist group behavior prediction by wavelet transform-based pattern recognition. *Discrete Dynamics in Nature and Society*, 1 - 16.

MAKINIST S (2018) Erişim Adresi: [Büyük Veri ve Yapay Zeka Laboratuvarı. Derin Öğrenme (Yapay Sinir Ağları-3): http://buyukveri.firat.edu.tr/2018/04/16/derin-ogrenme-yapay-sinir-aglari-3/]. Erişim Tarihi: 07/01/2021.

MANIRAJ S P, CHAUDHARY D, DEEP V H, SINGH V P (2019). Data aggregation and terror group prediction using machine learning algorithms. *International Journal of Recent Technology*, 1467-1469.

MIKSOVSKY P, MATOUSEK K, KOUBA Z (2002). Data pre-processing support for data mining. *IEEE International Conference on Systems, Man and Cybernetics*, Yasmine *Hammamet*, 4.

MO H, MENG X, LI J, ZHAO S (2017). Terrorist event prediction based on revealing data. *Proceedings of the IEEE 2nd International Conference on Big Data Analysis, Beijing*, 239-244.

MOREIRA C (2011). Learning to rank academic experts. Technical University of Lisbon.

NATH S V (2006). Crime Pattern Detection Using Data Mining. *International Conference on Web Intelligence and Intelligent Agent Technology Workshops, Hong Kong*, 41-44.

OZGUL F, ERDEM Z, BOWERMAN C (2009). Prediction of Unsolved Terrorist Attacks Using Group Detection Algorithms. *Intelligence and Security Informatics, Berlin*, 25-30.

PARR T, TURGUTLU K, CSISZAR C, HOWARD J (2021). *Beware Default Random Forest Importances*. Erişim Adresi [https://explained.ai/rf-importance/index.html]. Erişim Tarihi: 07/01/2021.

PILLAI I, FUMERA G, ROLI F (2017). Designing multi-label classifiers that maximize F measures: State of the art. *Pattern Recognition*, 394-404.

ROSENBLATT F (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review 65*, **6:** 386-408.

RUDER S, PLANK B (2018). Strong Baselines for Neural Semi-Supervised Learning under Domain Shift. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 1044-1054.

SAHA S, KURIAN A, BASU A (2017). Future Terrorist Attack Prediction using Machine Learning Techniques.

SAMUEL A (1988). Some studies in machine learning using the game of checkers, 210-229.

START (2019). Codebook: Inclusion Criteria And Variables. UNIVERSITY OF MARYLAND.

SUTHAR B, PATEL H, GOSWAMI A (2012). A Survey: Classification of Imputation Methods in Data Mining. *International Journal of Emerging Technology and Advanced Engineering*, 309-312.

TOURE I, GANGOPADHYAY A (2016). Real time big data analytics for predicting terrorist incidents. *2016 IEEE Symposium on Technologies for Homeland Security (HST), Waltham*, 1-6.

USTUNER M, SANLI F B (2020). Çok zamanlı polarimetrik SAR verileri ile tarımsal ürünlerin sınıflandırılması. *Jeodezi ve Jeoinformasyon Dergisi*, **7:** 6.

VERMA C, MALHOTRA S, VERMA V (2018). Predictive Modeling of Terrorist Attacks Using Machine Learning. *International Journal of Pure and Applied Mathematics*, 6.

EKLER

EK-1- GTD Veri Seti Özniteliklerine İlişkin Bilgiler.

Ana Grup	Değiken	GTD İndeks	AÇIKLAMA
I. GTD ID and Date	eventid	0	Birbirinden eşsiz olarak üretilen, olaya ilişkin niteleyici bir bilgi içermeyen ve yalnızca olayların veri tabanında takip edilebilmesi için kullanılmakta olan kimlik bilgisidir.
	iyear	1	Olayın meydana geldiği yılı göstermektedir. Uzun bir süreye yayılan olaylarda ise olayın başlangıç yılını göstermektedir.
	imonth	2	Olayın meydana geldiği ayı göstermektedir. Uzun bir süreye yayılan olaylarda ise olayın başlangıç ayını göstermektedir. 1970 ile 2011 yılları arasında gerçekleşen saldırılar için, olayın tam ayı bilinmiyora ise "0" olarak kaydedilir. 2011'den sonra gerçekleşen olaylar içinse approxdate alnından elde edilen orta nokta değeri kaydedilir.
	iday	3	Olayın meydana geldiği günü göstermektedir. Uzun bir süreye yayılan olaylarda ise olayın başlangıç gününü göstermektedir. 1970 ile 2011 yılları arasında gerçekleşen saldırılar için, olayın tam günü bilinmiyora ise "0" olarak kaydedilir. 2011'den sonra gerçekleşen olaylar içinse approxdate alnından elde edilen orta nokta değeri kaydedilir.
	approxdate	4	Olayın kesin tarihi bilinmediğinde veya belirsiz kaldığında, bu alan olayın yaklaşık tarihini metin olarak kaydetmek için kullanılmaktadır. Metin olarak çok farklı değerler içerebilmekte olup, ayrıca, bu alandaki veriler kullanılarak iyear, imonth, iday alanlarında ilgili güncelleştirmeler yapılmış durumdadır.
	extended	5	Bir olayın süresinin 24 saatten fazla olup olmadığını göstermek için kullanılmakta olup, 24 saatten fazla süren olaylar için 1 kısa süren olaylar için 0 olarak kaydedilmektedir.
	resolution	6	187011 kayıt için boş. Extended=1 olan olaylar için olayın çözümlendiği tarihin nümerik olarak kaydedilmesi için kullanılmaktadır.
II. Incident Information	summary	18	1997 yılından sonra meydana gelen olayların kısa anlatımını metin olarak kaydetmek amacıyla kullanılmaktadır. Dolayısıyla metin olarak çok farklı değerler içermektedir.
	crit1	19	İlgili olayın veritabanına dahil edilme kriterini gösterir. Bu alanın 1 olması, ilgili olayın, politik, ekonomik, dini veya sosyal bir hedefe ulaşmayı amaçladığından veri tabanına dahil edildiğini ifade etmektedir..
	crit2	20	İlgili olayın veritabanına dahil edilme kriterini gösterir. Bu alanın 1 olması, ilgli olayn daha geniş bir kitleye zorlama, gözdağı verme veya başka bir mesajı iletme niyetini barındırması sebebiyle veri tabanına dahil edildiğini ifade etmektedir
	crit3	21	İlgili olayın veritabanına dahil edilme kriterini gösteririr. Bu alanın 1 olması, ilgili olayın meşru savaş faaliyetleri ve uluslararası insanı hukuk dışında

			olmasından dolayı veri tabanına kaydedildiğini ifade etmektedir.
	doubtterr	22	Bir olayın dahil edilme kriterlerinin tümünü karşılayıp karşılamadığı konusunda belirsizlik olması ve ilgili olayın veri tabanına dahil edilmesi durumunda bu değişken için “1” olarak kodlanır. Yalnızca 1997 yılından sonra meydana gelen olaylar için mevcut olan bu alan, diğer durumlar için "-9" olarak kodlanmıştır.
	alternative	23	Yalnızca doubtterr alanı "1" olarak kodlanan olaylar için uygulanmaktadır ve 1997 yılından sonra meydana gelen olaylar için mevcuttur. Olayın terörizm dışındaki en olası kategorizasyonunu tanımlamak için kullanılmaktadır. 160394 adet olay için "boş" olarak kodlanmış olduğundan niteleyici bir özelliği bulunmamaktadır.
	alternative_txt	24	Bu alan, kategorik değişken olan "alternative" alanının metin karşılığını temsil etmektedir.
	multiple	25	Kategorik değişken olan bu alan, eylemlerin ayrı ayrı tek bir olay oluşturmadığı ancak birkaç saldırının bağlı olduğu durumlarda, söz konusu saldırının "çoklu" bir olayın parçası olduğunu belirtmek için kullanılır.Yalnızca 1997 yılından sonra meydana gelen olaylar için mevcuttur. 164771 adet olay için "0" olarak kodlanmış olduğu dikkate alındığında niteleyici bir özelliği bulunmamaktadır.
	related	134	Bir saldırı koordineli, çok parçalı bir olayın parçası olduğunda yani multiple alanının "1" olarak kodlandığı durumlarda, ilişkili olayların GTD ID'leri virgülle ayrılmış olarak bu alanda kaydedilmektedir. Metin değişkeni olan bu alan yalnızca 1997 yılından sonra meydana gelen olaylar için mevcut. "Multiple" alanının niteleyici bir özelliği bulunmadığı dikkate alındığında bu alanın da niteleyici bir özelliği bulunmamaktadır.
III. Incident Location	country	7	Bu alan, olayın gerçekleştiği ülkeyi veya yeri tanımlamak için kullanılmaktadır. Bir olayın meydana geldiği ülkenin tespit edilememesi durumunda ise "Bilinmeyen" olarak kodlanmaktadır. Ayrıca, pek çok ülkenin jeopolitik sınırları zaman içinde değişmiş olduğundan, bazı durumlarda terörist saldırıların yerini temsil eden ülkeler artık mevcut değildir.
	country_txt	8	Bu alan, kategorik değişken olan "country" alanının metin karşılığını temsil etmektedir.
	region	9	Bu alan olayın gerçekleştiği bölgeyi tanımlar. Bölgeler, olay için kodlanan ülkeye bağlı olarak 12 kategoriye ayrılmıştır.
	region_txt	10	Bu alan, kategorik değişken olan "region" alanının metin karşılığını temsil etmektedir.
	provstate	11	Bu değişken, olayın gerçekleştiği 1. derece ulusal yönetim bölgesinin adını (olay zamanında) kaydeder. Olmayanlar için şehir ile aynı değeri almaktadır. Metin olarak çok farklı değerler içeren bu alan, özellikle "country" ve "region" kategorik değişkenlerinin olayın gerçekleştiği yere ilişkin niteleyici bilgileri içerdiği dikkate alındığında yalnızca açıklayıcı nitelikte bir bilgi içermektedir.
	city	12	Metin olarak çok farklı değerler içeren bu alan, özellikle "country" ve "region" kategorik değişkenlerinin olayın gerçekleştiği yere ilişkin niteleyici bilgileri içerdiği dikkate alındığında yalnızca açıklayıcı nitelikte bir bilgi içermektedir.

	latitude	13	Bu alan, olayın gerçekleştiği şehrin enlemini (WGS1984 standartlarına göre) kaydeder. Özellikle"country" ve "region" kategorik değişkenlerinin olayın gerçekleştiği yere ilişkin niteleyici bilgileri içerdiği dikkate alındığında açıklayıcı nitelikte bir bilgi içermektedir.
	longitude	14	Bu alan olayın gerçekleştiği şehrin boylamını (WGS1984 standartlarına göre) kaydeder. Özellikle"country" ve "region" kategorik değişkenlerinin olayın gerçekleştiği yere ilişkin niteleyici bilgileri içerdiği dikkate alındığında açıklayıcı nitelikte bir bilgi içermektedir.
	specificity	15	Bu alan enlem ve boylam alanlarının coğrafi uzamsal çözünürlüğünü tanımlamak için kullanılan açıklayıcı bir alandır.
	vicinity	16	Kategorik değişken olan bu alanın "1" olması, bahse konu olayın, söz konusu şehrin hemen yakınında meydana geldiğini göstermektedir. "0" olması ise söz konusu olayın şehrin kendisinde meydana geldiğini göstermektedir. 150103 adet olay için "0" olarak kodlandığı ve olayın meydana geldiği şehir bilgisinin veri setine dahil edildiği dikkate alındığında niteleyici bir özelliğinin bulunmadığı ve açıklayıcı bir alan olduğu değerlendirilmektedir.
	location	17	Metin değişkeni olarak çok farklı değerler içeren bu alan, olayın yeri hakkında ek bilgi belirtmek için kullanılmaktadır.
IV. Attack Information	success	26	Kategorik değişkin olan bu alan olayın başarılı olması durumunda "1" olarak kodlanmaktadır. Başarısız olması durumunda ise "0" olarak kodlanmaktadır. Dolayısıyla saldırının sonucuna ilişkin niteleyici bilgi içermektedir.
	suicide	27	Kategorik değişken olan bu alan, failin saldırıdan canlı olarak kaçmak niyetinde olmadığına dair kanıt bulunan durumlarda "1", diğer durumlarda ise "0" olarak kodlanmaktadır. Dolayısıyla saldırının türüne ilişkin niteleyici bilgi içermektedir.
	attacktype1	28	Kategorik değişken olan bu alan saldırı tipini, 9 kategoride kodlamak için kullanılmaktadır.Dolayısıyla saldırı türüne ilişkin niteleyici bilgi içermektedir.
	attacktype1_txt	29	Bu alan, kategorik değişken olan "attacktype1" alanının metin karşılığını temsil etmektedir.
	attacktype2	30	Veri tabanına her olay için en fazla üç saldırı türü kaydedilebilmektedir.Genel olarak, saldırı bir dizi olaydan oluşmadıkça her olay için yalnızca bir saldırı türü kaydedilmektedir. Bu alan sadece "multiple" olarak kodlanan olaylarda ikinci saldırı türünü kaydetmek için kullanılmakta olup, 184442 adet olay için herhangi bir kayıt yapılmamış durumdadır.
	attacktype2_txt	31	Bu alan, kategorik değişken olan "attacktype2" alanının metin karşılığını temsil etmektedir.
	attacktype3	32	"attacktype2" alanına benzer şekilde çoklu olaylarda olayın üçüncü saldırı türünü kaydetmek için kullanılmakta olup, 190980 adet olay için herhangi bir kayıt yapılmamıştır.
	attacktype3_txt	33	Bu alan, kategorik değişken olan "attacktype3" alanının metin karşılığını temsil etmektedir.
V. Weapon Information	weaptype1	81	Bir saldırıda kullanılan 4 silah türü ve 4 silah alt türü her olay için kaydedilmektedir.Ayrıca, her olay için rapor edilen özel silah ayrıntılarına ilişkin bilgiler de kaydedilmektedir. Kategorik değişken bu alanda ise

			saldırıda kullanılan genel silah türü 13 farklı kategori içerisinde kodlanmaktadır.
	weaptype1_txt	82	Bu alan, kategorik değişken olan "weaptype1" alanının metin karşılığını temsil etmektedir.
	weapsubtype1	83	Kategorik değişken bu alanda, "weaptype1" alanında kodlanan silah türüne ilişkin detaylar alt kırılım olarak kodlanmaktadır. Dolayısıyla, "weaptype1" alanı ile birlikte bu alan saldırıda kullanılan silah türüne ilişkin detaylı niteleyici bilgileri temsil etmektedir.
	weapsubtype1_txt	84	Bu alan, kategorik değişken olan "weapsubtype1" alanının metin karşılığını temsil etmektedir.
	weaptype2	85	177273 adet olay için herhangi bir kayıt yapılmamıştır. Dolayısıyla saldırıya ilişkin niteleyici bilgi taşıma özelliği bulunmamaktadır.
	weaptype2_txt	86	Bu alan, kategorik değişken olan "weaptype2" alanının metin karşılığını temsil etmektedir.
	weapsubtype2	87	"weaptype2" alanının içerdiği boş kayıtlar sebebiyle niteleyici bir özelliğinin olmamasından dolayı, onun alt kırılımını temsil eden ve yine doğal olarak bir çok boş kayıt içeren bu alanın da niteleyici bir özelliği bulunmamaktadır.
	weapsubtype2_txt	88	Bu alan, kategorik değişken olan "weapsubtype2" alanının metin karşılığını temsil etmektedir.
	weaptype3	89	189431 adet olay için herhangi bir kayıt yapılmamıştır. Dolayısıyla saldırıya ilişkin niteleyici bilgi taşıma özelliği bulunmamaktadır.
	weaptype3_txt	90	Bu alan, kategorik değişken olan "weaptype3" alanının metin karşılığını temsil etmektedir.
	weapsubtype3	91	"weaptype3" alanının içerdiği boş kayıtlar sebebiyle niteleyici bir özelliğinin olmamasından dolayı, onun alt kırılımını temsil eden ve yine doğal olarak bir çok boş kayıt içeren bu alanın da niteleyici bir özelliği bulunmamaktadır.
	weapsubtype3_txt	92	Bu alan, kategorik değişken olan "weapsubtype3" alanının metin karşılığını temsil etmektedir.
	weaptype4	93	191392 adet olay için herhangi bir kayıt yapılmamıştır. Dolayısıyla saldırıya ilişkin niteleyici bilgi taşıma özelliği bulunmamaktadır.
	weaptype4_txt	94	Bu alan, kategorik değişken olan "weaptype4" alanının metin karşılığını temsil etmektedir.
	weapsubtype4	95	"weaptype4" alanının içerdiği boş kayıtlar sebebiyle niteleyici bir özelliğinin olmamasından dolayı, onun alt kırılımını temsil eden ve yine doğal olarak bir çok boş kayıt içeren bu alanın da niteleyici bir özelliği bulunmamaktadır.
	weapsubtype4_txt	96	Bu alan, kategorik değişken olan "weapsubtype4" alanının metin karşılığını temsil etmektedir.
	weapdetail	97	Metin değişkeni olan bu alan, olayda kullanılan silahlara ilişkin detay bilgilerin metin olarak kaydedilmesi amacıyla kullanılmakta olup, çok farklı kayıtlar içermektedir.
VI. Target/Victim Information	targtype1	34	Her olay için en fazla 3 adet hedef/mağdur bilgisi kaydedilmektedir. Kategorik değişken bu alanda 22 farklı kategoride hedefe/mağdura ilişkin tip bilgisi kodlanmaktadır. Dolayısıyla hedef alınan kişiye ve saldırının amacına yönelik niteleyici bilgiler içerebilmektedir.
	targtype1_txt	35	Bu alan, kategorik değişken olan "targtype1" alanının metin karşılığını temsil etmektedir.
	targsubtype1	36	Kategorik değişken bu alanda, "targtype1" alanında kodlanan hedef/mağdur türüne ilişkin detaylar alt kırılım olarak kodlanmaktadır. Dolayısıyla,

			"targtype1" alanı ile birlikte bu alan saldırının hedefine ve amacına ilişkin detaylı niteleyici bilgileri temsil etmektedir.
	targsubtype1_txt	37	Bu alan, kategorik değişken olan "targsubtype1" alanının metin karşılığını temsil etmektedir.
	corp1	38	Bu alanda hedeflenen tüzel kişinin veya devlet kurumunun ismi metin olarak kodlanmaktadır. Hedeflenen öğe belirtilmemişse "bilinmiyor", saldırının bir hedefi yoksa "uygulanamaz" olarak kodlanmaktadır. Metin olarak çok farklı değerler içerebilmekte olup açıklayıcı bir özelliği bulunmaktadır.
	target1	39	Bu alanda hedelenen veya mağdur edilen ve "target1" alanında bahsedilen varlığın bir parçası olan araba, bina gibi genel isimlere yer verilmektedir. Bahse konu alanda 10.000'den fazla unique değer yer almaktadır. Dolayısıyla niteleyici bir özelliğinden ziyade açıklayıcı bir özelliği bulunmaktadır.
	natlty1	40	Bu alanda kategorik değişken olarak saldırıya uğrayan hedefin uyruğu kaydedilmektedir ve çoğu zaman olayın meydana geldiği ülkeden farklı olabilmektedir. Saldırıya ve hedefe ilişkin niteleyici bilgi içermektedir.Uyruk bilgisinin kodlanması için "country" alanında kullanılan ülke kodları kullanılmaktadır.
	natlty1_txt	41	Bu alan, kategorik değişken olan "natlty1" alanının metin karşılığını temsil etmektedir.
	targtype2	42	179269 adet olay için herhangi bir kayıt yapılmamıştır. Dolayısıyla saldırıya ilişkin niteleyici bilgi taşıma özelliği bulunmamaktadır.
	targtype2_txt	43	Bu alan, kategorik değişken olan "targtype2" alanının metin karşılığını temsil etmektedir.
	targsubtype2	44	"targtype2" alanının içerdiği boş kayıtlar sebebiyle niteleyici bir özelliğinin olmamasından dolayı, onun alt kırılımını temsil eden ve yine doğal olarak bir çok boş kayıt içeren bu alanın da niteleyici bir özelliği bulunmamaktadır.
	targsubtype2_txt	45	Bu alan, kategorik değişken olan "targsubtype2" alanının metin karşılığını temsil etmektedir.
	corp2	46	"corp1" alanı ile aynı özellikleri göstermekte olduğundan benzer şekilde yalnızca açıklayıcı özelliği bulunmaktadır.
	target2	47	"target1" alanı ile aynı özellikleri göstermekte olduğundan benzer şekilde yalnızca açıklayıcı özelliği bulunmaktadır.
	natlty2	48	179586 adet olay için herhangi bir kayıt yapılmamıştır. Dolayısıyla saldırıya ilişkin niteleyici bilgi taşıma özelliği bulunmamaktadır.
	natlty2_txt	49	Bu alan, kategorik değişken olan "natlty2" alanının metin karşılığını temsil etmektedir.
	targtype3	50	190152 adet olay için herhangi bir kayıt yapılmamıştır. Dolayısıyla saldırıya ilişkin niteleyici bilgi taşıma özelliği bulunmamaktadır.
	targtype3_txt	51	Bu alan, kategorik değişken olan "targtype3" alanının metin karşılığını temsil etmektedir.
	targsubtype3	52	"targtype3" alanının içerdiği boş kayıtlar sebebiyle niteleyici bir özelliğinin olmamasından dolayı, onun alt kırılımını temsil eden ve yine doğal olarak bir çok boş kayıt içeren bu alanın da niteleyici bir özelliği bulunmamaktadır.
	targsubtype3_txt	53	Bu alan, kategorik değişken olan "targsubtype3" alanının metin karşılığını temsil etmektedir.

	corp3	54	"corp1" alanı ile aynı özellikleri göstermekte olduğundan benzer şekilde yalnızca açıklayıcı özelliği bulunmaktadır.
	target3	55	"target1" alanı ile aynı özellikleri göstermekte olduğundan benzer şekilde yalnızca açıklayıcı özelliği bulunmaktadır.
	natlty3	56	190181 adet olay için herhangi bir kayıt yapılmamıştır. Dolayısıyla saldırıya ilişkin niteleyici bilgi taşıma özelliği bulunmamaktadır.
	natlty3_txt	57	Bu alan, kategorik değişken olan "natlty3" alanının metin karşılığını temsil etmektedir.
VII. Perpetrator Information	gname	58	Her olay için en fazla 3 adet fail bilgisi kaydedilmektedir. Metin değişkeni olan bu alan saldırıyı gerçekleştiren grubun adını kaydetmek için kullanılmaktadır. Veri tabanında tutarlığı sağlamak adına bu alanda standartlaştırılmış grup adları kullanılmaktadır. Kaynak materyallerde resmi bir fail grup veya kuruluşun adının bildirilmemesi durumunda, bu alan failin jenerik kimliği hakkındaki bilgileri içerebilmektedir. Fail grup hakkında herhangi bir bilgi yoksa, bu alan "bilinmiyor" olarak kodlanmaktadır. Standartlaştırılmış grup adlarının kullanılmakta olduğu ve saldırının ana unsurlarından olan fail bilgisini belirtiği dikkate alınarak olaylar hakkında niteleyici bilgi içerdiği değerlendirilmektedir. Ayrıca, makine öğrenmesi algoritmalarında hedef değişken olarak kullanılması uygun görülmektedir.
	gsubname	59	Bu alan, saldırıyı gerçekleştiren grup hakkında ek ayrıntılar içernektedir. Saldırıyı gerçekleştiren grup hedef değişken olabileceği için input variable olarak yer almasına gerek olmadığı değerlendirilmektedir. Ayrıca metin değişken olan bu alanda ayrıntı bilgisine yer verilmekte olduğundan çok farklı değerler bulunması da söz konusudur.
	gname2	60	Bu alan, saldırı sorumluluğun birden fazla faile atfedildiğin durumlarda ikinci failin adının kaydedilmesi amacıyla kullanılmakta olup, 189249 adet olay için herhangi bir kayıt yapılmamıştır. Dolayısıyla saldırıya veya failine ilişkin niteleyici bilgi taşıma özelliği bulunmamaktadır.
	gsubname2	61	"gname2" alanının içerdiği boş kayıtlar sebebiyle niteleyici bir özelliğinin olmamasından dolayı, ona ilişkin ek bilgileri içeren ve yine doğal olarak bir çok boş kayıt içeren bu alanın da niteleyici bir özelliği bulunmamaktadır.
	gname3	62	Bu alan, saldırı sorumluluğun ikinden fazla faile atfedildiğin durumlarda üçüncü failin adının kaydedilmesi amacıyla kullanılmakta olup, 189249 adet olay için herhangi bir kayıt yapılmamıştır. Dolayısıyla saldırıya veya failine ilişkin niteleyici bilgi taşıma özelliği bulunmamaktadır.
	gsubname3	63	"gname3" alanının içerdiği boş kayıtlar sebebiyle niteleyici bir özelliğinin olmamasından dolayı, ona ilişkin ek bilgileri içeren ve yine doğal olarak bir çok boş kayıt içeren bu alanın da niteleyici bir özelliği bulunmamaktadır.
	motive	64	Bu alanda, kaynaklarda saldırı için belirli bir neden olduğundan açıkça bahsedildiğinde bahsedilen gerekçe kaydedilir. 139235 adet olay için herhangi bir kayıt yapılmamıştır. Dolayısıyla saldırıya ilişkin niteleyici bilgi taşıma özelliği bulunmamaktadır.

	guncertain1	65	Bu alan, kaynaklar tarafından bildirilen bilgilerin spekülasyona veya şüpheli sorumluluk iddialarına dayanıp dayanmadığını göstermek amacıyla kullanılmaktadır. Fail olunup olunmadığından şüphe duyulması durumunda bu alan "0" olarak kodlanmaktadır. Şüphe olmaması durumunda ise 0 olarak kodlanmaktadır. 176315 adet olay için "boş" veya "0" olarak kodlanmış olup, dağılımda dengesizlik içermektedir.
	guncertain2	66	Olayın ikinci failine ilişkin "guncertain1" alanı ile aynı kaydı tutmaktadır. "gname2" alanının içerdiği boş kayıtlar sebebiyle niteleyici bir özelliğinin olmamasından ve bu alanın da 190883 adet olay için "boş" veya "0" olarak kodlanmış olduğu dikkate alındığında olay hakkında herhangi bir niteleyici özellik taşımamadığı değerlendirilmektedir.
	guncertain3	67	Olayın üçüncü failine ilişkin "guncertain1" alanı ile aynı kaydı tutmaktadır. "gname3" alanının içerdiği boş kayıtlar sebebiyle niteleyici bir özelliğinin olmamasından ve bu alanın da 191394 adet olay için "boş" veya "0" olarak kodlanmış olduğu dikkate alındığında olay hakkında herhangi bir niteleyici özellik taşımamadığı değerlendirilmektedir.
	individual	68	Bu alan, saldırının bir grup ya da kuruluşa bağlı olduğu bilinmeyen kişi veya kişiler tarafından gerçekleştirilip gerçekleştirilmediğini göstermek amacıyla kullanılmaktadır. Kategorik değişken olan bu alan, yalnızca 1997 yılından sonra gerçekleşen olaylar için mevcut olup, faillerin grup veya kuruluşla bağlantılı olduklarının bilinmediği durumda "1" olarak kodlanmaktadır. Dolayısıyla, bu alan üzerinden olayın bireysel mi yoksa örgütsel mi olduğu bilgisine ulaşılması mümkün olduğundan olaya ilişkin niteleyici bilgi içermektedir.
	nperps	69	Bu alanda, nümerik olarak olaya katılan toplam terörist sayısı kaydedilmektedir.161386 adet olay için "bilinmiyor" veya "boş" olarak kayıt yapılmıştır. Ayrıca "Codebook'" dokümanında kaynaklarda bu değerle ilgili bilgilerde genellikle tutarsızlıklar olduğu ifade edilmektedir. Dolayısıyla olaya ilişkin niteleyici bilgi taşıma özelliği bulunmamaktadır.
	nperpcap	70	Bu alanda, nümerik olarak gözaltına alınan terörist sayısı kaydedilmektedir. 186835 adet olay için "bilinmiyor" ,"boş" veya "0" olarak kayıt yapılmıştır.Dolayısıyla olaya ilişkin niteleyici bilgi taşıma özelliği bulunmamaktadır.
	claimed	71	Bu alan, bir grubun veya kişinin / kişilerin saldırının sorumluluğunu üstlenip üstlenmediğini belirtmek için kullanılmaktadır. Bir kişinin veya grubun sorumluluğu üstlenmesi durumunda "1" olarak kodlanmaktadır. Dolayısıyla olayı gerçekleştiren faillerin özelliklerine yönelik niteleyici bilgi içermektedir. Ancak eylemlerin üstlenilme oranının %10 gibi küçük bir oranda olması nedeniyle veride dengesizlik bulunmaktadır. Ayrıca, bu alan yalnızca 1997 yılından sonra gerçekleşen olaylar için mevcuttur.
	claimmode	72	Kategorik değişken olan bu alan, faillerin sorumluluğu üstlenmek için hangi yöntemi kullandıklarını belirtmek için kullanılmaktadır. Ancak 169915 adet olay için herhangi bir kayıt

			yapılmamış olduğu dikkate alındığında niteleyici bir özelliği bulunmamaktadır.
	claimmode_txt	73	Bu alan, kategorik değişken olan "claimmode" alanının metin karşılığını temsil etmektedir.
	claim2	74	Bu alan, olaya ilişkin ikinci bir fail kişi veya grup olması durumunda sorumluluk üstlenilip üstlenilmediğini göstemek için kullanılmaktadır. 190905 adet olay için "bilinmiyor" veya "boş" olarak kayıt yapılmış olduğu dikkate alındığında niteleyici bir özelliği bulunmamaktadır.
	claimmode2	75	"claim2" alanının içerdiği boş kayıtlar sebebiyle niteleyici bir özelliğinin olmamasından dolayı, ona ilişkin sorumluluk üstlenme yöntemini gösteren ve yine doğal olarak bir çok boş kayıt içeren bu alanın da niteleyici bir özelliği bulunmamaktadır
	claimmode2_txt	76	Bu alan, kategorik değişken olan "claimmode2" alanının metin karşılığını temsil etmektedir.
	claim3	77	Bu alan, olaya ilişkin üçüncü bir fail kişi veya grup olması durumunda sorumluluk üstlenilip üstlenilmediğini göstemek için kullanılmaktadır. 191333 adet olay için "bilinmiyor" veya "boş" olarak kayıt yapılmış olduğu dikkate alındığında niteleyici bir özelliği bulunmamaktadır.
	claimmode3	78	"claim3" alanının içerdiği boş kayıtlar sebebiyle niteleyici bir özelliğinin olmamasından dolayı, ona ilişkin sorumluluk üstlenme yöntemini gösteren ve yine doğal olarak bir çok boş kayıt içeren bu alanın da niteleyici bir özelliği bulunmamaktadır
	claimmode3_txt	79	Bu alan, kategorik değişken olan "claimmode3" alanının metin karşılığını temsil etmektedir.
	compclaim	80	Bu alan, birden fazla grubun saldırı için ayrı sorumluluk talep edip etmediğini göstermek için kullanılmaktadaır.190997 adet olay için "bilinmiyor" veya "boş" olarak kayıt yapılmış olduğu dikkate alındığında niteleyici bir özelliği bulunmamaktadır.
VIII. Casualties and Consequences	nkill	98	Nümerik değişkin olan bu alanda, olay için saldırgan ölümleri de dahil olmak üzere teyit edilen toplam ölümlerin sayısı kaydedilmektedir. Dolayısıyla olayın büyüklüğü ve ilişkili örgüt hakkında niteleyici bilgi içerebilmektedir.
	nkillus	99	Nümerik değişken olan bu alanda, olay sonucunda ölen ABD vatandaşlarının sayısı kaydedilmekte olup, 190534 adet olay için "0" veya "boş" olarak kayıt yapılmış olduğu dikkate alındığında niteleyici bir özelliği bulunmamaktadır. Ayrıca, toplam ölüm sayısı nkill alanında zaten izlenmektedir.
	nkillter	100	Nümerik değişken olan bu alanda, yalnızca failin ölümleriyle sınırlı kayıtlar bulunmaktadır. 177012 adet olay için "0" veya "boş" olarak kayıt yapılmış olduğu dikkate alındığında niteleyici bir özelliği bulunmamaktadır. Ayrıca, toplam ölüm sayısı nkill alanında zaten izlenmektedir.
	nwound	101	Nümerik değişken olan bu alanda, hem faillere hem de mağdurlara yönelik olarak teyit edilen ölümcül olmayan yaralanmaların sayısını kaydedilmekte olup 125538 adet olay için "boş" veya "0" olarak kodlanmıştır.
	nwoundus	102	Nümerik değişken olan bu alanda, ABD vatandaşlarına özel olarak, hem faillere hem de mağdurlara yönelik, teyit edilen ölümcül olmayan yaralanmaların sayısını kaydedilmektedir.190780 adet olay için "0" veya "boş" olarak kayıt yapılmış olduğu dikkate alındığında niteleyici bir özelliği bulunmamaktadır

	nwoundte	103	Nümerik değişken olan bu alanda, faiilere özel olarak "nwound" alanı ile benzer kayıtları tutulmakta olup, 188605 adet olay için "0" veya "boş" olarak kayıt yapılmış olduğu dikkate alındığında niteleyici bir özelliği bulunmamaktadır
	property	104	Kategorik değişken olan bu alan, olaydan kaynaklanan maddi hasar olup olmadığını göstermek amacıyla kullanılmaktadır. Olayın maddi hasarla sonuçlanması durumunda "1" olarak kodlanmaktadır. Olayın niyetine veya ilişkili örgüte ilişkin niteleyici bilgi içerebilmektedir.
	propextent	105	Kategorik değişken olan bu alan, maddi hasarın kapsamını belirtmek için kullanılmaktadır. 124179 adet olay için "boş" ve 32869 adet olay için "bilinmiyor" olarak kaydedilmiş olmasından dolayı niteleyici özelliği bulunmamaktadır.
	propextent_txt	106	Bu alan, kategorik değişken olan "propextent" alanına ilişkin açıklayıcı bilgileri metin olarak kaydetmek için kullanılmakta olup, çok farklı değerler almaktadır. Ayrıca,"propextent" alanının niteleyici özelliği bulunmamaktadır.
	propvalue	107	Nümerik değişken olan bu alan, toplam hasarın ABD doları tutarını göstermekte olup, 181255 adet olay için "boş" veya "bilinmiyor" veya "0" olarak kaydedilmiş olduğu dikkate alındığında niteleyici bir özelliği bulunmamaktadır..
	propcomment	108	Bu alan, kategorik değişken olan "propextent" alanının metin karşılığını temsil etmektedir.
	ishostkid	109	Kategorik değişken olan bu alanda, mağdurların rehin alınıp alınmadığı veya kaçırılıp kaçırılmadığı bilgisi kaydedilmektedir. Mağdurların rehin alınması veya kaçırılması durumunda "1" olarak kodlanmaktadır. Dolayısıyla olayın niyetine ve türüne dolayısıyla olayı gerçekleştiren kişi veya örgüte yönelik niteleyici bilgi içerebilmektedir.
	nhostkid	110	Nümerik değişken olan bu alanda, toplam rehine veya kaçırılan mağdur sayısını kaydedilmekte olup 178161 adet olay için "boş", "bilinmiyor" veya "0" olarak kaydedildiği dikkate alındığında niteleyici özelliği bulunmamaktadır.
	nhostkidus	111	Amerkan vatandaşlarına özel nhoskid kaydını tutmaktadır. 191084 kayıt için boş, bilinmiyor veya 0.
	nhours	112	Nümerik değişken olan bu alanda, “Saldırı Tipi” “Rehin Alma (Kaçırma)”, “Rehin Alma (Barikat Olayı)” veya başarılı bir “Hijacking” ise, olayın süresi kaydedilmektedir. 190795 adet olay için "boş", "bilinmiyor" veya "0" olarak kaydedildiği dikkate alındığında niteleyici özelliği bulunmamaktadır.
	ndays	113	Nümerik değişken olan bu alanda, "nhours" alanında tutulan sürenin 24 saati aşması durumunda benzer süre kaydı gün olarak tutulmaktadır.187571 kayıt için "boş", "bilinmiyor" veya "0" olarak kaydedildiği dikkate alındığında niteleyici özelliği bulunmamaktadır.
	divert	114	Metin değişkeni olan bu alanda, “Saldırı Tipi” “Rehin Alma (Kaçırma)” veya başarılı bir “Hijacking” ise korsanların bir aracı yönlendirdiği ülkeyi veya kaçırılan kurbanların taşındığı ve tutulduğu ülkeyi ilişkin bilgiler metin olarak kaydedilmektedir.187335 adet olay için "boş" olarak kaydedildiği dikkate alındığında niteleyici bir özelliği bulunmamaktadır.

	kidhijcountry	115	Metin değişkeni olan bu alanda, “Saldırı Tipi” “Rehin Alma” veya “Kaçırma” ise bu alanda olayın çözüldüğü veya sonlandığı ülke listelenmektedir. 188155 adet olay için "boş" olarak kaydedildiği dikkate alındığında niteleyici bir özelliği bulunmamaktadır.
	ransom	116	Kategorik değişken olan bu alanda, olayın fidye talebi içerip içermediği gösterilmektedir. Fidye talep edilmesi durumunda "1" olarak kodlanmaktadır. 114534 adet olay için "boş" veya "bilinmiyor" olarak kaydedildiği ve fidye talebi içeren kayıt sayısının 1366 olduğu dikkate alındığında niteleyici özelliğinin olmadığı değerlendirilmektedir.
	ransomamt	117	Bu alanda talep edilen fidye tutarı bilgisi tutulmaktadır. "ransom" alanının niteleyici özelliği olmamasından dolayı bu alanın da niteleyici bir özelliği bulunmamaktadır.
	ransomamtus	118	Bu alanda, USD olarak talep edilen fidye tutarı tutulmakta olup, "ransom" alanının niteleyici özelliği olmamasından dolayı bu alanın da niteleyici bir özelliği bulunmamaktadır.
	ransompaid	119	Bu alanda ödenen fidye tuturı tutulmakta olup, "ransom" alanının niteleyici özelliği olmamasından dolayı bu alanın da niteleyici bir özelliği bulunmamaktadır.
	ransompaidus	120	Bu alanda USD olarak ödenen fidye tuturı tutulmakta olup, "ransom" alanının niteleyici özelliği olmamasından dolayı bu alanın da niteleyici bir özelliği bulunmamaktadır.
	ransomnote	121	Bu alanda, para haricindeki talepler vb. hususlara ilişkin açıklayıcı bilgiler tutulmakta olup, metin değişken olarak çok farklı değerler içermesi ve "ransom" alanının niteleyici özelliği olmamasından dolayı bu alanın da niteleyici bir özelliği bulunmamaktadır.
	hostkidoutcome	122	Kategorik değişken olan bu alan rehine alma veya kaçırma olaylarında kurbanlarının durumunun nasıl sonuçlandığını kaydetmektedir. 179420 adet olay için "boş" olduğu dikkate alındığında niteleyici özelliği bulunmamaktadır.
	hostkidoutcome_txt	123	Bu alan, kategorik değişken olan "hostkidoutcome" alanının metin karşılığını temsil etmektedir.
	nreleased	124	Nümerik değişken olan bu alanda, olaydan kurtulan rehinelerin sayısı kaydedilmektedir. 184000 adet olay için "boş" olduğu dikkate alındığında niteleyici özelliği bulunmamaktadır.
IX. Additional Information and Sources	addnotes	125	Metin değişkeni olan bu alan, saldırı ile ilgili ek ayrıntıları kaydetmek için kullanılan açıklayıcı bir alandır. 160507 adet olay için "boş" olarak kaydedildiği dikkate alındığında niteleyici bir özelliği bulunmamaktadır.
	scite1	126	Bu alan, belirli bir olay hakkında bilgi derlemek için kullanılan ilk kaynağı gösterir. Olaydan bağımsız olarak veri tabanına kayıt için destekleyici unsurdur. Dolayısıyla olaya ilişkin niteleyici bir bilgi içermemektedir.
	scite2	127	Bu alan, belirli bir olay hakkında bilgi derlemek için kullanılan ikinci kaynağı gösterir. Olaydan bağımsız olarak veri tabanına kayıt için destekleyici unsurdur. Dolayısıyla olaya ilişkin niteleyici bir bilgi içermemektedir.
	scite3	128	Bu alan, belirli bir olay hakkında bilgi derlemek için kullanılan üçüncü kaynağı gösterir. Olaydan bağımsız olarak veri tabanına kayıt için destekleyici

			unsurdur. Dolayısıyla olaya ilişkin niteleyici bir bilgi içermemektedir.
	dbsource	129	Veri toplama projesine veya grubuna ilişkin bilgilerin kaydedilmesi kullanılmaktadır.Olaydan bağımsız olarak veri toplama süreci ile ilgilidir. Dolayısıyla olaya ilişkin niteleyici bir bilgi içermemektedir.
	INT_LOG	130	Kategorik değişken olan bu alan, fail grubun uyruğu ile saldırının yeri arasındaki bağlantıyı, diğer bir deyişle fail grubun bir saldırı gerçekleştirmek için sınırı geçip geçmediğini göstermek için kullanılmaktadır. Saldırının uluslararası olduğu durumlarda "1" olarak kodlanmaktadır. 97037 adet olay için "bilinmiyor", 86853 adet olay içinde "0" olarak kodlanmıştır.
	INT_IDEO	131	Kategorik değişkem bu alan, fail bir grubun farklı bir uğruğun vatandaşlarını hedef alıp almadığını göstermek için kullanılmaktadır. 97169 adet olay için "bilinmiyor", 69741 adet olay içinse "0" olarak kodlanmıştır.
	INT_MISC	132	Kategorik değişken olan bu alan, saldırının yeri ile hedef (ler) / mağdur (lar)'ın uyruğunun aynı olup olmadığını göstermek için kullanılmaktadır.488 adet olay için "bilinmiyor", "169564" adet olay içinse 0 olarak kodlanmıştır.
	INT_ANY	133	Kategorik değişken olan bu alan, saldırının, herhangi bir boyutta (LOG;IDEO;MISC) uluslararası olup olmadığını göstermek için kullanılmaktadır. 88018 adet olay için "bilinmiyor", 63860 adet olay içinse "0" olarak kodlanmıştır.

ÖZGEÇMİŞ

I. Bireysel Bilgiler

Adı: *****

Soyadı: *****

Doğum yeri ve tarihi: *****

Uyruğu: *****

Medeni durumu: *****

Askerlik Durumu: *****

İletişim Adresi: *****

Tel: *****

II. Eğitim

Yüksek Lisans: *****

Lisans: *****

Lisans: *****

Lise: *****

Yabancı Dil: *****

III. Mesleki Deneyimi

*****

IV. Diğer Bilgiler

*****

*****