T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNE ÖĞRENİMİNDE VALİDASYON TEKNİKLERİ

Görkem TEMEL

YÜKSEK LİSANS TEZİ

Matematik Mühendisliği Anabilim Dalı

Matematik Mühendisliği Programı

Danışman

Prof. Dr. Ayla ŞAYLI

Mayıs, 2025

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNE ÖĞRENİMİNDE VALİDASYON TEKNİKLERİ

Görkem TEMEL tarafından hazırlanan tez çalışması **02.07.2025** tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Mühendisliği Anabilim Dalı Matematik Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Ayla ŞAYLI
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Prof. Dr. Ayla ŞAYLI, Danışman
Yıldız Teknik Üniversitesi _____

Prof. Dr. İbrahim EMİROĞLU, Üye
Yıldız Teknik Üniversitesi _____

Dr. Öğr. Üyesi Mustafa Zahid Gürbüz, Üye
Doğuş Üniversitesi _____

Danışmanım Prof. Dr. Ayla ŞAYLI sorumluluğunda tarafımca hazırlanan "MAKİNE ÖĞRENİMİNDE VALİDASYON TEKNİKLERİ" başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Görkem TEMEL

Anneme, Babama, Eylül'e

Ve Elif Can'a

TEŞEKKÜR

Lisans ve yüksek lisans eğitimim süresince bilgi ve deneyimiyle akademik gelişimime yön veren, her zaman desteğini ve rehberliğini esirgemeyen kıymetli hocam Prof. Dr. Ayla ŞAYLI'ya en içten teşekkürlerimi sunarım. Tez sürecimde göstermiş olduğu sabır, yol göstericiliği ve değerli katkıları benim için büyük bir rehber olmuştur.

Bu süreçte her koşulda yanımda olan, sabır ve fedakârlıkla beni destekleyen, inançları ve sevgileriyle daima güç veren canım aileme sonsuz teşekkür ederim.

Görkem TEMEL

İÇİNDEKİLER

SİMGE LİSTESİ

K	Alt Küme Sayısı
M	Çapraz Validasyon Sonrası Modelin Ortalama Performansı
k	İterasyon Sayısı
$M_{\text{İ}}$	İterasyondaki Model Performansı (Her İterasyondaki Performans Değeri)
X	Orijinal Gözlem Değeri
N	Örnek Sayısı (Veri Kümesindeki Toplam Örnek Sayısı)
μ	Özelliğin Ortalaması
σ	Özelliğin Standart Sapması
D	Veri Kümesi (Toplam Veri Kümesi)

KISALTMA LİSTESİ

AUC	Eğri Altındaki Alan (Area Under The Curve)
Catboost	Kategori Artırma (Category Boosting)
CV	Çapraz Validasyon (Cross Validation)
FN	Yanlış Negatifler (False Negatives)
FP	Yanlış Pozitifler (False Positives)
FPR	Yanlış Pozitif Oranı (False Positive Rate)
GB	Gradyan Artırma (Gradient Boosting)
Hold-Out	Ayrılmış Veri (Hold-Out)
K-Fold	K-Katlı (K-Fold)
LE	Etiket Kodlaması (Label Encoding)
LGBM	Hafif Gradyan Artırma Makinası (Light Gradient Boosting Machine)
LR	Lojistik Regresyon (Logistic Regression)
RF	Rastgele Orman (Random Forest)
ROC	İşletim Karakteristiği Eğrisi (Receiver Operating Characteristic Curve)
TN	Doğru Negatifler (True Negatives)
TP	Doğru Pozitifler (True Positives)
TPR	Doğru Pozitif Oranı (True Positive Rate)
TSS	Zaman Serisi Bölmesi (Time Series Split)
XGB	Aşırı Gradyan Artırma (Extreme Gradient Boosting)

ŞEKİL LİSTESİ

TABLO LİSTESİ

ÖZET

Makine Öğreniminde Validasyon Teknikleri

Görkem TEMEL

Matematik Mühendisliği Anabilim Dalı
Matematik Mühendisliği Programı
Yüksek Lisans Tezi

Danışman: Prof. Dr. Ayla ŞAYLI

Projenin araştırma konusu, sigorta veri seti üzerinde makine öğrenmesinde farklı doğrulama tekniklerinin kullanımı olan "Talep Dosyaları İçin Dava Tahmin Modelinin Validasyonu" dur. Proje kapsamında, eksper raporu sisteme düştüğü anda, rapordaki bilgiler doğrultusunda ilgili hasar tutarının sigorta şirketi tarafından reddedilmesi durumunda, hasar sahibinin dosyayı mahkemeye taşıma olasılığı tahmin edilmiştir. Bu sayede, hasar dosyaları hakkında ileriye dönük bilgi edinilmesi ve ret/kabul kararının daha bilinçli verilmesi hedeflenmiştir. Böylece hem hasar miktarı dava miktarından daha az olabilecek dosyalarda maliyetin artmasının önüne geçilecek hem de hasar reddi sonucu ortaya çıkacak olan müşteri memnuniyetsizliği ortadan kaldırılacaktır.

Hasar dosyalarından dolayı sigorta fimasına dava açılması sigorta sektöründeki en önemli sorunlardan biridir. Öncelikle, normalize edilmemiş ve anormallik verilerine dayanan dava açılma veya açılmaması işlemlerinin yapı ve özelliklerini incelemek için veri madenciliği teknikleri kullanılmıştır. Öte yandan, dava açılma veya dava açılmamasının otomatik olarak tahmin edilmesi için makine öğrenmesi

modelleri uygulanmış ve bu modellerin güvenilirliğini değerlendirmek amacıyla validasyon teknikleri çalışılmıştır. Bu doğrultuda, farklı validasyon teknikleri kullanılarak sınıflandırma algoritmalarının dava açılmasının tahminlerindeki performans değerlendirme metriklerinin analizleri yapılmıştır.

Makine öğrenmesi modellerinin veri kalıplarını öğrenerek dava açılma işlemlerini tanımlayabilmesi kadar, bu modellerin doğru ve güvenilir tahminler üretebilmesi için kullanılan validasyon tekniklerinin belirlenmesi de kritik öneme sahiptir. Hasar dosyası için dava açılması veya açılmamasının tahmin edilmesi çalışması, yalnızca sınıflandırma algoritmalarının performansını değerlendirmekle kalmayıp, aynı zamanda farklı validasyon tekniklerinin model başarısı üzerindeki etkilerini karşılaştırarak en uygun olanın belirlenmesine odaklanmaktadır. Bu süreç, mahkemeye giden hasar dosyalarının fazladan maliyetinin (dava ücreti, avukat masrafı vb.) azaltılması ve müşteri memnuniyetinin artırılması açısından büyük önem taşımaktadır. Bu yorumlamalar hasar analistleri kendilerine bir hasar dosyası geldiğinde hasarın durumunu analiz ederek ilgili hasarı reddettiğinde bu hasar dosyasının mahkemeye gitme olasılığına bakarak, hasar tutarı özelinde ilgili hasarı reddetme ya da reddetmeme durumunda karar verilmesine yardımcı olacaktır.

Anahtar Kelimeler: Validasyon teknikleri, sınıflandırma algoritmaları, makine öğrenmesi, dava modelleri

ABSTRACT

Validation Techniques in Machine Learning

Görkem TEMEL

Department of Mathematical Engineering

Master Thesis

Supervisor: Prof. Dr. Ayla ŞAYLI

The research area of the project is different validation techniques in machine learning on the insurance dataset, "Litigation Model's Validation for Claim Files". Within the scope of this project, in the case of the insurance company rejection of the insured claim's amount based on the expert report, the litigation models of insured are calculated. The main aim is to have forward-looking information about the claims files before the requested amount rejection to make more informed decisions if the requested amount is not above the additional costs associated with lawsuits (such as court and attorney fees) faced by the company and also for customer dissatisfaction that may arise because of claim rejections.

Whether a damage file results in a lawsuit is one of the most critical problems in the insurance sector. To begin with, data mining techniques are used to examine the structure and characteristics of both sued and non-sued cases, based on data normalization and anomalies. Furthermore, machine learning models have been applied to automatically predict whether a claim will be litigated or not, and validation techniques are used to evaluate the reliability of these models. In this

respect, the performance metrics of classification algorithms in predicting lawsuits were analyzed using different validation techniques.

In addition to machine learning models being able to identify suing tendencies by learning from data patterns, it is also crucial to determine the appropriate validation techniques to ensure accurate and reliable predictions. The study focuses not only on evaluating the performance metrics of classification models in predicting lawsuits, but also on comparing the effects of different validation methods on model success. This process is essential in terms of reducing the additional costs associated with lawsuits (such as court and attorney fees) and maintaining customer satisfaction. When damage analysts receive a claims file, they analyze the file's status; in cases of rejection, they can have a preliminary idea about the likelihood of the claimant suing the company. This insight will assist in making a better-informed decision on whether to reject the claim, particularly when the compensation amount is significant.

Keywords: Validation techniques, classification algorithms, data pre-processing, machine learning, litigation models

1
GİRİŞ

1.1 Tezin Amacı

Hasar dosyalarında dava açılma tahmin modelinde, eksper raporu geldiği anda, hasar dosyasındaki özellikler kullanılarak, ilgili hasar dosyasının mahkemeye taşınıp taşınmayacağının tahmin edilmesi bir sınıflandırma problemidir. Bu çalışmada, sigorta şirketinde görevli olan analistin, sigortalının hukuki sürece başvurma ihtimalini önceden tahmin edebilmesi sağlanarak karar verme süreçlerine destek olunması hedeflenmektedir.

Ancak, tahmin modelinin güvenilirliğini ve doğruluğunu artırmak için yalnızca makine öğrenmesi algoritmalarının kullanılması yeterli değildir; aynı zamanda modelin validasyon teknikleriyle test edilmesi gerekmektedir. Bu nedenle, proje kapsamında geliştirilen tahmin modelinin başarısını ölçmek için farklı validasyon teknikleri çalışılmıştır. Bu aşamada ilgili hasar dosyasında dava açılacağının sigorta firmasında çalışan görevli tarafından önceden tahmin edilebilir olması, dosyayı reddetme ya da kabul etme aşamasında dosyaya bakan çalışanın karar vermesine yardımcı olacaktır. Bu sayede daha doğru kararlar vererek süreci hem şirket hem de müşteri lehine yönetmesine olanak sağlanacaktır.

Çalışmanın yapılacağı veri kümesinde, gerekli hazırlıklar ve analizler yapıldıktan sonra (veriyi tanıma, temizleme, özellik seçimi vs.), elde edilen hedef özellik (dava açılma/açılmama durumu, binary classification) özelinde keşifsel veri analizi, özellik mühendisliği (feature engineering), model oluşturma, model parametre ayarlama (parameters tuning) gibi işlemler gerçekleştirilecektir.

Proje kapsamında, tahmin modelini geliştirmek için kullanılacak olan veri kümesinde bulunacak olan özellikler hasar toplam tutarı, ön rapor, ihbar başvuru şekli, cam modül mü, as skoru renk, tam hasar mı, faturalı iş mi, hasar kalemi açılış tarihi, rücu durumu, atama grubu, ihbarı alan kurum, hasar sebebi, servis türü, servis tipi, hasar tarihi, ihbar tarihi, hasar şekli, anlaşmalı çam servis, üretim müdürlüğü, acente il adı, acente bölge adı, müşteri tipi, hasarsızlık kademesi, araç kullanım şekli, marka, model yılı, kusur oranı, onarım işlemi yapıldı mı, tamir tutarı var mı, boya tutarı var mı, araç bedeli ve kullanım şeklidir. Hasar dava açma ilişkisi özelliği de tahmin edilecek olan bağımlı özelliği yani hedefi ifade etmektedir.

Geliştirilecek olan proje kapsamında, dava açmaya giden hasar dosyası tahmin edildiği üzere çok az olmaktadır. Buda dengesiz bir veri seti oluşturmakta ve dengesiz veri setine göre sınıflama algoritması kullanılmasını gerektirmektedir.

1.2 Hipotez

Bu çalışmada, farklı validasyon teknikleri kullanılarak oluşturulan sınıflandırma algoritmalarının, modelin doğruluk ve genelleme performansı üzerinde belirgin etkiler yarattığı hipotez edilmektedir. Özellikle, dengesiz sınıf dağılımına sahip trafik sigortası hasar verilerinde, doğru validasyon teknik seçiminin, sigortalının dava açma olasılığını tahmin etme başarısını artıracağı düşünülmektedir. Bu kapsamda, farklı validasyon teknikleri karşılaştırılarak, hangi tekniğinin bu tür veri setlerinde daha güvenilir sonuçlar verdiği ortaya konulacaktır. Ayrıca, veri setine ilişkin çeşitli özellik mühendisliği (feature engineering) ve analizlerle desteklenen deneysel çalışmalar ile dava açma skoru tahmininin doğruluğunu artırmaya yönelik katkılar sunulacaktır.

1.3 Tezin Konusu

Trafikte seyir halinde bulunan sigortalının herhangi bir nedenden dolayı kazaya karışması durumunda, ilgili hasar verilerinden yola çıkılarak, hasar ödemesi reddedildiğinde sigortalının sigorta şirketine dava açma olasılığı değerlendirilmektedir. Bu proje kapsamında, trafik ve kasko branşındaki hasar dosyaları için sigortalının şirketi dava açma ihtimali makine öğrenmesi algoritmaları kullanılarak tahmin edilecektir.

Ancak, makine öğrenmesi modellerinin tahmin başarısının güvenilir bir şekilde ölçülmesi için validasyon teknikleri kritik bir rol oynamaktadır. Bu nedenle, çalışma kapsamında farklı validasyon teknikleri çalışılarak, oluşturulan tahmin modelinin doğruluk, genelleme yeteneği ve hata oranları değerlendirilecektir. Makine öğrenmesi algoritmaları ile hesaplanan "dava açılma veya açılmama" hedefi, hasar dosyasını inceleyen analist için bir karar noktası oluştururken, kullanılan validasyon teknikleri sayesinde modelin ne kadar güvenilir olduğu test edilerek karar süreçlerinin daha sağlam temellere oturtulması sağlanacaktır.

1.4 Literatür Özeti

Makine öğrenmesi, karar destek sistemleri oluşturmak ve veri odaklı öngörülerde bulunmak amacıyla günümüzde birçok alanda yaygın olarak kullanılmaktadır. Özellikle sigorta, finans, sağlık ve hukuk gibi alanlarda büyük veri kümelerinden anlamlı bilgiler çıkarılması, bu teknolojilerin sunduğu en önemli avantajlardan biridir. Trafik sigortası gibi geniş hacimli ve karmaşık verilerin işlendiği alanlarda, makine öğrenmesi algoritmaları, veri üzerinden hukukî süreçlerin öngörülmesi gibi görevlerde etkili sonuçlar verebilmektedir.

Bu süreçte kullanılan makine öğrenmesi modellerinin başarısını belirleyen en önemli faktörlerden biri ise modelin doğruluğunun ve genelleme yeteneğinin güvenilir bir şekilde ölçülmesidir. Bu noktada validasyon (doğrulama) teknikleri devreye girmekte, modelin hem eğitim hem de gerçek dünya verisi üzerindeki performansını değerlendirmek için kritik bir rol üstlenmektedir.

Validasyon teknikleri, modelin aşırı öğrenme (overfitting) ya da yetersiz öğrenme (underfitting) gibi problemlerden etkilenip etkilenmediğini anlamada ve modelin farklı veri bölümleri üzerindeki tutarlılığını test etmede kullanılır. Doğru validasyon tekniği seçilmediğinde, modelin güvenilirliği hakkında elde edilen sonuçlar yanıltıcı olabilir. Bu nedenle literatürde farklı validasyon teknikleri geliştirilmiş ve çeşitli veri setleri üzerinde test edilmiştir.

K-Katlı Çapraz Validasyon (K-Fold Cross Validation), Gruplu K-Katlı Çapraz Validasyon, Ayrılmış Veri Validasyon (Hold-Out Validation) ve Katmanlı K-Katlı Çapraz Validasyon (Stratified K-Fold Cross Validation) gibi teknikler, farklı veri

yapıları ve problemleri için literatürde yaygın olarak kullanılan validasyon teknikleri arasında yer almaktadır. Bu tekniklerden her biri, örneklem büyüklüğü, veri dengesi, hesaplama maliyeti ve modelin genel performansı üzerindeki etkileri bakımından detaylıca incelenmiştir.

Makine öğrenmesinde validasyon tekniklerine dair literatür çalışmaları, bu tekniklerin teorik arka planını ve uygulamalardaki pratik katkılarını ortaya koymakta, aynı zamanda model güvenilirliğini sağlamak adına bu tekniklerin neden kritik olduğunu vurgulamaktadır. Bu tez kapsamında, trafik sigortası hasar verisi üzerinden sigortalının dava açma yoluna başvurma ihtimalinin tahmininde kullanılan modellerin başarısı, çeşitli validasyon teknikleriyle değerlendirilerek, elde edilen sonuçların ne kadar güvenilir olduğu analiz edilecektir.

Ayrıca bu tez kapsamındaki trafik sigortası hasar verisi üzerinden sigortalının dava açma yoluna başvurma ihtimalinin tahmininde kullanılan modellerin başarısı, çeşitli validasyon teknikleriyle değerlendirilerek, elde edilen sonuçların ne kadar güvenilir olduğuna dair yayınlar bulunmaktadır.

Yukarıda belirtilen hususlarla ilgili literatür çalışmaları alt bölümlerde açıklanmıştır.

1.4.1 Validasyon Teknikleri Alanındaki Literatür Çalışmaları

Makine öğrenmesi modellerinin güvenilirliğini ve genellenebilirliğini değerlendirmek için kullanılan validasyon teknikleri, modelleme sürecinin en kritik adımlarından biridir. Bu teknikler, özellikle eğitim verisi üzerinde yüksek başarı sağlayan bir modelin yeni ve görülmemiş veriler üzerindeki performansını ölçmek amacıyla geliştirilmiştir.

Literatürde en yaygın kullanılan validasyon teknikleri farklı veri büyüklüklerine ve problem yapılarına göre avantajlar ve dezavantajlara sahiptir. Örneğin, Kohavi (1995) tarafından yapılan kapsamlı bir karşılaştırmalı çalışmada, 10-Katlı Çapraz Validasyonunun hem düşük varyans hem de düşük sapma ile en dengeli performansı sağladığı gösterilmiştir. Bu teknik, özellikle orta büyüklükteki veri setlerinde model başarısını değerlendirmek için tercih edilmektedir [1].

Büyük veri setlerinde ise hesaplama maliyetini azaltmak adına Ayrılmış Veri Validasyon (Hold-Out Validation) veya Tekrarlı Katmanlı K-Katlı Çapraz Validasyon (Repeated Stratified K-Fold Cross Validation) gibi teknikler tercih edilebilmektedir [2]. Özellikle sınıf dengesizliğinin söz konusu olduğu veri kümelerinde, Japkowicz ve Stephen (2002) tarafından önerilen Katmanlı K-Katlı Çapraz Validasyon (Stratified K-Fold Cross Validation), hem azınlık hem de çoğunluk sınıfın orantılı olarak her katmanda temsil edilmesini sağlayarak daha güvenilir bir değerlendirme ortamı sunmaktadır [3].

Makine öğrenimi modellerinde yoğun olarak kullanılan başlıca validasyon teknikleri aşağıdaki tabloda özetlenmiştir. Validasyon tekniklerinin tarihsel gelişimi ve kökenleri Tablo 1.1'de özetlenmiştir.

Tablo 1.1 Validasyon Tekniklerinin Tarihsel Gelişimi ve Kökenleri

Teknik Adı	Yıl	Gelişme(ler)
Ayrılmış Veri Validasyon (Hold-Out Validation)	**1951**	John Tukey'in istatistiksel yöntemlerinden türeyen kavramlar [4]
K-Katlı Çapraz Validasyon (K-Fold Cross Validation)	**1974**	Seymour Geisser'in "predictive sample reuse" teorisi [5]
Katmanlı K-Katlı Çapraz Validasyon (Stratified K-Fold Cross Validation)	**1990'lar**	Makine öğrenmesi topluluğunda standartlaşmıştır [1]
Gruplu K-Katlı Çapraz Validasyon (Group K-Fold Cross Validation)	**2000'ler**	Kaggle gibi veri bilimi yarışmalarında ortaya çıkmıştır [6]
Tekrarlı K-Katlı Çapraz Validasyon (Repeated K-Fold Cross Validation)	**2001**	Ron Kohavi'nin önerdiği validasyon tekniğinden türemiştir [1]
Tekrarlı Katmanlı K-Katlı Çapraz Validasyon (Repeated Stratified K-Fold Cross Validation)	**2005**	Scikit-learn ve benzeri kütüphanelerin katkısıyla yaygınlaşmıştır [7]
Katmanlı Gruplu K-Katlı Çapraz Validasyon (Stratified Group K-Fold Cross Validation)	**2015**	Kaggle Medical yarışmaları ve sağlık verisi çalışmaları sayesinde yaygınlaştı [8]
Zaman Serisi Bölmesi-İleri Zincirleme Çapraz Validasyon (Time Series Split -Forward Chaining Cross Validation)	**2012**	Scikit-learn geliştirici topluluğu (ör. Andreas Müller) [7]

Son yıllarda, validasyon stratejilerinin sadece bir ölçüm aracı olmaktan öte, modelin yapılandırılmasında ve seçilmesinde aktif rol oynadığı anlaşılmıştır. Bu

nedenle, geliştirilecek herhangi bir makine öğrenmesi modelinin güvenilirliğini sağlamak için, doğru validasyon tekniğinin seçimi büyük önem taşımaktadır.

1.4.2 Trafik Sigortası ve Dava Açma Tahmini Üzerine Literatür Çalışmaları

Sigorta sektöründe, hasar dosyalarının analiz edilerek potansiyel dava açma süreçlerinin öngörülmesi, risk yönetimi ve operasyonel verimlilik açısından büyük önem taşımaktadır. Bu alanda yapılan çalışmalar, sahte hasar tespiti ve dava açma olasılığının tahmini gibi konuları kapsamaktadır.

Şahin, Ayvaz ve Çalımfidan (2020) tarafından yapılan bir çalışmada, özel bir sigorta şirketinin kasko sigortasına ait hasar verileri kullanılarak sahte hasarların tespiti için çeşitli makine öğrenmesi modelleri karşılaştırılmıştır. Çalışmada, K-En Yakın Komşuluk, Karar Ağaçları, Lojistik Regresyon ve Yapay Sinir Ağları gibi sınıflama algoritmaları denenmiş ve makine öğrenmesinin sahte hasar tespiti konusunda etkili olduğu sonucuna varılmıştır [9].

Bu tür çalışmalar, sigorta şirketlerinin hasar dosyalarını daha etkin bir şekilde analiz etmelerine ve potansiyel dava süreçlerini önceden tahmin etmelerine olanak tanımaktadır. Ancak, dava açma tahmini konusunda daha fazla çalışmaya ihtiyaç duyulmaktadır.

1.5 Organizasyon Yapısı

Tezin organizasyon yapısı altı bölümden oluşmaktadır.

1. Bölümde çalışmanın amacı, hipotezleri ve konuya ilişkin literatür özeti sunulmaktadır.

2. Bölümde makine öğrenmesinde kullanılan validasyon (doğrulama) teknikleri detaylı biçimde ele alınmakta, farklı tekniklerin çalışma prensibi, avantajları ve dezavantajları açıklanmaktadır.

3. Bölümde dava açma tahmini için kullanılan sınıflandırma algoritmalarına yer verilmekte ve bu algoritmaların temel prensipleri açıklanmaktadır.
4. Bölümde trafik sigortası hasar verisi üzerinde uygulanan veri ön işleme, özellik mühendisliği ve dengesiz veri sorununa karşı alınan önlemler gibi veri hazırlama adımları ayrıntılı biçimde sunulmaktadır.

5. Bölümde, uygulama süreci detaylı olarak ele alınmakta; kullanılan yazılım kütüphaneleri, uygulanan makine öğrenmesi algoritmaları ve validasyon teknikleri karşılaştırmalı biçimde sunulmakta, elde edilen sonuçlara yer verilmektedir.

Son olarak, 6. Bölümde yapılan çalışmanın genel bir değerlendirmesi yapılmakta, elde edilen sonuçlar özetlenmekte ve gelecek çalışmalara yönelik öneriler sunulmaktadır.

2
VALİDASYON TEKNİKLERİ

Makine öğrenimi modellerinin kabiliyetini değerlendirmek, modelleme sürecinin temel bileşenlerinden biridir. Bu süreçte kullanılan validasyon teknikleri, modelin yalnızca eğitim verisinde başarılı olup olmadığını değil, aynı zamanda görülmemiş veriler üzerinde ne derece etkili olduğunu belirlemek için geliştirilmiştir. Modelin aşırı öğrenme (overfitting) veya yetersiz öğrenme (underfitting) gibi problemlere karşı duyarlılığını anlamak, model seçimi ve hiperparametre optimizasyonu gibi aşamalarda kritik bir rol oynamaktadır. [10]

Validasyon tekniklerinin her biri alt bölümlerde detaylı olarak açıklanmıştır.

2.1 Ayrılmış Veri Validasyon (Hold-Out Validation)

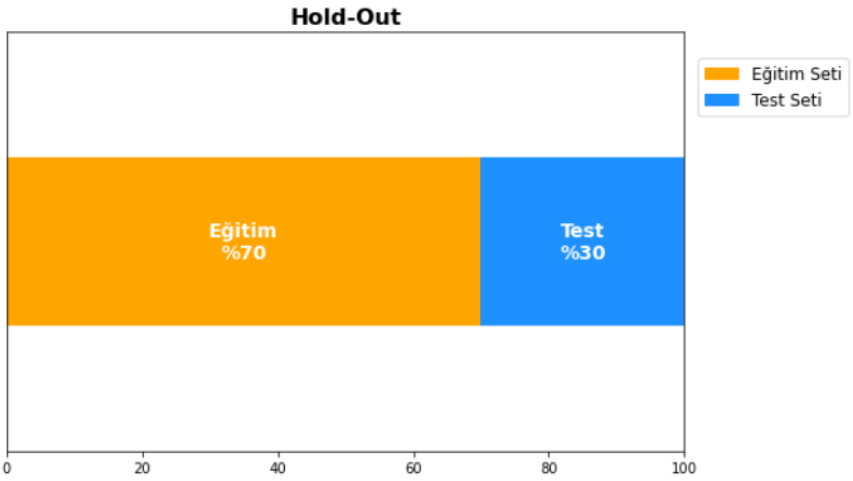
Ayrılmış Veri Validasyon (Hold-Out Validation), veri kümesinin rastgele iki alt kümeye bölünmesiyle uygulanır. Eğitim (training) kümesi modelin öğrenmesini sağlarken, test (test) kümesi modelin performansını değerlendirmek için kullanılır. Genellikle veri setinin %70-80'lik bölümü eğitim için, %20-30'luk bölümü ise test için ayrılır.[11]

Bu tekniğin temel avantajı, hesaplama açısından düşük maliyetli olmasıdır. Ancak, veri setinin nasıl bölündüğüne bağlı olarak modelin performansı değişebilir. Özellikle küçük veri kümelerinde, bazı önemli örnekler eğitim setinde bulunmazsa model eksik öğrenebilir.

Genellikle veri bölme oranları şu şekilde belirlenir:

- **p% eğitim için kullanılır** (genellikle p=0.7 veya p=0.8),
- **(1−p)% test için kullanılır** (genellikle 0.20. veya 0.30).

Veri setinin bu oranlara göre ikiye ayrılması, makine öğrenmesi modellerinin eğitimi ve değerlendirilmesinde en temel tekniklerden biridir. Bu strateji literatürde Ayrılmış Veri Validasyon (Hold-Out Validation) olarak adlandırılır. Şekil 2.1'de Ayrılmış Veri Validasyon (Hold-Out Validation) kapsamında eğitim ve test veri setlerinin bölünme yapısı görsel olarak sunulmuştur.



Şekil 2.1 Ayrılmış Veri Validasyonu (Hold-Out Validation) ile Veri Setinin Eğitim ve Test Olarak Bölünmesi

Ayrılmış Veri Validasyon (Hold-Out Validation), çeşitli avantajlara sahip olup hesaplama maliyetinin düşük olması, büyük veri setleri ile ölçeklenebilir olması ve kolay uygulanabilirliği ile öne çıkmaktadır. Bu teknikte veri seti tek seferde eğitim ve test kümelerine ayrıldığından, modelin eğitimi ve değerlendirilmesi hızlı bir şekilde gerçekleştirilebilir. Ayrıca, büyük veri setlerinde ekstra yinelemelere gerek duymadan etkili bir şekilde çalışması, bu tekniğin önemli bir avantajıdır. Diğer validasyon tekniklerine kıyasla daha az kod ve hesaplama gerektirmesi ise uygulama kolaylığını artırmaktadır.

Bununla birlikte, Ayrılmış Veri Validasyon (Hold-Out Validation) bazı dezavantajları da bulunmaktadır. Veri setinin rastgele bölünmesi nedeniyle, özellikle küçük veri setlerinde bazı kritik örneklerin yalnızca test kümesine düşme olasılığı vardır. Bu durum, modelin belirli veri noktalarını öğrenememesine ve genelleme kabiliyetinin azalmasına neden olabilir. Ayrıca, farklı bölme işlemlerinin farklı sonuçlar üretmesi, modelin performansının tekrarlanabilirliğini olumsuz yönde etkileyebilir. Son olarak, test kümesinin yeterince büyük olmaması

durumunda, modelin gerçek dünyadaki performansını doğru bir şekilde ölçmek güçleşebilir ve modelin genelleme yeteneği hakkında yanıltıcı sonuçlar elde edilebilir.

2.1.1 K-Katlı Çapraz Validasyon (K-Fold Cross Validation)

K-Katlı Çapraz Validasyon (K-Fold Cross Validation / K-Fold CV), istatistiksel modelleme ve makine öğrenimi algoritmalarının doğruluğunu değerlendiren önemli bir tekniktir. Ayrılmış Veri Validasyon (Hold-Out Validation) tekniğinin eksikliklerini gidermek amacıyla geliştirilmiş olup, modelin farklı veri alt kümeleri üzerinde test edilmesini sağlayarak aşırı uyumu (overfitting) önlemeyi amaçlar. K-Fold 'un temel mantığı, veri setinin K adet alt kümeye bölünmesi ve her bir alt kümenin sırasıyla test seti olarak kullanılmasıdır. Bu süreç, modelin genellenebilirliğini daha sağlam bir şekilde değerlendirmenin bir yoludur. K-Fold, veri setinin K adet eşit parçaya bölünmesiyle başlar. Her bir parça, sırasıyla test seti olarak seçilirken, geriye kalan K-1 parça ise eğitim seti olarak kullanılır. Bu süreç, toplam K kez tekrarlanır. Sonuç olarak, her veri noktası bir kez test setinde yer alır ve K defa eğitim setine dahil olur. Bu teknik, modelin her veri noktasına karşı duyarlılığını artırır ve genelleme yeteneğini optimize eder.[12]

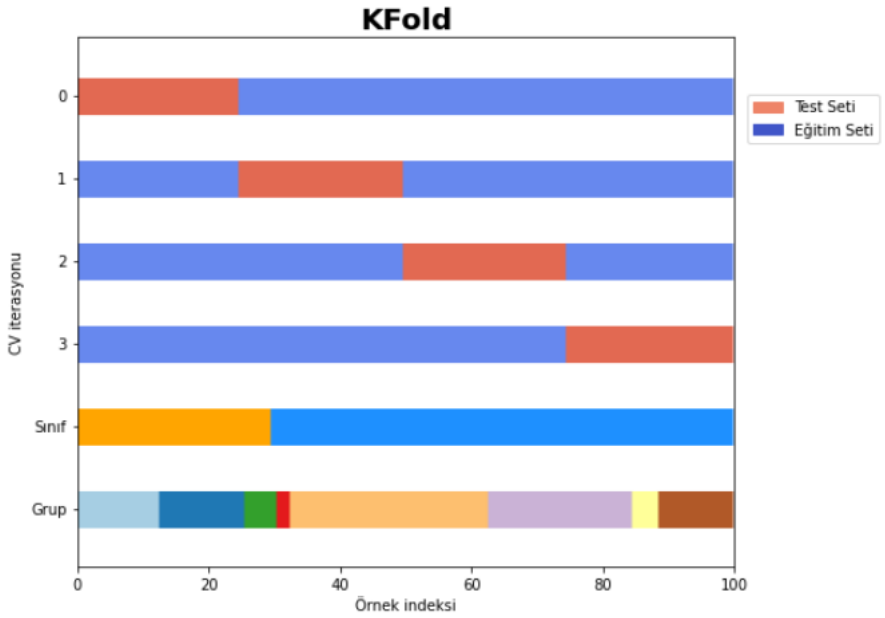
2.1.2 Uygulama Süreci

K-Fold Çapraz Validasyon tekniğinin adımları şu şekilde özetlenebilir:

- **Veri Setinin Bölünmesi:** Veri seti, K eşit parçaya bölünür. Bu bölme işlemi, veri setinin homojenliği göz önünde bulundurularak yapılmalıdır. Eğer veri setinde hedef özelliğinin sınıflarında dengesizlik varsa, Katmanlı K-Fold CV tercih edilebilir Katmanlı K-Fold CV, her katmanda sınıf dağılımını koruyarak daha sağlıklı sonuçlar elde edilmesini sağlar.[13]
- **Eğitim ve Test Aşamaları**: Her bir katmanda, bir alt küme test seti olarak seçilirken, geri kalan K-1 alt küme eğitim seti olarak kullanılır. Model, eğitim verisiyle eğitildikten sonra, test verisi üzerinde performansı değerlendirilir.
- **Sonuçların Ortalaması**: Modelin her bir katmandaki doğruluğu kaydedildikten sonra, bu doğrulukların ortalaması alınarak modelin genel

performansı hesaplanır. Bu işlem, modelin aşırı uyum yapıp yapmadığını anlamaya yardımcı olur.

K-Fold CV tekniği, veri kümesini K parçaya bölerek her parçayı bir kez test verisi, kalanlarını eğitim verisi olarak kullanır. Şekil 2.2'de bu tekniğin temel işleyişi basit bir şekilde şematik olarak görselleştirilmiştir.



Şekil 2.2 K-Fold Çapraz Validasyon Tekniğinin Şematik Gösterimi

Veri kümesi D, N örnek içeren bir veri seti olarak tanımlansın:

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\} \tag{2.1}$$

Bu veri kümesi K eşit parçaya bölünerek D_1,D_2,…,D_K alt kümeleri oluşturulur:

$$D = D_1 \cup D_2 \cup \cdots \cup D_K, \quad D_i \cap D_j = \emptyset \quad (\text{eğer } i \neq j) \tag{2.2}$$

Her iterasyonda eğitim ve test setleri şu şekilde tanımlanır:

$$D_{\text{train}}^{(k)} = D \setminus D_k \tag{2.3}$$

$$D_{\text{test}}^{(k)} = D_k \tag{2.4}$$

Burada k, K iterasyonundan herhangi birini ifade eder. Model, her iterasyonda eğitim verileriyle fonksiyonunu öğrenir ve test verileri üzerinde değerlendirilir. Çapraz Validasyon süreci sonunda, modelin ortalama performansı aşağıdaki gibi hesaplanır:

$$M = \frac{1}{K}\sum_{k=1}^{K} M_k \quad (2.5)$$

2.1.3 Avantajları ve Dezavantajları

K-Fold CV, birkaç önemli avantaj sunar. İlk olarak, bu teknik her veri noktasının test setinde yer almasına olanak tanır. Bu sayede modelin genelleme yeteneği daha iyi değerlendirilir.[14] Ayrıca, tüm verilerin eğitim ve test aşamalarında kullanılması, modelin veriye duyarlılığını artırarak aşırı uyum (overfitting) riskini azaltır. K-Fold CV, modelin doğruluğu hakkında daha sağlam bir tahmin sunar. Özellikle küçük veri setleriyle çalışıldığında, modelin eğitim verilerine olan bağımlılığı minimalize edilerek daha güvenilir sonuçlar elde edilebilir.

Ancak, K-Fold CV 'nun bazı dezavantajları da vardır. En önemli dezavantajı, hesaplama maliyetinin yüksek olmasıdır. Çünkü modelin eğitim süreci her bir katman için tekrarlanmalıdır. Bu durum, büyük veri setleri ve karmaşık modellerde oldukça zaman alıcı olabilir. Ayrıca, K-Fold CV 'nun verinin çok homojen olduğu durumlarda daha az bilgi sağlama eğiliminde olduğu gözlemlenmiştir. Bu nedenle, modelin farklı veri alt kümelerindeki performansını değerlendirmek önemlidir.

2.2 Katmanlı K-Katlı Çapraz Validasyon (Stratified K-Fold Cross Validation)

Makine öğrenimi modellerinin performansını değerlendirmek için kullanılan çapraz Validasyon, modelin genelleme yeteneğini artırarak veri kümesine aşırı uyum (overfitting) problemini önlemeye yardımcı olmaktadır. Katmanlı K-Katlı CV (Stratified K-Fold CV), veri setini K adet eşit büyüklükte alt kümeye böler ve her adımda birini test kümesi olarak kullanırken kalan K-1 kümesini eğitim için ayırır. Ancak, özellikle dengesiz sınıf dağılımına sahip veri kümelerinde, K-Fold bazı alt kümelerde nadir görülen sınıfların tamamen dışlanmasına veya orantısız dağılmasına neden olabilir.[15]

Bu problemi aşmak için kullanılan Stratified K-Fold CV, her bir alt kümede sınıf oranlarını mümkün olduğunca orijinal veri kümesindeki oranlarla aynı tutarak katmanlı (stratified) bir bölme işlemi gerçekleştirir. Böylece, her eğitim ve test

kümesi, veri setinin genel sınıf dağılımını daha iyi yansıtır ve modelin her sınıf için dengeli bir şekilde değerlendirilmesine olanak tanır.

2.2.1 Uygulama Süreci

Stratified K-Fold CV tekniği aşağıdaki adımlarla uygulanır:

1. **Veri Kümesi Ayrıştırması:** Veri kümesi, belirlenen K katmana bölünür.
2. **Sınıf Oranlarının Korunması:** Her katman, orijinal veri kümesindeki sınıf oranlarına uygun olarak oluşturulur.
3. **K Adet Model Eğitimi ve Değerlendirilmesi:**
    - K iterasyon boyunca, her defasında bir kat test kümesi olarak ayrılırken kalan K-1 kat modelin eğitimi için kullanılır.
    - Model her iterasyonda test verisi üzerinde değerlendirilerek metrik sonuçları kaydedilir.
4. **Sonuçların Birleştirilmesi:** Her iterasyondaki değerlendirme metriklerin hesaplanarak ortalaması modelin genel performansı belirlenir.

2.2.2 Avantajları ve Dezavantajları

Stratified K-Fold CV tekniğinin avantajları şunlardır:

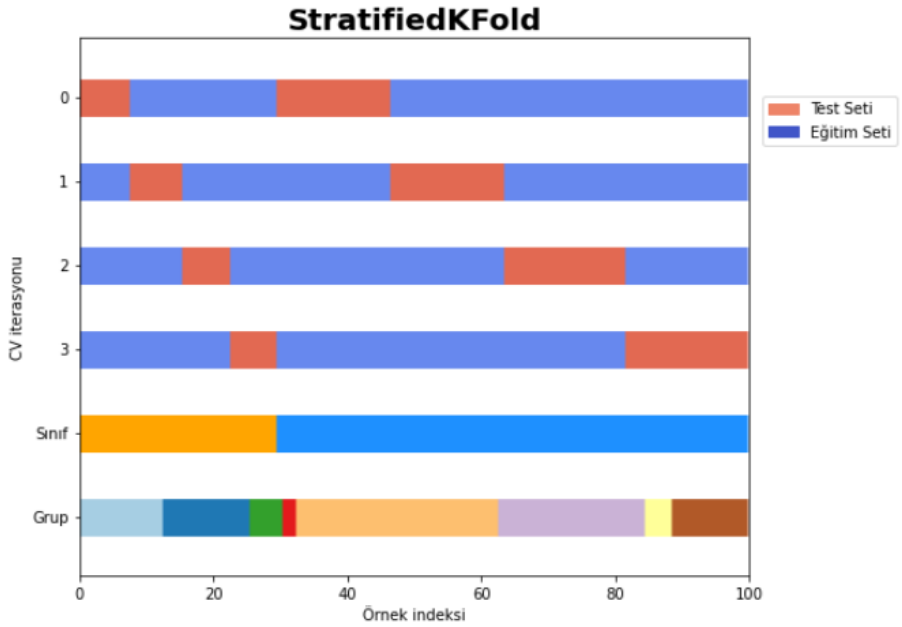
- **Sınıf Dengesizliği Problemini Azaltır**: Veri setinde nadir görülen sınıfların tüm katmanlarda temsil edilmesini sağlayarak modelin daha tutarlı bir şekilde değerlendirilmesine yardımcı olur.[16]
- **Genelleme Performansını Artırır**: Standart K-Fold'a kıyasla, modelin gerçek dünya verileri üzerindeki performansını daha iyi yansıtır.
- **Hesaplama Maliyeti Açısından Etkinlik**: Veri seti boyutu büyük olmadığı sürece, hesaplama maliyeti açısından K-Fold ile benzer seviyededir.

Ancak, Stratified K-Fold validasyon tekniğinin bazı dezavantajları da bulunmaktadır:

- **Hesaplama Maliyeti**: Özellikle büyük veri kümelerinde, K kez model eğitimi gerektirdiği için hesaplama süresi artabilir.

- **Küçük Veri Setlerinde Hassasiyet**: Veri seti küçükse, her katmanda sınıf dağılımını tam olarak korumak zor olabilir.

Şekil 2.3'te bu tekniğin temel işleyişi basit bir şekilde görselleştirilmiştir.



Şekil 2.3 Katmanlı K-Katlı Çapraz Validasyon Tekniğinin Şematik Gösterimi

2.2.3 Uygulama Alanları

Stratified K-Fold özellikle aşağıdaki alanlarda yaygın olarak kullanılmaktadır:

- **Tıbbi Teşhis Sistemleri**: Dengesiz veri dağılımının yaygın olduğu tıbbi teşhis verilerinde, hastalık sınıflandırmalarında daha adil bir model değerlendirmesi sağlar.[17]
- **Doğal Dil İşleme (NLP)**: Duygu analizi, konu sınıflandırma ve spam tespiti gibi dengesiz veri kümelerinin sıkça karşılaşıldığı uygulamalarda kullanılır.
- **Siber Güvenlik**: Zararlı yazılım tespiti veya anomali algılama sistemlerinde modelin düşük sıklıkla görülen tehditleri daha iyi öğrenmesini sağlar.

Stratified K-Fold çapraz validasyon, özellikle dengesiz veri kümeleri ile çalışırken modelin daha dengeli bir şekilde değerlendirilmesini sağlayan etkili bir tekniktir. Veri kümesindeki sınıf oranlarını koruyarak, modelin her sınıf için adil bir şekilde test edilmesini garanti eder. Bu nedenle, dengesiz veri kümeleri içeren makine

öğrenimi problemlerinde standart K-Fold tekniğine kıyasla daha güvenilir sonuçlar elde edilmesini sağlar.

2.3 Gruplu K-Katlı Çapraz Validasyon (Group K-Fold Cross Validation)

Makine öğrenimi modellerini değerlendirirken kullanılan bir diğer çapraz validasyon tekniği de Group K-Fold'dur. Standart K-Fold tekniği, veri setini K adet alt kümeye bölerken, veri içerisindeki örneklerin bağımsız olduğu varsayımını yapar. Ancak bazı veri setlerinde, örnekler belirli gruplar (örneğin aynı hastaya ait ölçümler, aynı kullanıcıdan gelen veriler) arasında ilişkilidir. Bu tür bağımlılıkların göz ardı edilmesi, modelin değerlendirilmesinde yanıltıcı sonuçlara yol açabilir.[17]

Group K-Fold Çapraz Validasyon, bu bağımlılıkları dikkate alarak, aynı gruba ait verilerin hem eğitim hem de test setinde bulunmamasını sağlar. Böylece model, daha gerçekçi bir şekilde yeni, görülmemiş gruplar üzerinde test edilmiş olur.

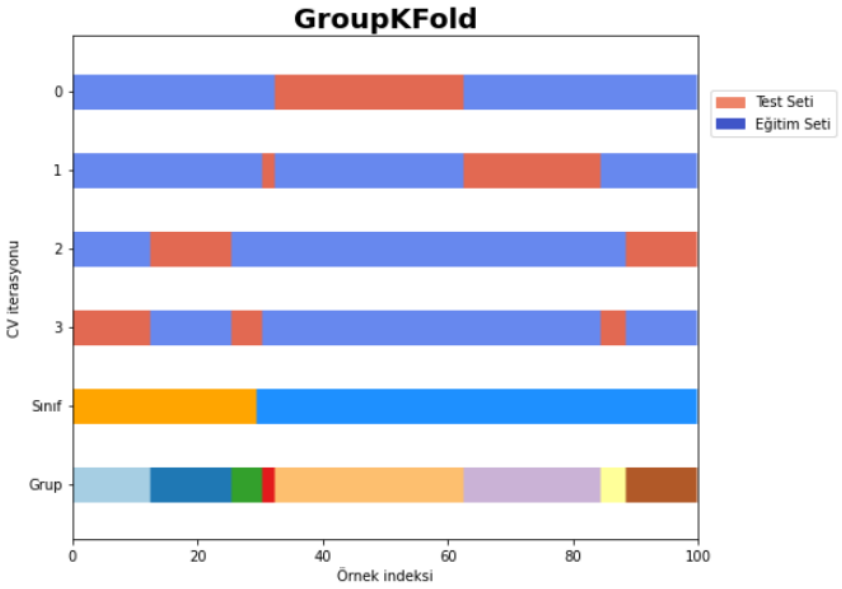
2.3.1 Uygulama Süreci

Group K-Fold çapraz validasyon tekniği, özellikle gruplar arası bağımlılıkların söz konusu olduğu veri kümelerinde, modelin performansını daha gerçekçi bir şekilde değerlendirmek amacıyla geliştirilmiştir.

Bu tekniğin uygulanmasında ilk adım, veri kümesindeki her bir örneğin ait olduğu grup etiketiyle ilişkilendirilmesidir. Böylece, aynı gruba ait verilerin birlikte değerlendirilmesi ve eğitim-test ayrımı sırasında birbirine karışmaması sağlanır. İkinci aşamada, tüm gruplar, önceden belirlenen K sayısına uygun şekilde, eşit ya da yaklaşık eşit büyüklükte katmanlara bölünür. Her iterasyonda, bu gruplardan biri test seti olarak ayrılırken, geri kalan K-1 grup eğitim setini oluşturur. Eğitim aşamasında model, yalnızca eğitim gruplarına ait verilerle eğitilir ve ardından test grubuna ait veriler üzerinde doğrulama yapılır. Bu süreç K kez tekrarlanarak her grup bir kez test verisi olarak kullanılır.

Son aşamada ise, her iterasyon sonucunda elde edilen metrikler (örneğin Doğruluk, F1-Skoru gibi) birleştirilir ve ortalamaları alınarak modelin genel performansı hesaplanır. Böylelikle Group K-Fold tekniği, modelin yeni, daha önce görülmemiş

gruplar üzerindeki başarısını daha tutarlı bir şekilde ölçmeyi mümkün kılar.[18] Şekil 2.4'te bu tekniğin temel işleyişi basit bir şekilde şematik olarak görselleştirilmiştir.



Şekil 2.4 Gruplu K-Katlı Çapraz Validasyon Tekniğinin Şematik Gösterimi

2.3.2 Avantajları ve Dezavantajları

Group K-Fold CV'nun avantajları şunlardır:

- **Bağımlı Verilerin Karışmasını Önler:** Aynı gruba ait örneklerin eğitim ve test setlerinde bulunmaması sağlanarak veri sızıntısının (data leakage) önüne geçilir ve modelin gerçek genelleme kapasitesi daha doğru ölçülür.
- **Gerçekçi Performans Ölçümü Sağlar:** Özellikle tıbbi verilerde, kullanıcı davranış verilerinde veya deneysel çalışmalarda, modelin yeni gruplar üzerindeki başarısı daha iyi tahmin edilir.

Bununla birlikte bazı dezavantajları da bulunmaktadır:

- **Grupların Dengeli Dağılımı Zor Olabilir:** Eğer bazı gruplar çok büyükse veya grup sayısı sınırlıysa, katlar arasında adil bir dağılım yapmak zorlaşabilir.
- **Veri Setinin Bölünmesi Karmaşıktır:** Standart K-Fold'a kıyasla daha karmaşık bir bölme işlemi gerektirir ve uygulanması daha dikkatli yapılmalıdır.

2.4 Tekrarlı K-Katlı Çapraz Validasyon (Repeated K-Fold Cross Validation)

Makine öğrenimi modellerinin genelleme performansını değerlendirmek için kullanılan bir diğer teknik Repeated K-Fold çapraz validasyonudur. Standart K-Fold tekniğinde, veri seti K parçaya ayrılır ve her parça bir defa test verisi olarak kullanılır. Ancak tek bir bölme işlemi, özellikle küçük veya dengesiz veri setlerinde, performans ölçümlerinde varyansa neden olabilir. Repeated K-Fold, bu sorunu azaltmak için K-Fold işlemini birden fazla kez, her seferinde veri setini yeniden karıştırarak tekrarlar. Böylece modelin farklı veri bölmelerine karşı duyarlılığı ölçülür ve daha güvenilir ortalama performans sonuçları elde edilir.[19] Şekil 2.5'te bu tekniğin temel işleyişi basit bir şekilde şematik olarak görselleştirilmiştir.

Şekil 2.5 Tekrarlı K-Katlı Çapraz Validasyon Tekniğinin Şematik Gösterimi

2.4.1 Uygulama Süreci

Repeated K-Fold tekniğinde öncelikle veri seti rastgele karıştırılır ve ardından standart K-Fold çapraz validasyon uygulanır. Bu süreç, önceden belirlenen tekrar sayısı boyunca tekrarlanır. Her tekrarda farklı bir rastgele bölme yapıldığından, model her seferinde farklı eğitim ve test setleri üzerinde değerlendirilir. Sonuçta, tüm iterasyonlardan elde edilen performans metrikleri birleştirilir ve ortalamaları alınarak modelin genel performansı hesaplanır.[20]

2.4.2 Avantajları ve Dezavantajları

Repeated K-Fold CV'nun başlıca avantajları şunlardır:

- **Daha Sağlam Performans Ölçümü:** Birden fazla bölme ile yapılan değerlendirmeler, modelin farklı veri alt kümelerine karşı genelleme yeteneğini daha güvenilir şekilde yansıtır.
- **Varyansı Azaltır:** Performans ölçümlerindeki varyans azalır ve daha stabil bir tahmin elde edilir.

Dezavantajları ise şunlardır:

- **Artan Hesaplama Maliyeti:** K-Fold zaten K kez eğitim gerektirirken, Repeated K-Fold'da bu süreç tekrarlandığı için toplam eğitim sayısı ciddi şekilde artar.
- **Veri Karıştırma Bağımlılığı:** Özellikle zaman serisi verilerde rastgele karıştırma, veri yapısını bozarak hatalı sonuçlar doğurabilir.

2.5 Tekrarlı Katmanlı K-Katlı Çapraz Validasyon (Repeated Stratified K-Fold Cross Validation)

Repeated Stratified K-Fold, Repeated K-Fold tekniğinin sınıf oranlarını koruyan bir türevidir. Özellikle sınıf dengesizliği bulunan veri setlerinde, her katın(fold) ve her tekrarın orijinal veri kümesindeki sınıf oranlarına sadık kalacak şekilde oluşturulması hedeflenir.

Bu teknik hem veri karıştırılması hem de sınıf dengesinin korunması özelliklerini birleştirerek modelin daha adil ve doğru şekilde değerlendirilmesini sağlar.

2.5.1 Uygulama Süreci

Bu teknikte, veri seti her tekrar öncesinde karıştırılır, ardından stratified (katmanlı) bir K-Fold işlemi uygulanır. Yani, her kat(fold), orijinal veri setindeki sınıf dağılımını olabildiğince yansıtır. Bu süreç belirlenen tekrar sayısı boyunca sürdürülür.Elde edilen tüm validasyon sonuçları ortalanarak modelin genel performansı hesaplanır. [21] Şekil 2.6'da bu tekniğin temel işleyişi basit bir şekilde şematik olarak görselleştirilmiştir.

Şekil 2.6 Tekrarlı Katmanlı K-Katlı Çapraz Validasyon Tekniğinin Şematik Gösterimi

2.5.2 Avantajları ve Dezavantajları

Repeated Stratified K-Fold CV'nun başlıca avantajları şunlardır:

- **Sınıf Dengesini Korur:** Özellikle azınlık sınıfların küçük katlar(foldlar) içinde temsil edilmesini sağlar.
- **Daha Güvenilir Değerlendirme:** Hem veri karıştırılması hem de sınıf koruma kombinasyonu, model değerlendirmelerinde daha doğru sonuçlar üretir.

Dezavantajları ise şunlardır:

- **Hesaplama Süresi Uzunluğu:** Çok sayıda tekrar ve stratified bölme işlemi, işlem süresini önemli ölçüde artırabilir.
- **Küçük Sınıflarda Yetersizlik:** Az sayıda örneğe sahip sınıflar için stratification süreci zorluk çıkarabilir.

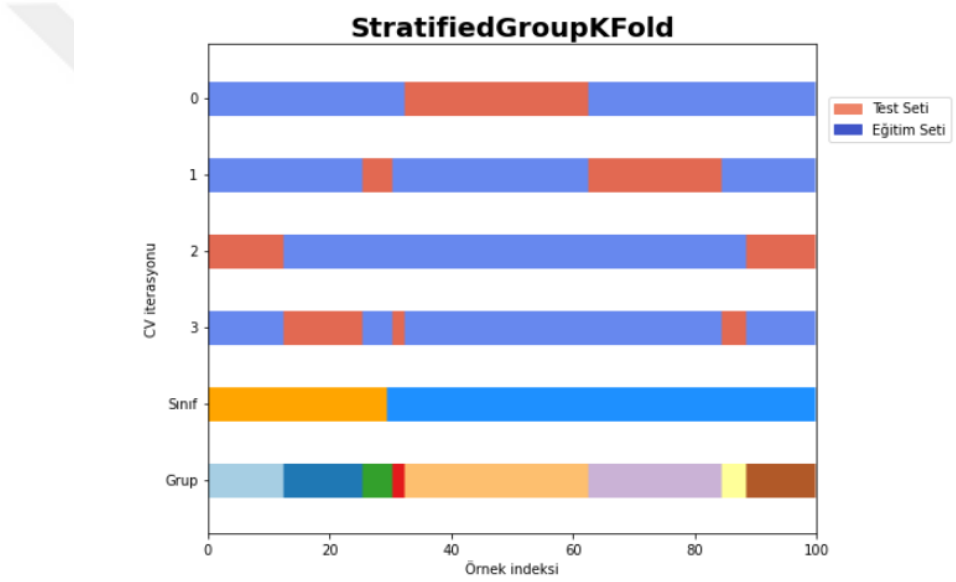
2.6 Katmanlı Gruplu K-Katlı Çapraz Validasyon (Stratified Group K-Fold Cross Validation)

Stratified Group K-Fold çapraz validasyon, grup bağımlılıklarını koruyarak aynı zamanda sınıf dengesini sağlamaya yönelik bir tekniktir. Grup bazlı bağımlılıkların söz konusu olduğu veri setlerinde, aynı gruba ait verilerin hem eğitim hem de test

setlerinde bulunmaması sağlanır. Buna ek olarak, her kat(fold) içerisinde sınıf oranlarının da orijinal veri setine benzer olmasına dikkat edilir. [22]

2.6.1 Uygulama Süreci

İlk adımda her veri örneği bir grup etiketi ile ilişkilendirilir. Daha sonra tüm gruplar, hem grup bağımlılıklarına hem de sınıf oranlarına dikkat edilerek, eşit ya da yaklaşık eşit katmanlara ayrılır. Eğitim ve test setleri, grup bağımlılıkları karıştırılmadan ve sınıf dengesi korunarak oluşturulur. Bu süreç K iterasyon boyunca devam eder ve her iterasyondaki değerlendirme metrikleri birleştirilerek ortalama performans hesaplanır. [23] Şekil 2.7'da bu tekniğin temel işleyişi basit bir şekilde görselleştirilmiştir.



Şekil 2.7 Katmanlı Gruplu K-Katlı Çapraz Validasyon Tekniğinin Şematik Gösterimi

2.6.2 Avantajları ve Dezavantajları

Stratified Group K-Fold CV'nun başlıca avantajları şunlardır:

- **Grup ve Sınıf Bağımlılıklarının Korunması:** Gerçekçi ve dengeli performans ölçümü sağlar.
- **Veri Sızıntısını Önler:** Aynı gruptaki verilerin eğitim ve test setlerine karışmaması sağlanır.

Dezavantajları ise şunlardır:

- **Uygulama Karmaşıklığı:** Stratified ve Group mantığını aynı anda uygulamak bölme algoritmasını zorlaştırır.
- **Dengesiz Gruplarda Zorluk:** Büyük grupların varlığı katlar(foldlar) arası dengeyi bozabilir.

2.7 Zaman Serisi Bölmesi-İleri Zincirleme Çapraz Validasyon (Time Series Split -Forward Chaining Cross Validation)

Zaman serisi verilerinde kullanılan Time Series Split (Forward Chaining) tekniği, zaman bağımlılığını koruyarak model değerlendirmesi yapmayı hedefler. Geleneksel rastgele bölme işlemleri zaman serisi yapısını bozabileceğinden, bu teknik geçmiş verilerle eğitim yapıp, gelecekteki verilerle test gerçekleştirir. [24]

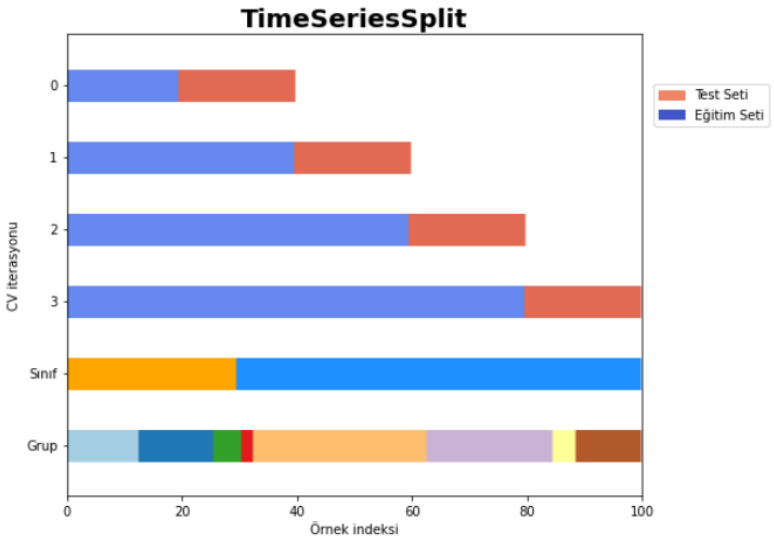
2.7.1 Uygulama Süreci

Zaman Serisi Bölmesi-İleri Zincirleme CV'nun tekniğinin uygulanmasında, zaman serilerinin doğası gereği veri setinin kronolojik sırası korunur ve modelin eğitimi yalnızca geçmiş verilere dayalı olarak gerçekleştirilir. Bu yaklaşım, zaman bağımlılığını dikkate alarak geleceğe yönelik tahminleme yapılmasına olanak tanır. Uygulama süreci, aşağıdaki adımlar doğrultusunda ilerler:

- Veri Setinin Kronolojik Sıralanması: Öncelikle, zaman serisi verisi zaman özelliklerine göre sıralanır. Bu, modelin geleceği yalnızca geçmişe bakarak tahmin etmesini garanti altına almak için kritik öneme sahiptir.
- Eğitim ve Test Setlerinin Belirlenmesi: İlk iterasyonda, veri setinin erken dönemine ait belirli bir kısmı eğitim (train) seti olarak ayrılır. Bu bölümün hemen ardından gelen zaman aralığı test (validation) seti olarak seçilir. Örneğin: Eğitim: [t1, t2, t3], Test: [t4]
- İleri Zincirleme (Forward Chaining) Yaklaşımı: Her yeni iterasyonda eğitim seti, bir önceki test setini de kapsayacak şekilde genişletilir ve test seti de bir sonraki zaman dilimini içerecek şekilde güncellenir. Böylece veri yapısı:
  2. iterasyon: Eğitim: [t1, t2, t3, t4] — Test: [t5]
  3. iterasyon: Eğitim: [t1, t2, t3, t4, t5] — Test: [t6] şeklinde ilerler.
- Zaman Akışının Korunması: Bu zincirleme yapı, zamanın doğal akışını bozmaz ve veri sızıntısını (data leakage) önler. Eğitim seti daima geçmiş

verilere dayanırken, test seti gelecekteki verilerden oluşur. Bu durum, modelin gerçek hayattaki kullanım senaryosuna daha yakın bir değerlendirme yapılmasını sağlar [24].

Şekil 2.8'de bu tekniğin temel işleyişi basit bir şekilde şematik olarak görselleştirilmiştir.



Şekil 2.8 Zaman Serisi Bölmesi Çapraz Validasyon Tekniğinin Şematik Gösterimi

2.7.2 Avantajları ve Dezavantajları

Time Series Split -Forward Chaining CV'nun başlıca avantajları şunlardır:

- **Zaman Bağımlılığına Uyumlu:** Gerçek dünyadaki zaman akışına uygun performans ölçümü sağlar.
- **Veri Sızıntısını Önler:** Test seti, eğitim verisinden önceki dönemi kapsamaz.

Başlıca dezavantajları şunlardır:

- **Eğitim Seti Dengesizliği:** İlk birkaç iterasyonda eğitim seti küçük olduğundan model performansı düşük olabilir.
- **Geleceğe Genelleme Zorluğu:** Bazı zaman serisi yapılarında model, ileri zamana doğru genellemede zorlanabilir.

2.8 Validasyon Tekniklerinin Genel Özeti ve Gelişimi

Validasyon tekniklerinin evrimi ve her bir teknik arasındaki farklar ile gelişim süreçleri, özellikle makine öğrenmesi ve veri bilimi alanlarında giderek daha fazla önem kazanmaktadır. Tablo 2.1, bu tekniklerin ne işe yaradığını, birbirleriyle nasıl ilişkilendiğini ve her birinin önceki teknikleri nasıl geliştirdiğini ayrıntılı bir şekilde özetlemektedir. Bu tablo, farklı validasyon tekniklerinin kullanım alanlarını, avantajlarını ve sınırlamalarını daha iyi anlamaya yardımcı olacak kapsamlı bir genel bakış sunmaktadır.

Tablo 2.1 Validasyon Tekniklerinin Karşılaştırılması ve Evrimi

Teknik Adı	İşlevi	Evrim Noktası
Hold-Out	Veri seti belirli bir oranda eğitim ve test setlerine rastgele bölünür. Model, eğitim verisinde eğitilir ve test verisinde performansı ölçülür.	Eğitim esnasında test verisinin kullanılmamasını sağlaması
K-Fold	Veri seti K eşit kata / parçaya bölünür; her bir parça sırayla test seti olarak kullanılır ve kalanlar eğitim için kullanılır. Nihai başarı performans metriklerinin ortalaması alınarak hesaplanır.	Eğitim ve test verilerinin katlı olarak çoklu yapılması ve Hold-Out'a göre daha tutarlı performans ölçümü sağlaması.
Stratified K-Fold	Her katın hedef sınıf dağılımı hazırlamadan sonraki orijinal veri setindeki hedef özelliğinin sınıf orana uygun olacak şekilde katmanlı oluşturulur.	Katmanlardaki hedef sınıfları, hazırlamadan sonraki orijinal veri setindeki hedef sınıf dağılım oranlarına göre oluşturması ve böylece daha gerçekçi temsiliyetin sağlaması
Group K-Fold	Verileri gruplayarak aynı gruba ait verilerin (ör. kullanıcılar, cihazlar, denekler) aynı kat içinde tutulmasını sağlar.	Grup bağımlılığı olan veri setlerinde katlardaki grupsal bilgi sızmasını engellemesi
Repeated K-Fold	K-Fold prosedürü, farklı rastgele bölmelerle birçok kez tekrarlanır. Böylece varyans azaltılır.	Farklı bölünmelere göre modelin katlardaki stabilitesini sağlaması
Repeated Stratified K-Fold	Her tekrarda sınıf oranı korunarak Stratified K-Fold prosedürü birçok kez farklı rastgele bölmelerle tekrarlanarak uygulanır.	Her tekrar sırasında sınıf oranlarının korunmasını sağlaması ve böylece daha tutarlı ve dengeli tekrarlı validasyon sunması
Stratified Group K-Fold	Veriler gruplara göre ayrılırken aynı zamanda her katmandaki katlarda hedef sınıf oranı da korunur. Özellikle grup bağımlılığı ve sınıf dengesizliği birlikte önemliyse kullanılır.	Hem grup bütünlüğünü hem de her katmandaki katlarda hedef sınıf oranlarının korunması
Time Series Split (Forward Chaining)	Eğitim ve test verileri zaman sırasına göre ayrılır. Eğitim verisi her adımda büyürken test verisi hep geleceği temsil eder. Geçmişe bakarak geleceği tahmin etmeye odaklıdır. Bu teknik zamana özel çalışır; grup ve sınıf dengesini doğrudan sağlamayı hedeflemez.	Zaman serisi verilerde geçmiş verilerle eğitip geleceği tahmin edecek bölmeler yapması ve zaman bağımlılığını koruması.

3
SINIFLANDIRMA ALGORİTMALARI

3.1 Sınıflandırma Algoritması

Sınıflandırma algoritmaları, gözetimli öğrenme (supervised learning) yöntemi içinde yer alır. Eğitim Seti'ndeki (Train Set) verilerinin hedef etiketi sınıflarına dayalı model oluşturmayı gerçekleştirmek için kullanılır ve bu modelden yararlanılarak Test Seti'ndeki (Test Set) veriler veya yeni veriler için en uygun sınıfı belirlemede tahminler yapılır. Günümüzde sınıflandırma, metin madenciliğinden sağlık analizlerine, finansal tahminlerden görüntü tanımaya kadar pek çok alanda etkili bir biçimde kullanılmaktadır.

Bu bölümde, literatürde çoğunlukla tercih edilen sınıflandırma algoritmaları ele alınmıştır. Öncelikle topluluk öğrenme algoritmalarından biri olan Rastgele Orman (Random Forest) algoritması incelenmiştir. Daha sonra artan bir doğruluk sağlamayı hedefleyen Gradyan Artırma (Gradient Boosting) algoritmaları ve bu algoritmaların gelişmiş sürümleri olan XGBoost, LightGBM ve CatBoost detaylandırılmıştır. Son olarak, istatistiksel temelli bir algoritma olan Lojistik Regresyon (Logistic Regression) algoritmasına yer verilmiştir. Algoritmalar kendi içinde avantaj ve dezavantajlara sahip olmakla birlikte, veri türüne ve probleme göre farklı performanslar sergilemiştir. Bu nedenle, sınıflandırma problemi çözülürken birden fazla algoritmanın kullanılması ve sonuçlarının karşılaştırmalı olarak değerlendirmelerin yapılması önem arz etmektedir.

3.1.1 Rastgele Orman Algoritması (Random Forest Algorithm)

Rastgele Orman (RF) algoritması, hem regresyon hem de sınıflandırma problemlerinde kullanılan bir denetimli sınıflandırma algoritmasıdır. Çok boyutlu verilerle etkili çalışır ve eksik veriler veya bilinmeyen veri boyutları durumlarında

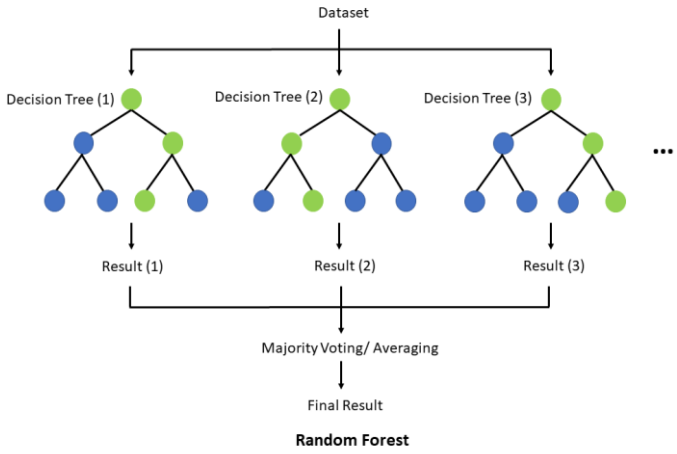
da başarılı sonuçlar elde edebilir. Ayrıca, denetimsiz kümeleme çalışmalarında da kullanımı yaygındır. [25]

Rastgele Orman algoritması, bağımsız çalışan ağaçlar kullanarak yüksek başarıya ulaşmayı hedefler. Ağaç sayısı arttıkça daha tutarlı sonuçlar alınır. Algoritma, temelde Karar Ağacı algoritmasına dayanır ancak kök düğümünün ve düğüm bölme işlemlerinin rastgele belirlenmesi ile fark yaratır.

Algoritma dört temel adımda çalışır:

- Birden fazla ağaç oluşturulur.
- Yeni bir veri noktası sınıflandırılmak üzere her ağaçla test edilir.
- Her ağaç, verdiği sınıflandırma ile o sınıf için oy kullanır.
- Orman, en fazla oyu alan sınıfı seçerek sınıflandırmayı tamamlar. [26]

RF algoritmasının aşamaları, aşağıda yer alan Şekil 3.1'de şematik olarak sunulmuştur.

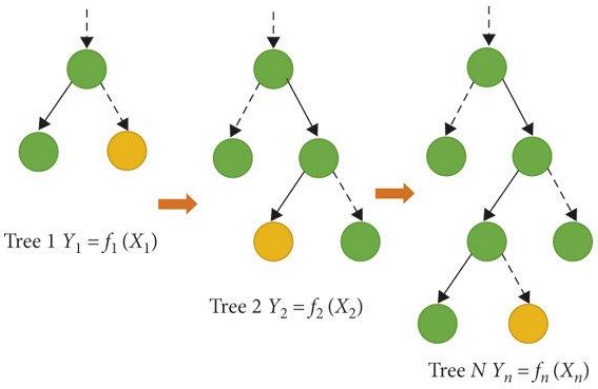


Şekil 3.1 Rastgele Orman Algoritması Sınıflandırma Örneği [27]

3.1.2 Gradyan Artırma Algoritması (Gradient Boosting Algorithm)

Gradyan Artırma (GB), bir tür artırma algoritmasıdır ve sınıflandırma ile regresyon problemlerinde etkin bir şekilde kullanılır. Bu algoritma, ilk kez 2001 yılında Friedman tarafından önerilmiştir. [28] Diğer tekniklerden farklı olarak, bu algoritmanın en belirgin özelliği, hata oranlarını minimize etmek için gradyan inişi yöntemini kullanmasıdır. [25]

Gradyan Artırma Sınıflandırma Algoritması, başlangıçta verileri basit modellerle işler ve ardından hatalı verileri incelemeye alır. Bu hatalar, sınıflandırılması zor verileri belirlememize yardımcı olur ve algoritma, öncelikle bu tür verilere odaklanır. Son adımda, tüm tahminler üzerinde her bir belirleyiciye yeterli ağırlık verilerek modelin tamamlanması sağlanır. Gradyan Artırma algoritmasının bu aşamaları, aşağıda yer alan Şekil 3.2'de şematik olarak sunulmuştur.[29]



Şekil 3.2 Gradyan Artırma Algoritmasını Açıklayan Şematik Diyagramı [30]

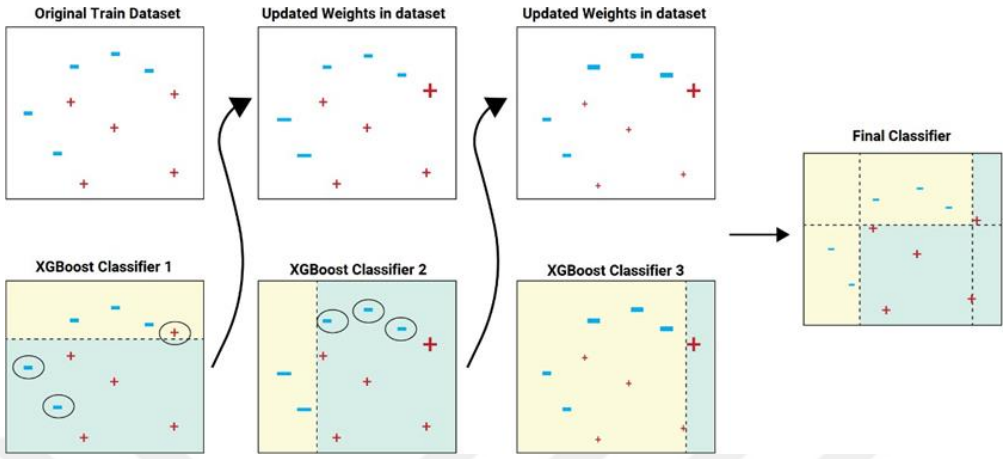
3.1.3 Aşırı Gradyan Artırma Algoritması (Extreme Gradient Boosting Algorithm)

Aşırı Gradyan Artırma (XGB) Sınıflandırma Algoritması, özellikle yeni olması ve yüksek başarı oranları ile diğer algoritmalar arasında dikkat çeker. GB algoritmasının daha gelişmiş bir versiyonudur. Modelin başarısı, eğitim sürecinde aşırı uyum sorunlarının minimize edilmesiyle doğru orantılı olarak artmaktadır. Aşırı uyum engellenebildikçe, modelin başarısı yükselir. 2016 yılında Chen ve Guestrin tarafından yazılan bir makale ile sınıflandırma algoritmaları arasında kendine yer bulmuştur.[31]

Bu algoritma, diğer popüler algoritmalara kıyasla çok daha hızlı çalışır. Daha az kaynak kullanarak daha verimli sonuçlar elde etmeyi amaçlayan çeşitli yazılım ve donanım optimizasyonları içerir. Karar ağaçlarına dayalı algoritmalar arasında en iyilerinden biri olarak kabul edilmektedir.

XGB algoritması, verileri küçük parçalara bölerek analiz eder ve bu parçalar üzerinde çalışır. Bu sayede, her bir veriye ayrı ayrı bakmaya gerek kalmadan, daha verimli sonuçlar elde edilir. Parçaların sayısı artırıldıkça, algoritma daha küçük

detayları inceleyerek tahminlerde daha doğru sonuçlar verir. Şekil 3.3'te, XGB algoritmasının nasıl çalıştığını adım adım açıklayan bir örnek yer almaktadır.



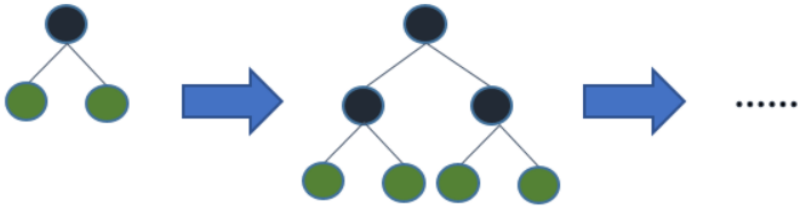
Şekil 3.3 Aşırı Gradyan Artırma Algoritması Sınıflandırma Örneği [32]

3.1.4 Light Gradyan Artırma Makinası Algoritması (Light Gradient Boosting Machine Algorithms)

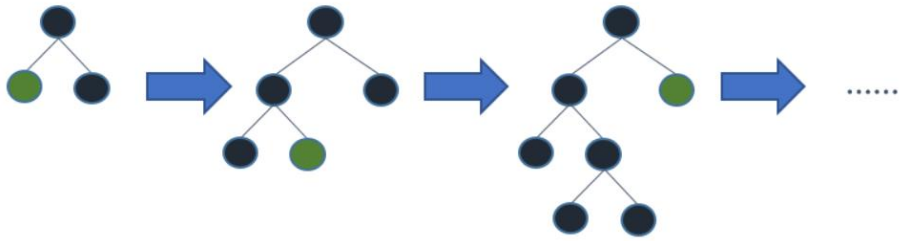
Light Gradyan Artırma Makinası (LightGBM) algoritması, Microsoft tarafından 2017 yılında geliştirilen ve GB'dan daha verimli hale getiren bir makine öğrenmesi algoritmasıdır. Özellikle büyük veri setlerinde hızlı eğitim süreleri ve yüksek doğruluk oranları sağlamak amacıyla tasarlanmıştır. LightGBM, hem sınıflandırma hem de regresyon problemleri için etkili bir çözüm sunar ve diğer GB algoritmalarına kıyasla daha hızlı çalışır. Bunun yanı sıra, daha az bellek kullanarak daha hızlı sonuçlar elde edilmesini sağlar.[33]

LightGBM'nin başarısı, veri işleme ve ağacın büyüme stratejilerindeki yenilikçi yaklaşımlarına dayanır. Algoritma, veriyi hızlı bir şekilde işlemek için histogram tabanlı bir yaklaşım kullanır. Bu sayede eğitim süresi önemli ölçüde azalır. Ayrıca, LightGBM, karar ağaçlarını dal bazlı (leaf-wise) büyütme stratejisi ile oluşturur. Bu yöntem, seviye bazlı (level-wise) büyümeye kıyasla daha derin ağaçlar oluşturulmasına ve böylece daha yüksek doğruluk elde edilmesine olanak tanır.[34]

Seviye bazlı (level-wise) ve dal bazlı (leaf-wise) büyüme stratejileri arasındaki fark aşağıdaki Şekil 3.4 ve 3.5'te görsel olarak verilmiştir.



Şekil 3.4 Seviye Bazlı (Level-Wise) Büyüme



Şekil 3.5 Dal Bazlı (Leaf-Wise) Büyüme [35]

LightGBM, büyük veri kümeleriyle çalışırken performans açısından önemli bir avantaj sunar. Hem hızlı eğitim süreleri hem de verimli bellek kullanımı ile büyük veri işlemlerinde tercih edilmektedir. Bu algoritma, finansal analizlerden sağlık sektörüne kadar geniş bir uygulama yelpazesinde kullanılmaktadır ve bu alanlardaki veri setlerinde yüksek doğruluk ile çalışmaktadır.

3.1.5 Kategori Artırma Algoritması (Category Boosting Algorithms)

Kategori Artırma Algoritması (CatBoost) algoritması, 2017 yılında Yandex tarafından geliştirilmiştir. [36] Adını "Category" ve "Boosting" kelimelerinin birleşiminden alan algoritma, özellikle kategorik verilerle çalışırken üstün performans sergileyen, GB tabanlı bir sınıflandırma algoritmasıdır. CatBoost, GB algoritmalarının gelişmiş bir versiyonu olup, hem doğruluğu hem de kullanım kolaylığı ile öne çıkmaktadır.

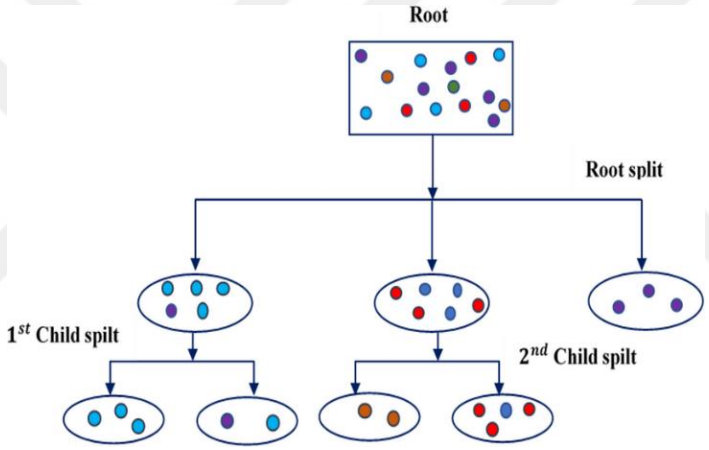
CatBoost algoritmasının temel çalışma prensibi şu adımlardan oluşur:

- Eğitim verisi üzerinde sıralı öğrenme (ordered boosting) tekniği kullanılarak overfitting riski azaltılır.
- Kategorik özellikler, önceden kodlamaya (encoding) gerek kalmadan, algoritma içinde doğrudan işlenir.

- Öznitelik mühendisliği süreçleri otomatik olarak optimize edilir.
- Model, GPU desteği ile yüksek boyutlu veri setlerinde hızlı bir şekilde eğitilebilir.

CatBoost'un en önemli avantajlarından biri, veri sızıntısını (data leakage) önlemek amacıyla geliştirilmiş sıralı veri işleme yöntemidir. Bu sayede model, eğitim sırasında hedef özellikle ilgili ileriye dönük bilgiye erişmeden karar verir. Ayrıca modelin hiperparametre ayarlamalarına karşı diğer algoritmalara göre daha dayanıklı olduğu bilinmektedir.

CatBoost özellikle dengesiz ve yüksek boyutlu veri kümeleri üzerinde başarılı sonuçlar verdiğinden dolayı birçok sınıflandırma probleminde tercih edilmektedir. Şekil 3.6, CatBoost algoritmasının genel işleyiş yapısını göstermektedir.[37]



Şekil 3.6 Kategori Artırma Algoritması (Catboost) [38]

3.1.6 Lojistik Regresyon Algoritması (Logistic Regression Algorithm)

Lojistik Regresyon (LogReg / LR), istatistiksel bir sınıflandırma algoritmasıdır. Genellikle ikili sonuçlar veren kategorik verilerde kullanılır. Model, açıklayıcı özellik ile yanıt özelliği arasındaki ilişkiyi gösteren doğrusal (yaygın olarak) bir model olarak çalışır. [39]

LR, olasılık tahminleri yapmak için sıklıkla tercih edilen bir algoritmadır, çünkü 0 ile 1 arasında sınırlandırılmış olasılıklar üretmek için matematiksel olarak uygun bir yapıya sahiptir. Ayrıca, parametre tahminleri oldukça kolaydır. Uzun yıllardır bilinen ve çok kullanılan bir algoritma olması nedeniyle sıklıkla tercih edilmektedir. [40]

3.2 Değerlendirme Metrikleri

Makine öğrenmesi modellerinin geliştirilmesi kadar, bu modellerin doğruluk, güvenilirlik ve genellenebilirlik açısından değerlendirilmesi de son derece kritiktir. Bir modelin yalnızca yüksek doğruluk üretmesi, onun gerçek dünya uygulamalarında da başarılı olacağı anlamına gelmez. Bu nedenle, sınıflandırma problemlerinde modelin başarısını çok yönlü biçimde analiz edebilen çeşitli performans metrikleri geliştirilmiştir.

Değerlendirme metrikleri; modelin doğru ve yanlış tahminlerini analiz etmeye, sınıflar arasındaki dengesizlikleri ortaya koymaya ve modelin genel doğruluğunun ötesinde nasıl davrandığını göstermeye yardımcı olur. Bu metrikler, özellikle denetimli öğrenme algoritmalarında, eğitim sürecinin sonunda elde edilen sonuçların objektif olarak karşılaştırılmasını sağlar. Karışıklık matrisi bu metriklerin temelini oluşturmakta olup; Doğruluk (Accuracy), Kesinlik (Precision), Duyarlılık (Recall), F1-Skoru ve Gini katsayısı gibi çeşitli metriklere temel teşkil eder. Bu bölümde, sınıflandırma algoritmalarının değerlendirilmesinde kullanılan başlıca metrikler detaylı bir biçimde ele alınmıştır.

3.2.1 Karışıklık Matrisi (Confusion Matrix)

Modelin başarısını değerlendirmek için kullanılan çeşitli metrik bulunmaktadır. Bu metriklerin gelişiminde, karışıklık matrisi (confusion matrix) büyük bir rol oynamaktadır. Karışıklık matrisi, tahmin edilen değerler ile gerçek değerlerin karşılaştırılmasını sağlayan bir yapıya sahiptir. Bu matristen türetilen metrikler, modelin performansını daha iyi değerlendirebilmek için kullanılır. [41]

Tablo 3.1 Karışıklık Matrisi

Toplam Kayıt Sayısı= P + N	**Tahmin Positif**	**Tahmin Negatif**
Gerçek Pozitif (P = TP + FN)	TP: Tahmin Doğru	FN: Tahmin Yanlış
Gerçek Negatif (N = FP + TN)	FP: Tahmin Yanlış	TN: Tahmin Doğru

Tablo 3.1'deki karışıklık matrisindeki parametreler aşağıda açıklanmıştır:

- TP: Gerçekte 1 olan sınıfın 1 olarak tahmininin doğru yapıldığı kayıt sayısı

- FN: Gerçekte 1 olan sınıfın 0 olarak tahmininin yanlış yapıldığı kayıt sayısı
- FP: Gerçekte 0 olan sınıfın 1 olarak tahmininin yanlış yapıldığı kayıt sayısı
- TN: Gerçekte 0 olan sınıfın 0 olarak tahmininin doğru yapıldığı kayıt sayısı

Bu parametrelere göre çeşitli performans metrikleri oluşturulmuştur. Alt başlıklarda sadece bu tezde kullanılan metrikler özet olarak verilmektedir.

3.2.2 Doğruluk (Accuracy)

Gerçekte 1 olan sınıfın 1 ve 0 olan sınıfın 0 olarak tahminin doğru yapıldığı toplam kayıt sayısının tüm kayıtların sayısına oranını gösterir. (3.1) numaralı formül ile gösterilir. Bu değerin 1'e yaklaşması, modelin genel sınıflandırma performansının yüksek olduğunu gösterir.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{3.1}$$

3.2.3 Duyarlılık (Recall / Sensitivity)

Gerçekte 1 olan sınıfın 1 olarak tahmininin doğru yapıldığı kayıt sayısının gerçekte 1 sınıfına ait olan toplam kayıtların sayısına oranını gösterir. (3.2) numaralı formül ile gösterilir. Duyarlılık (Recall) değerinin 1'e yaklaşması, modelin gerçek pozitifleri yakalama becerisinin yüksek olduğunu gösterir.

$$Recall = \frac{TP}{TP+FN} \tag{3.2}$$

3.2.4 Kesinlik (Precision)

Gerçekte 0 olan sınıfın 0 olarak tahmininin doğru yapıldığı kayıt sayısının gerçekte 0 sınıfına ait olan toplam kayıtların sayısına oranını gösterir. (3.3) numaralı formül ile gösterilir. Precision değerinin 1'e yaklaşması, modelin pozitif tahminlerinin isabet oranının yüksek olduğunu gösterir.

$$Precision = \frac{TN}{TN+FP} \tag{3.3}$$

3.2.5 F1-Skoru (F1-Score)

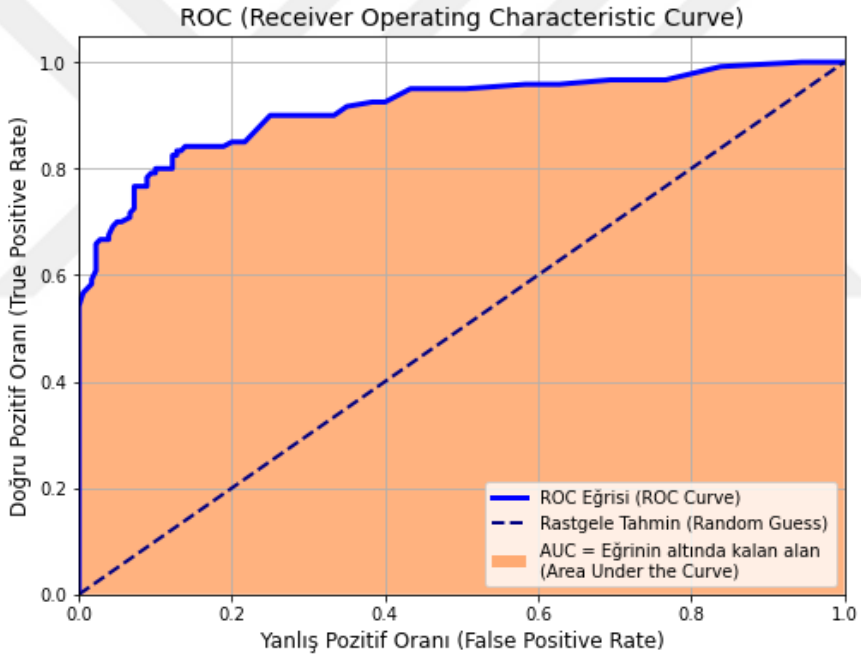
Kesinlik ve duyarlılık değerlerinin harmonik ortalamasıdır. (3.4) numaralı formül ile gösterilir. F1-Skorunun 1'e yaklaşması, modelin hem isabetli hem de kapsayıcı tahminler yaptığını gösterir.

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (3.4)$$

3.2.6 ROC Eğrisi (Receiver Operating Characteristic Curve)

ROC (Receiver Operating Characteristic) eğrisi, sınıflandırma **algoritmalarının** başarı performansını değerlendirmek için kullanılan görsel bir araçtır. ROC eğrisi, True Positive Rate (TPR) ile False Positive Rate (FPR) arasındaki ilişkiyi gösterir. Modelin farklı eşik değerleri için bu oranların grafiği çizilerek elde edilir. Eğrinin altında kalan alan, modelin ayırt edicilik yeteneği hakkında bilgi verir. Şekil 3.7'de verilmiştir**.**

- True Positive Rate (TPR) = Recall = $\frac{TP}{TP+FN}$ (3.2)
- False Positive Rate (FPR) = $\frac{FP}{FP+TN}$ (3.5)



Şekil 3. 7 ROC Eğrisi (ROC Curve)

Eğer ROC eğrisi, köşegenin (FPR = TPR) oldukça üzerinde yer alıyorsa bu durum modelin pozitif ve negatif sınıfları yüksek doğrulukla ayırt ettiğini gösterir. Eğrinin köşegene yakın olması ise modelin sınıflar arasında ayrım yapamadığını, rastgele tahminleme düzeyinde çalıştığını gösterir. [42]

3.2.7 AUC (Area Under the Curve)

AUC (Area Under the ROC Curve), ROC eğrisinin altında kalan alanın büyüklüğünü ifade eder ve sınıflandırma modelinin ayırt etme yeteneğinin tek bir sayısal değerle özetlenmesini sağlar.

$$\text{AUC} = \int_0^1 \text{TPR}(x)\ dx\ \ (x = \text{FPR}) \tag{3.6}$$

AUC değeri 0 ile 1 arasında değişir.

- AUC = 1 → Mükemmel sınıflandırma performansı
- AUC = 0.5 → Rastgele tahmin (model ayırt edici değil)
- AUC <0.5 → Modelin sınıfları ters sınıflandırdığı, yani kötü performans sergilediği anlamına gelir

Yüksek AUC değeri, modelin pozitif ve negatif sınıfları başarılı biçimde ayırabildiğini gösterir. [43]

3.2.8 Gini Katsayısı (Gini Coefficient)

Gini katsayısı, sınıflandırma modelinin ayırt edici gücünü ölçen ve AUC değeri üzerinden hesaplanan bir metriktir. Bu katsayı özellikle sigorta, finans ve pazarlama alanlarında yaygın biçimde kullanılmaktadır. [44] Gini katsayısı aşağıdaki formülle ifade edilir:

$$\text{Gini} = 2 \cdot \text{AUC} - 1 \tag{3.7}$$

Gini katsayısı **-1 ile 1** aralığında değer alabilir:

- **Gini ≈ 1** → Mükemmel ayırt edicilik
- **Gini ≈ 0** → Ayırt edicilik yok (rastgele tahmin düzeyi)
- **Gini < 0** → Modelin sınıfları ters sınıflandırdığı, yani kötü performans gösterdiği anlamına gelir

4
VERİ HAZIRLIĞI

4.1 Veri Toplama

Hasar dosyalarının Dava açma tahmin modeli kapsamında kullanılacak veri seti, özel bir sigorta şirketinin hasar dosyalarıyla ilgili geçmiş kayıtlarından oluşturulmuştur. Çalışmada, doğrudan hazır bir veri seti kullanılmamış; ilgili sigorta şirketinin veri tabanında bulunan hasar dosyalarına ait detaylar, şirket içi raporlar ve ek harici kaynaklardan elde edilen bilgiler bir araya getirilerek kapsamlı gerçek değerleri içeren bir veri seti oluşturulmuştur.

Veri setinin ham hali, **817.738 kayıt (satır) ve 155 özellikten (sütun)** oluşmaktadır. Bu geniş veri seti, hasar dosyalarının Dava Açma ihtimalini tahmin etmek üzere çok boyutlu ve detaylı bilgi sunmaktadır.

Toplanan veriler aşağıda verilen farklı kaynaklardan bir araya getirilmiştir:

1. **Sigorta Şirketi Veri Tabanı:** Hasar dosyalarına ilişkin temel bilgileri içeren kayıtlar.
2. **Sigorta Raporları:** Eksper raporları, ret/kabul durumları ve ilgili özellikler.
3. **Harici Kaynaklar:** Dava açma süreçlerine dair geçmiş davalar ve müşteri şikâyet kayıtları.

Veri setinin oluşturulmasında gereksiz veya tekrar eden veriler makine öğrenmesi algoritmaları açısından anlamlı bir sonuç vermeyeceği için temizlenmiştir. Aşağıda, veri setine dâhil edilen temel özellikler listelenmiştir:

- **Hasar Dosyasına Ait Bilgiler:** Hasar toplam tutarı, ön rapor, ihbar başvuru şekli, hasar tarihi, ihbar tarihi, hasar şekli.
- **Sigortalı ile İlgili Bilgiler:** Müşteri tipi, hasarsızlık kademesi, araç kullanım şekli, marka, model yılı.

- **Hasar Durumu ve Süreç Bilgileri:** Rücu durumu, tam hasar mı, faturalı iş mi, servis türü, onarım işlemi yapıldı mı gibi bilgiler
- **Dava Açma Süreç Bilgileri:** Daha önceki benzer hasar dosyalarının dava edilme oranları, reddedilen hasarların sonucu.

Veriler Python programlama dili kullanılarak Jupyter Notebook ortamında birleştirilmiş ve ön işleme sürecinden geçirilmiştir. Veri seti oluşturulduktan sonra, gereksiz sütunlar çıkarılarak nihai veri seti belirlenmiştir.

Toplanan tüm veriler için özelliklerin tanımı, veri tipi, toplam kaç farklı değere sahip olduğu ve sayısal özelliklerin minimum-maksimum değerleri detaylandırılarak analiz edilmiştir. Tablo 4.1'de veri setine ait tüm özellikler detaylı olarak açıklanmaktadır.

Tablo 4.1 Veri Setine Ait Özellikler

Özellik Adı	Tanım	Tip	Farklı Değer Sayısı	Min.	Max.
DAVA_ACMA_DOSYASI ILISKISI	Dava açılmış olma durumu (0=Yok, 1=Var)	Kategorik	2	-	-
MUSTERI_TIPI	Müşteri tipi (bireysel, tüzel)	Kategorik	2	-	-
HASAR_TARIHI	Hasarın gerçekleştiği tarih	Tarih	4880	2007-02-19	2022-08-24
HASAR_KALEMI_ACILIS_TARIHI	Hasar kaleminin açıldığı tarih	Tarih	4360	2007-12-11	2022-08-25
IHBAR_TARIHI	Hasarın sigortaya ihbar edildiği tarih	Tarih	4270	2007-12-11	2022-08-24
HASAR_SEBEBI	Hasarın meydana gelme sebebi (kaza, doğal afet vb.)	Kategorik	9	-	-
TRAMER_HASAR_SEBEBI	Araç geçmişindeki trafik hasar sebebi	Kategorik	4	-	-
ACENTE_IL_ADI	Sigorta acentesinin bulunduğu il	Kategorik	81	-	-
TAM_HASAR_MI	Araç tam hasarlı mı (evet/hayır)	Kategorik	2	-	-
IHBAR_BASVUTU_SEKLI	Hasar ihbar başvuru şekli (online, telefon, vb.)	Kategorik	2	-	-

Tablo 4.1 Veri Setine Ait Özellikler (devamı)

Özellik Adı	Tanım	Tip	Farklı Değer Sayısı	Min.	Max.
ARAC_SINIFI	Araç sınıfı (sedan, SUV, hatchback vb.)	Kategorik	39	-	-
YAKIT_TIPI	Araç yakıt tipi (benzin, dizel, elektrikli vb.)	Kategorik	20	-	-
MODEL_YILI	Araç üretim yılı	Sayısal	71	1946	2022
MARKA	Araç markası (Ford, BMW, Toyota vb.)	Kategorik	367	-	-
BEYGIR_GUCU	Araç beygir gücü	Sayısal	689	0	8974
OTOA_YEDEKPARCA_ TUTAR	Yedek parça değişimi maliyeti	Sayısal	161392	0	1.922.905
OTOA_ISCILIK_TUTAR	İşçilik maliyeti	Sayısal	21967	0	400.001
OTOA_TOPLAM_TUTAR	Toplam işlem maliyeti (yedek parça, işçilik vb.)	Sayısal	175805	0	2.382898
OTOA_DTY_DEGISIM_ ISLEMI_ VAR_MI	Yedek parça değişimi yapıldı mı (evet/hayır)	Kategorik	2	-	-
OTOA_DTY_DEGISIM_ ISLEM_ ADEDI	Yedek parça değişim işlemi sayısı	Sayısal	178	0	436
OTOA_SIGORTALI_ ARAC_KM	Sigortalı aracın kilometresi	Sayısal	5611	0	111111111
OTOA_KUSUR_ORANI	Araç kusur oranı	Sayısal	40	0	100
HASAR_SAYI	Hasar sayısı	Sayısal	387	0	16816
HASAR_TPL_TUT	Toplam hasar tutarı	Sayısal	65900	0	793.724.360
HASAR_ORT_TUT	Ortalama hasar tutarı	Sayısal	68247	0	49.607.773
HASAR_RED_FLAG	Hasar reddedildi mi (evet/hayır)	Kategorik	2	-	-

4.2 Veri Ön-İşleme

Makine öğrenmesi projelerinde, verilerin ilk toplanmasından modellerin çalıştırılmasına kadar geçen süreçte çeşitli veri ön işleme adımları kritik rol oynamaktadır. Veri ön işleme süreci, projelerin başarısını doğrudan etkileyen temel aşamalardan biridir. Kullanılan öğrenme algoritması ne kadar güçlü olursa olsun, eğer ön işleme adımları eksik veya hatalı yürütülürse, projenin sağlıklı sonuçlar üretmesi mümkün olmayabilir. Bu bölümde, söz konusu veri işleme süreçleri ayrıntılı biçimde ele alınmaktadır.

4.2.1 Eksik ve Gürültülü Veri Temizliği

Veri setinde yer alan eksik ve gürültülü değerler, modelin öğrenme sürecini olumsuz etkileyebilecek en temel sorunlar arasında yer almaktadır. Bu nedenle, ilk olarak özelliklerdeki eksik veri oranları analiz edilmiştir. Eksik değer oranı %90'dan fazla olan özelliklerin bilgi taşıma kapasitesinin düşük olduğu değerlendirilmiş ve bu özellikler veri setinden çıkarılmıştır. Yapılan analiz sonucunda, %90'dan fazla eksik değere sahip 21 adet özellik tespit edilmiştir. Bu adımlar sayesinde daha dengeli ve güvenilir bir modelleme zemini oluşturulmuştur.

4.2.2 Veri Entegrasyonu ve Dönüşümü

Ham veriler üzerinde yapılan işlemler sonucunda, veri seti çeşitli dönüşüm ve entegrasyon adımlarından geçirilmiştir. Bu adımlar, kimi zaman anlamlı bilgi içermeyen, tekrarlı ya da modelin performansını olumsuz etkileyebilecek nitelikteki özelliklerin çıkarılmasını; kimi zaman ise mevcut verilerden yeni özellikler türetilmesini kapsamaktadır. Özellikle makine öğrenmesi uygulamalarında, modelin öğrenme kapasitesini artırmak amacıyla yeni ve anlamlı özellikler oluşturmak önemli bir adımdır. Bu çalışma kapsamında da, mevcut özellikler temel alınarak hem açıklayıcı hem de hedef özelliklere katkı sağlayacak yeni özellikler türetilmiş ve veri setine entegre edilmiştir.

4.2.2.1 Tarih Özellikleri

Veri setindeki tarih ve saat bilgileri içeren özelliklerden, özellikle hasar tarihi gibi kritik zaman bilgileri içerenlerden daha anlamlı ve modelin öğrenme performansını

artıracak yeni özellikler türetilmiştir. Bu işlem, tarihsel verilerin detaylı analizini mümkün kılmak ve model doğruluğunu artırmak amacıyla gerçekleştirilmiştir.

Özellikle Hasar Tarihi, Hasar Kalemi Açılış Tarihi ve İhbar Tarihi gibi üç ana tarih özelliğinden çeşitli zaman bileşenleri çıkarılmıştır. Her bir tarih özelliği için aşağıdaki türetilmiş özellikler oluşturulmuştur:

- Yıl (HASAR_TARIHI_year)
- Ay (HASAR_TARIHI_month)
- Çeyrek (quarter) (HASAR_TARIHI_quarter)
- Haftanın günü (day of week) (HASAR_TARIHI_dayofweek)
- Yıl içindeki gün (day of year) (HASAR_TARIHI_dayofyear)
- Ay içindeki gün (day of month) (HASAR_TARIHI_dayofmonth)
- Mevsim (season) (HASAR_TARIHI_season)
- Hafta sonu olup olmadığı (is weekend) (HASAR_TARIHI_is_weekend)
- Ay başlangıcı (is month start)
- Ay sonu (is month end)
- Çeyrek başlangıcı (is quarter start)
- Çeyrek sonu (is quarter end)
- Yıl başlangıcı (is year start)
- Yıl sonu (is year end)

Bunlara ek olarak, döngüsel (periodik) zaman bileşenlerinin model tarafından daha iyi kavranabilmesi için sinüs ve kosinüs dönüşümleri de uygulanmıştır:

- Yıl içindeki günün sinüs ve kosinüs değerleri (HASAR_TARIHI_dayofyear_sin, HASAR_TARIHI_dayofyear_cos)
- Ayın sinüs ve kosinüs değerleri (HASAR_TARIHI_month_sin, HASAR_TARIHI_month_cos)

Bu kapsamlı tarihsel özellik mühendisliği, yıl bazlı eğilimlerin, mevsimsel değişimlerin ve haftanın özel günlerinin modele yansıtılmasını sağlamaktadır.

Böylece, tarih bilgileri hem lineer hem de periyodik olarak modele entegre edilerek, Dava Açma tahmini performansı artırılmıştır. Tablo 4.2'de tarih özelliklerinden türetilen bazı örnek özellikler sunulmuştur.

Tablo 4.2 Dönüştürülmüş Tarih Özellikleri

Hasar_Tarih	Year	Month	Day	Weekday	Week	Is_Weekend	Is_Quarter_Start
2020-03-15	2020	3	15	Pazar	11	True	True
2021-06-07	2021	6	7	Pazartesi	23	False	False
2022-09-20	2022	9	20	Salı	38	False	False
2023-12-25	2023	12	25	Pazar	52	True	False

4.2.2.2 Tarih Farkları Özellikleri

Veri setindeki bazı önemli olaylara ait tarih bilgilerinden hareketle, bu olaylar arasındaki süre farkı gün cinsinden hesaplanarak yeni özellikler türetilmiştir. Örneğin, ihbar tarihi ile hasar tarihi, hasar kalemi açılış tarihi ile ihbar tarihi gibi tarih çiftleri dikkate alınmıştır. Bu işlem sayesinde modelin olayların zamanlaması ile ilişkili örüntüleri daha iyi öğrenmesi amaçlanmıştır. Bu kapsamda oluşturulan özellikler:

- HASAR_KALEMI_ACILIS_TARIHI_IHBAR_TARIHI_DAY_DIFF
- HASAR_KALEMI_ACILIS_TARIHI_HASAR_TARIHI_DAY_DIFF
- IHBAR_TARIHI_HASAR_TARIHI_DAY_DIFF

Şeklinde veri setine eklenmiştir. Bu özellikler, olaylar arasındaki sürenin suç/hata/hasar senaryosu üzerindeki etkisini analiz edebilmek amacıyla modellemeye dahil edilmiştir.

4.2.2.3 Kategorik Özelliklerin İşlenmesi

Veri setindeki kategorik veriler, genellikle metin veya etiket değerleri içerir. Bu tür veriler, modelleme sürecinde doğru şekilde işlenmelidir. İlk olarak, object(string) veri tipine sahip özellikler kategorik olarak belirlenmiştir. Ayrıca, isimlerinde VAR_MI, FLG veya FLAG terimlerini içeren özelliklerde kategorik veri olarak işaretlenmiştir.

Son olarak, bu kategorik veriler kategorik veri tipine dönüştürülmüştür. Bu işlem, belleği optimize ederek modelleme sırasında daha hızlı hesaplamalar yapılmasını sağlar.

4.2.3 Kodlama (Encoding)

Makine öğrenmesi algoritmalarının yalnızca sayısal verilerle çalışabilmesi nedeniyle, modelleme sürecinde kullanılan tüm kategorik özelliklerin uygun biçimde sayısal formatlara dönüştürülmesi gerekmektedir. Bu amaçla çalışmada iki farklı kodlama (encoding) tekniği uygulanmıştır:

- Etiket Kodlaması (Label Encoding)
- TF-IDF Kodlaması (Term Frequency–Inverse Document Frequency Encoding)

4.2.3.1 Etiket Kodlaması (Label Encoding)

Etiket Kodlaması (LE) kullanılarak her benzersiz kategoriye bir tamsayı değeri atanmış ve bu sayede veri setindeki tüm kategorik değerler sayısal değerlere dönüştürülmüştür.

LE tekniğiyle her kategorik değerler, modelin anlayabileceği sayısal bir formata getirilmiş ve sıralama gerektirmeyen kategorik özelliklerde kullanılabilir hale getirilmiştir. Tablo 4.3'te kodlama uygulanan özellikler listelenmiştir.

Tablo 4.3 Etiket Kodlaması Uygulanan Özellikler

Özellik Adı	Veri Tipi	Encoding Tekniği
OTOA_HASAR_SEKLI	Kategorik	Label Encoding
OTOA_ARAC_KULLANIM_SEKLI	Kategorik	Label Encoding
OTOA_ARAC_MARKA	Kategorik	Label Encoding
OTOA_ARAC_MODEL	Kategorik	Label Encoding
OTOA_ARAC_TIP	Kategorik	Label Encoding
POLICE_TURU	Kategorik	Label Encoding
MUSTERI_SEGMENTI	Kategorik	Label Encoding
KULLANIM_TIPI	Metin	TF-IDF Encoding

Örneğin Tablo 4.4'te OTOA_HASAR_SEKLI özelliğine LE uygulandığında özelliğin son durumu verilmiştir.

Tablo 4.4 Hasar Şekli Özelliği için Etiket Kodlaması Örneği

OTOA_HASAR_SEKLI	LE Sonrası Durum
Çarpışma	0
Cisme Çarpma	1
Araç Park Halinde	2
Çarpışma	0
Cam	3
Devrilme/Takla Atma	4
Yanma	5
Hayvana Çarpma	6
Radyo/Teyp Hırsızlığı	7

4.2.3.2 TF-IDF Kodlaması (Term Frequency–Inverse Document Frequency Encoding)

Metin yapısına sahip olan "KULLANIM_TIPI" özelliği, içerdiği anlamsal farklılıkların daha etkin şekilde temsil edilebilmesi amacıyla TF-IDF Kodlaması (Term Frequency–Inverse Document Frequency Encoding) kullanılarak sayısal değerlere dönüştürülmüştür. TF-IDF Kodlaması, bir kelimenin bir dokümanda ne kadar önemli olduğunu ölçen bir tekniktir ve metin içerisindeki sık ancak ayırt edici olmayan kelimelerin etkisini azaltarak, modelin daha anlamlı kelimelere odaklanmasını sağlar.

TF-IDF dönüşümü uygulanırken aşağıdaki adımlar izlenmiştir:

- Tüm metinler, büyük/küçük harf uyumsuzluklarının giderilmesi amacıyla küçük harfe dönüştürülmüştür.
- Türkçeye özgü karakterler (ç, ğ, ş, ö, ü, ı vb.) sadeleştirilmiş, Unicode uyumsuzlukları giderilmiştir.
- TF-IDF vektörleştirme işlemi sırasında en fazla 20 farklı kelime dikkate alınarak her biri için ayrı sütunlar oluşturulmuştur.

Bu işlem sonucunda "KULLANIM_TIPI" özelliğinden elde edilen örnek özellikler Tablo 4.5'te gösterilmiştir.

Tablo 4.5 Kullanım Tipi Özelliği için Kodlama Tekniği Örneği

KULLANIM_TIPI	Otomobil	Sürücü	Koltuk	Çekici	Otobüs	Minibüs
OTOMOBİL (SÜRÜCÜ DÂHİL 9 KOLTUK)	0.577	0.577	0.577	0	0	0
ÇEKİCİ	0	0	0	1	0	0
OTOBÜS (SÜRÜCÜ DÂHİL 18-30 KOLTUK)	0	0.5	0.5	0	0.5	0
MİNİBÜS (SÜRÜCÜ DÂHİL 10-17 KOLTUK)	0	0.5	0.5	0	0	0.5

4.2.4 Normalizasyon

Modelleme sürecinde kullanılan sayısal veriler, farklı ölçeklerde ve büyüklüklerde olabileceğinden, öğrenme algoritmalarının daha etkili çalışabilmesi adına normalizasyon işlemi gerçekleştirilmiştir. Bu çalışmada, nümerik değerli özellikler üzerinde Z-score standardizasyonu olarak da bilinen Standart Skor Normalizasyonu (StandardScaler) tekniği kullanılmıştır.

Bu teknik ile her bir gözlem değeri, o özelliğin ortalamasından çıkarılıp standart sapmasına bölünerek yeniden ölçeklendirilmiştir. Aşağıda uygulanan formül ile gösterilmiştir:

$$X_{\text{new}} = \frac{X-\mu}{\sigma} \tag{4.1}$$

Formülde X özellik değerlerini, μ özellik değerlerinin ortalamasını ve σ standart sapmayı göstermektedir.

Bu yöntem sayesinde tüm sayısal veriler sıfır ortalama ve bir standart sapma olacak şekilde dönüştürülmüş, bu da algoritmanın özellikle mesafe-temelli hesaplamalar veya gradyan tabanlı optimizasyonlar yaparken daha verimli çalışmasına katkı sağlamıştır.

Veri setindeki normalizasyon işlemi, eğitim ve test veri setleri ayrı ayrı ele alınarak gerçekleştirilmiştir. Eğitim verisinde fit_transform, test verisinde ise aynı dönüşüm parametreleri kullanılarak transform işlemi uygulanmıştır. Bu sayede veri sızıntısı (data leakage) önlenmiş ve adil bir model değerlendirme ortamı oluşturulmuştur.

4.2.5 İşlenen Verinin Tanımı

Tüm veri ön işleme adımlarının sistematik ve özenli bir şekilde uygulanmasının ardından, başlangıçta 817.738 kayıt (satır) ve 155(kolon) özellikten oluşan veri seti,

204 özellik ve 817.738 kayıt içeren ve yapısal bütünlüğü sağlanmış bir formata dönüştürülmüştür. Veri ön işleme süreci kapsamında özellikler niteliklerine göre kategorik, sayısal ve analiz dışı bırakılan (drop) özellikler olarak sınıflandırılmıştır. Bu doğrultuda, 25 özellik veri analizine katkı sağlamadığı ya da teknik nedenlerle uygun bulunmadığı için veri setinden çıkarılmıştır. Kalan özellikler arasında 63 adet kategorik (cat_cols) ve 121 adet sayısal (num_cols) ve 3 adet tarihsel (date_cols) özellik yer almaktadır.

Tablo 4.6 Veri Setindeki Özellik Türleri ve Sayıları

Özellik Türü	**Özellik Sayısı**
Kategorik Özellikler	63
Sayısal Özellikler	121
Tarihsel Özellikler	3
Çıkarılan Özellikler	25

Özellik sayısındaki artış, öncelikle tarihsel veriler üzerinde yapılan kapsamlı dönüşümlerden kaynaklanmaktadır. Orijinal tarihsel sütunlardan yıl, ay, gün, hafta numarası, haftanın günü ve hafta sonu gibi bileşenler çıkarılarak yeni özellikler oluşturulmuş; ayrıca farklı tarih çiftleri arasındaki gün bazında farklar hesaplanarak zamansal özellikler türetilmiştir. Bu sayede tarihsel veriler, model için daha anlamlı ve ayrıntılı hale getirilmiştir. Veri setindeki 3 orijinal tarihsel özellik ise doğrudan analizde kullanılmayarak veri setinden çıkarılmıştır. Ayrıca, metinsel ve kategorik veriler üzerinde karakter düzeltmeleri, TF-IDF ile sayısal temsile dönüştürme ve Etiket Kodlama (Label Encoding) gibi dönüşüm teknikleri uygulanmış; bazı durumlarda bu işlemler sonucunda yeni özellik sütunları eklenmiştir. Yüksek oranda (%90) eksik veya anlamlı bilgi içermeyen özellikler ise veri setinden çıkarılmıştır. Her bir özelliğin türü, uygulanan kodlama veya normalizasyon türü (LE, TF-IDF, Standart Skor Normalizasyonu) ve hangi özelliklerle ilişkilendirilerek oluşturulduğu dikkatle analiz edilmiş ve detaylı açıklayıcı bilgiler derlenmiştir. Böylece, modelleme sürecine geçmeden önce veri seti hem istatistiksel olarak anlamlı hem de makine öğrenmesi algoritmalarıyla uyumlu, eksiksiz ve kaliteli bir yapıya kavuşmuştur.

4.2.5.1 Sayısal Özellikler

Veri seti üzerinde gerçekleştirilen ön işleme süreci sonucunda toplam **121 adet sayısal özellik** elde edilmiştir. Bu özelliklerin bir kısmı, tarihsel alanlardan türetilen zaman bileşenleri ile kategorik özelliklerden yapılan sayısal kodlamalardan oluşmaktadır. Ancak yalnızca doğrudan ham veriden gelen ve türetilmemiş olan 64 temel sayısal özellik, veri setinin özgün yapısını yansıtmaktadır.

Bu sayısal özellikler; araç teknik bilgileri, hasar kayıtları, onarım detayları ve mali değerler gibi farklı kategorilere yayılmış durumdadır. Bu bölümde, tüm özelliklerin detaylarına yer verilmemiş; bunun yerine, söz konusu 64 temel sayısal özellik arasından, farklı bilgi türlerini temsil edecek şekilde seçilmiş 12 örnek özelliğe ait tanımlayıcı istatistikler Tablo 4.7'de sunulmuştur.

Tablo 4.7 Sayısal Özelliklerin Betimsel İstatistikleri

Özellik Adı	Kayıt Sayısı	Ortalama	Standart Sapma	Min Değer	Max Değer
MODEL_YILI	817737	2010.072	7.126498	1946	2022
AGIRLIK	810166	1171.156	1555.136	0	9865
BEYGIR_GUCU	809742	73.31196	78.27038	0	8974
HASAR_KAZA_SAYI	710375	190.3813	1317.028	0	16767
HASAR_TPL_TUT	710375	2443147	12878629	0	793724360.2
HASAR_CAM_ORT_TUT	710375	239.991	847.0927	0	45377.25
HASAR_RED_SAYI	710375	2.283505	19.70536	0	3825
HASAR_SAYI	533324	1.176247	1.544639	0	7
HASAR_KAZA_MIN_TUT	483569	5925.019	58874.62	0	6652164
HASAR_KAZA_MAX_TUT	483569	248293.6	728560.9	0	13035989.87
PIYASA_DEGERI	472533	29197580	5.26E+09	0	1E+12
OTOA_KUSUR_ORANI	430935	84.65904	32.96306	0	127
OTOA_DTY_PARCA_TUTAR	429567	2336.612	9379.964	0	1922903.62

4.2.5.2 Kategorik Özellikler

Veri setinde toplam **64 adet kategorik** özellik bulunmaktadır. Bu özellikler, verinin sınıflandırılabilir niteliklerini temsil eder ve modelleme sürecinde önemli rol oynar. Kategorik özelliklerin dağılımı, en sık görülen değerler (mod), benzersiz (unique) değer sayısı ve üst sıklıkta olan kategorilerin oranları Tablo 4.8'de özetlenmiştir. Bu sayede, her bir kategorik özelliğin yapısı ve örnek dağılımı kolaylıkla incelenebilir.

Tablo 4.8 Kategorik Özelliklerin Betimsel İstatistikleri

Özellik Adı	Benzersiz Değer Sayısı	En Çok Tekrarlanan Değer	Örnek Dağılım
MUSTERI_TIPI	2	Özel	Özel: %58, Tüzel: %42
ACENTE_IL_ADI	82	İSTANBUL	İSTANBUL: %42, ANKARA: %12, İZMİR: %7
ACENTE_BOLGE_ADI	11	KADIKÖY BÖLGE MÜDÜRLÜĞÜ	KADIKÖY BÖLGE: %25, İÇ ANADOLU: %17
HASAR_SEBEBI	9	ARACIN ÇARPIŞMASI	ARACIN ÇARPIŞMASI: %94, İNSANA ÇARPMA: %4
TAM_HASAR_MI	2	HAYIR	HAYIR: %97, EVET: %3
OTOA_HASAR_SEKLI	14	Çarpışma	Çarpışma: %50, Park Halinde: %3
OTOA_SERVIS_TURU	3	OZEL	OZEL: %50, YETKILI: %3
OTOA_DOSYA_TIPI	3	EKSPERLI	EKSPERLI: %51, MODUL: %2
TRAMER_HASAR_SEBEBI	5	ÇARPMA-ÇARPISMA-DEVRILME	ÇARPMA: %61, CAM KIRILMASI: %1
YAKIT_TIPI	21	DIZEL	DIZEL: %48, BENZINLI: %14
KULLANIM_TIPI	21	OTOMOBİL (SÜRÜCÜ DÂHİL 9 KOLTUK)	OTOMOBİL: %44, KAMYONET: %20, KAMYON: %8
KATILIMCI_TURU	71	HAK SAHİBİ	HAK SAHİBİ: %14, ACENTE: %14, SİGORTALI: %14

4.3 Veri İçindeki Hedef Sınıf Dağılımı

Dengesiz bir veri seti, hedef özelliğe ait sınıflar arasındaki dengesizliğin bir sonucudur. Genel olarak, bir sınıfın diğer sınıfa göre çok daha fazla ya da az sayıda olması, modelin öğrenme sürecini olumsuz etkileyebilir. Bu durum, sınıflandırma problemlerinde modelin çoğunluk sınıfı öğrenmeye meyilli olmasına neden olurken, azınlık sınıfa ait örneklerin doğru tahmin edilme oranını düşürmektedir. Dengesiz veri setlerinde, hem eğitim hem de test verilerinde çoğunluk sınıfın baskın olması nedeniyle modelin genel başarı oranı yüksek görünse de, bu sonuç yanıltıcı olabilir. Çünkü bu başarı, sadece baskın sınıfın doğru tahmin edilmesinden kaynaklanmaktadır. Oysa her iki sınıfın da eşit oranda temsil edildiği dengeli bir veri setiyle elde edilen başarı, modelin genellenebilirliği açısından çok daha güvenilir kabul edilmektedir. Bu tez çalışmasında, sınıflar arasındaki dengesizliği

gidermek amacıyla örnekleme teknikleri uygulanmış ancak bu uygulamalar model üzerinde yanlılığa neden olmuştur. Bu nedenle, nihai modelleme sürecinde örnekleme teknikleri kullanılmamış ve verinin doğal dağılımı üzerinden ilerlenmiştir.

4.3.1 Dava Açma Hedefi

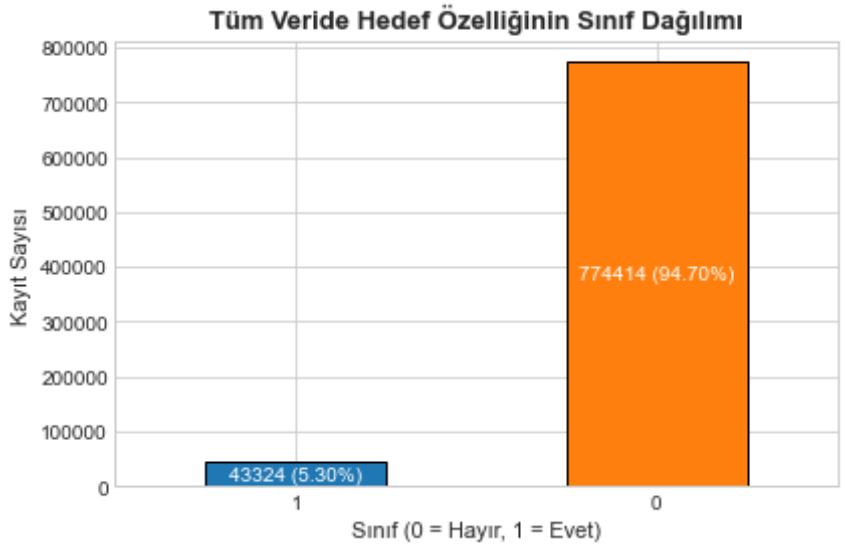
DAVA_ACMA_DOSYASI_ILISKISI özelliği, iki sınıfa sahiptir:

- 0 Sınıfı: Hukuk dosyası ile ilişkisi yoktur. (Dava açılmamış)
- 1 Sınıfı: Hukuk dosyası ile ilişkilidir. (Dava açılmış)

Tüm veri setindeki örnek sayıları ve oranları şu şekildedir:

- 0 sınıfı: 774.414 adet (%94,7)
- 1 sınıfı: 43.324 adet (%5,3)

Bu dağılım, Şekil 4.1'de görselleştirilmiştir ve hedef özellikte ciddi bir sınıf dengesizliği olduğunu ortaya koymaktadır.



Şekil 4.1 Dava Açma Hedefine Ait Sınıflar Arası Dağılım

5
VALİDASYON TEKNİKLERİ ANALİZİ

5.1 Kullanılan Python Kütüphaneleri

Bu tez çalışmasında, veri ön işleme, analiz, modelleme ve değerlendirme süreçlerinin tamamı Python programlama dili kullanılarak gerçekleştirilmiştir. Bu süreçte, Python'ın veri bilimi ve makine öğrenmesi alanında yaygın olarak kullanılan kütüphanelerinden yararlanılmıştır.

5.1.1 Pandas Kütüphanesi

Veri setlerinin yüklenmesi, birleştirilmesi, temizlenmesi ve genel veri manipülasyon işlemlerinde Pandas kütüphanesi aktif olarak kullanılmıştır. Özellikle zaman serisi verisinden tarihsel öznitelikler çıkaran time_features fonksiyonu, Pandas'ın datetime özellikleri yardımıyla geliştirilmiştir [45].

5.1.2 Numpy Kütüphanesi

Sayısal işlemler, istatistiksel analizler ve sinüzoidal dönüşümler gibi matematiksel hesaplamalarda Numpy kütüphanesinden yararlanılmıştır. Zaman serisi verileri için sinüs ve kosinüs tabanlı döngüsel özellikler bu kütüphane yardımıyla üretilmiştir [46].

5.1.3 Matplotlib Kütüphanesi

Görselleştirme amacıyla kullanılan Matplotlib kütüphanesi, özellikle korelasyon haritaları, eksik veri analizleri, sınıflandırma sonuçlarının karışıklık matrisleri gibi grafiklerin çizilmesinde kullanılmıştır. corr_map, plot_missing_values ve plot_confusion_matrix fonksiyonlarında etkin rol oynamıştır [47].

5.1.4 Seaborn Kütüphanesi

Veri görselleştirmede Matplotlib kütüphanesini tamamlayıcı bir yapıya sahip olan Seaborn, kategorik ve sayısal özellik analizlerinde kullanılmıştır. Cat_analyser ve num_analyser fonksiyonları, görsel analizlerin sade ve anlaşılır şekilde sunulmasını sağlamıştır [48].

5.1.5 Scikit-Learn (sklearn) Kütüphanesi

Modelleme sürecinin temel bileşenlerinden biri olan Scikit-learn, model eğitimi ve değerlendirmesinde aktif olarak kullanılmıştır. Model performansını ölçmede kullanılan ROC ve AUC gibi değerler, validasyon işlemleri, clf_trainer fonksiyonu kapsamında bu kütüphane üzerinden gerçekleştirilmiştir. Ayrıca BaseEstimator, BaseCrossValidator ve StratifiedKFold gibi fonksiyonlar kullanılarak esnek bir modelleme altyapısı kurulmuştur [7].

5.1.6 XGBoost, LightGBM ve CatBoost Kütüphaneleri

Gelişmiş Gradient Boosting algoritmalarını sağlayan bu üç kütüphane, sınıflandırma modellerinin eğitilmesinde kullanılmıştır. Her biri için model tahminlerini toplamak üzere get_model_test_preds, get_catboost_model_test_preds gibi fonksiyonlar geliştirilmiş ve bu modellerin kaydedilmesi/yüklenmesi için gerekli save_model, load_model, save_catboost_model ve load_catboost_model fonksiyonları tanımlanmıştır [49][50][51].

5.1.7 Joblib ve OS Kütüphaneleri

Eğitilen modellerin kalıcı olarak saklanması ve gerektiğinde tekrar kullanılabilmesi amacıyla Joblib kütüphanesinden yararlanılmıştır. Dosya sistemi ile ilgili işlemler ise OS kütüphanesi yardımıyla gerçekleştirilmiştir [52][53].

5.2 Kullanılan Sınıflandırma Algoritmaları

Tez çalışmasında, validasyon tekniklerinin sınıflandırma başarıları üzerindeki etkisini değerlendirmek amacıyla çeşitli makine öğrenmesi algoritmalarından yararlanılmıştır. Bu algoritmalar, clf_trainer fonksiyonu aracılığıyla sistematik biçimde eğitim ve test verileri üzerinde denenmiş, her kat için model performansları

hesaplanmış ve sonuçlar karşılaştırmalı olarak sunulmuştur. Bu kapsamda kullanılan sınıflandırma algoritmaları şunlardır:

- CatBoost
- XGB (XGBoost)
- LGB Classifier (LightGB)
- Logistic Regression (LogReg)
- Random Forest (RF)

5.3 Kullanılan Validasyon Teknikleri

Bu tez çalışmasında model performansını daha güvenilir ve genellenebilir şekilde ölçümlemek amacıyla çeşitli validasyon teknikleri kullanılmıştır. İlk aşamada temel bir yaklaşım olarak veri seti **%80 eğitim ve %20 test verisi** olacak şekilde rastgele ikiye ayrılmış ve bu işlem Hold-Out validasyonu gerçekleştirilmiştir.

Bununla birlikte, modelin farklı veri alt kümeleri üzerinde nasıl performans gösterdiğini daha detaylı inceleyebilmek adına çeşitli validasyon teknikleri uygulanmıştır. Kullanılan başlıca validasyon teknikleri aşağıdaki gibidir:

K-Katlı Çapraz Validasyon (K-Fold Cross Validation): Veri seti eşit büyüklükte 5 parçaya bölünerek her bir parça test verisi olarak kullanılmış, kalan kısımlar ise eğitim amacıyla değerlendirilmiştir. Süreç rastgelelik içerecek şekilde shuffle=True parametresiyle gerçekleştirilmiş ve tekrarlanabilirlik için random_state değeri sabitlenmiştir. Burada n_splits=5 değeri, veri setinin **5 parçaya bölünmesi ve 4 parçanın (%80) eğitim,** aynı şekilde 4 **katta verinin %80'lik kısmı eğitim için ve %20**'lik kısmı **test için** kullanılmıştır.

Katmanlı K-Katlı Çapraz Validasyon (Stratified K-Fold Cross Validation): Dengesiz veri yapısının sınıf dağılımı korunarak daha tutarlı bölünmeler sağlamak amacıyla, sınıflar arası denge gözetilerek 5 katlı katmanlı çapraz validasyon tekniği uygulanmıştır.

Gruplu K-Katlı (Group K-Fold): Aynı gruba ait gözlemlerin hem eğitim hem test setine aynı anda düşmesini engellemek amacıyla gruplara göre ayrıştırma yapan bu teknik, özellikle grup bağımlı veriler için tercih edilmiştir.

Katmanlı Gruplu K-Katlı (Stratified Group K-Fold): Veri setinde hem sınıf dağılımının korunması hem de grup bütünlüğünün bozulmaması hedeflenerek, sınıf etiketlerine göre katmanlandırma ve grup bazlı ayrıştırma birlikte uygulanmıştır. Bu teknik, grup temelli ve dengesiz veri setlerinde daha güvenilir ve temsili bölünmeler sağlamaktadır.

Tekrarlı K-Katlı ve Tekrarlı Katmanlı K-Katlı (Repeated K-Fold ve Repeated Stratified K-Fold): Modelin farklı veri bölünmeleri üzerindeki tutarlılığını ölçmek için 5 katlı çapraz validasyon işlemi belirli sayıda tekrarlanarak hem rastgeleliğin hem de istatistiksel güvenilirliğin sağlanması hedeflenmiştir.

Zaman Serisi Bölmesi (Time Series Split): Zaman serisi analizlerine özgü olan bu teknikte veri sırası korunarak ardışık şekilde 5 parçaya ayrılmış ve geçmiş verilere dayalı tahminleme yapılması sağlanmıştır.

Tüm validasyon tekniklerinde n_splits=5 parametresi alınmıştır. Böylece **işlenmiş verinin %80'ni eğitim, %20'ni test** amacıyla kullanılmıştır.

5.4 Kullanılan Performans Değerlendirme Metrikleri

Modelin performansını değerlendirebilmek adına uygun performans metriklerinin belirlenmesi büyük önem taşımaktadır. Bu çalışma kapsamında, Doğruluk (Accuracy), Duyarlılık (Recall), Kesinlik (Precision), F1-Skoru ve Gini katsayısı gibi metrikler tercih edilmiştir. Bu metrikler, dengesiz veri setlerinde modelin yalnızca çoğunluk sınıfını değil, azınlık sınıfını da ne ölçüde doğru tahmin ettiğini anlamada kritik bir rol oynamaktadır.

Duyarlılık (Recall), Kesinlik (Precision) ve F1-Skoru gibi metrikler, özellikle sınıf bazında performansı değerlendirmek için kullanılır ve her bir sınıf (örneğin 0 ve 1) için ayrı ayrı hesaplanabilir. Bu sayede modelin hem pozitif sınıfı (genellikle ilgi duyulan veya azınlık sınıfı) hem de negatif sınıf üzerindeki performansı detaylı olarak incelenebilir. Böylece, modelin azınlık sınıfı üzerindeki başarısı ya da başarısızlığı açıkça görülebilir.

Öte yandan, Doğruluk (Accuracy) değeri ve Gini katsayısı gibi metrikler ise tüm veri seti üzerinden genel performansı ölçer ve sınıf bazında ayrı ayrı değer verilmesi mümkün veya anlamlı değildir. Çünkü doğruluk, tüm doğru tahminlerin toplam

tahmin sayısına oranı olarak hesaplanır ve sınıf ayrımı gözetmeden genel başarıyı yansıtır. Gini katsayısı ise modelin ayırt edicilik gücünü tek bir skor olarak ifade eder ve sınıf bazında ayrı hesaplanmaz.

Bu sebeple, Doğruluk değeri ve Gini katsayısı için sınıf bazlı sonuçlar sunulmazken; Duyarlılık (Recall), Kesinlik (Precision) ve F1-Skoru için hem 0 hem de 1 sınıfı için ayrı ayrı değerler raporlanmaktadır. Böylece, modelin hem genel performansı hem de sınıflar arasındaki performans farklılıkları eksiksiz bir şekilde değerlendirilmiş olur.

5.5 Dava Açma Hedefi Tahmin Sonuçları

Bu bölümde, Dava Açma hedefinin sınıflandırılmasına yönelik olarak uygulanan farklı validasyon teknikleri ve beş farklı makine öğrenmesi algoritması (CatBoost, LightGBM, XGBoost, Random Forest ve LR) ile elde edilen model başarıları değerlendirilmektedir. Bu kapsamda Doğruluk (Accuracy), Duyarlılık (Recall), Kesinlik (Precision) ve F1-Skoru değerleri ile Gini katsayısı üzerinden ayrı ayrı analiz gerçekleştirilmiştir. Her bir metrik için oluşturulan tablolar, ilgili performans ölçütünün farklı validasyon stratejilerine göre algoritmalar üzerindeki etkisini karşılaştırmalı olarak sunmaktadır.

Aşağıda sunulan sonuçlar, yalnızca modelin **test** veri seti üzerindeki tahmin performansına dayanmaktadır. Eğitim (train) verisi yalnızca modelin öğrenme sürecinde kullanılmıştır.

5.5.1 Doğruluk (Accuracy) Sonuçları

Dava Açma hedefini tahmin etmek amacıyla kullanılan sınıflandırma modellerinin başarısı ilk olarak Doğruluk (Accuracy) metriği ile değerlendirilmiştir. Doğruluk, Gerçekte 1 olan sınıfın 1 ve 0 olan sınıfın 0 olarak tahminin doğru yapıldığı toplam kayıt sayısının tüm kayıtların sayısına oranını gösterir. Bu bağlamda farklı çapraz validasyon teknikleri kullanılarak elde edilen Doğruluk değerleri karşılaştırmalı olarak analiz edilmiştir.

Tablo 5.1, sınıflandırma algoritmalarının çeşitli validasyon tekniklerine göre elde ettiği Doğruluk (Accuracy) değerlerini sunmaktadır.

Tablo 5.1 Dava Açma Hedefinin Farklı Validasyon Tekniklerine göre Sınıflandırma Algoritmalarından Elde Edilen Doğruluk (Accuracy) Değerleri

Doğruluk (Accuracy)		Algoritmalar				
		CatBoost	LGBM	XGB	RF	LogReg
Validasyon Teknikleri	Hold-Out	0.87	0.82	0.87	0.76	**0.95**
	K-Fold	0.90	0.81	0.88	0.74	**0.95**
	Stratified K-Fold	0.89	0.82	0.87	0.78	**0.95**
	Group K-Fold	0.90	0.81	0.91	0.76	**0.95**
	Repeated K-Fold	0.89	0.82	0.86	0.78	**0.95**
	Repeated Stratified K-Fold	0.87	0.81	0.89	0.77	**0.95**
	Stratified Group K-Fold	0.88	0.84	0.86	0.79	**0.95**
	Time Series Split	**0.95**	0.86	0.90	0.72	**0.95**

- Tüm validasyon tekniklerinde Lojistik Regresyon algoritması ve sadece Time Series Split validasyonunda hem Lojistik Regresyon algoritması hem de CatBoost algoritması %95 ile en başarılı sonucu vermiştir.
- K-Fold, Lojistik Regresyon için yine %95 doğruluk üretmiş, CatBoost %90 ile ikinci sırayı almıştır.
- Group K-Fold tekniğinde XGBoost %91 doğrulukla kendi içinde en iyi sonucunu verirken, Lojistik Regresyon ve CatBoost algoritmaları da %95 ve %90 ile güçlü performans göstermiştir.

5.5.2 Duyarlılık (Recall / Sensitivity) Sonuçları

Tablo 5.2, çeşitli validasyon tekniklerine göre sınıflandırma algoritmalarından elde edilen Duyarlılık (Recall / Sensitivity) değerlerini içermektedir.

Tablo 5.2 Dava Açma Hedefinin Farklı Validasyon Tekniklerine göre Sınıflandırma Algoritmalarından Elde Edilen Duyarlılık (Recall / Sensitivity) Değerleri

Duyarlılık (Recall / Sensitivity)		Hedef Sınıfı	Algoritmalar				
			CatBoost	LGBM	XGB	RF	LogReg
Validasyon Teknikleri	Hold-Out	0	0.90	0.89	0.91	0.86	0.99
		1	**0.83**	0.82	0.82	0.80	0.13
	K-Fold	0	0.91	0.89	0.91	0.84	0.99
		1	0.81	0.82	0.82	0.79	0.13
	Stratified K-Fold	0	0.90	0.89	0.92	0.87	0.99
		1	0.81	**0.83**	0.81	0.79	0.13
	Group K-Fold	0	0.91	0.88	0.93	0.84	0.99
		1	0.76	0.80	0.73	0.77	0.11
	Repeated K-Fold	0	0.90	0.89	0.90	0.87	0.99
		1	0.82	0.82	0.82	0.79	0.12
	Repeated Stratified K-Fold	0	0.89	0.88	0.89	0.86	0.99
		1	**0.83**	**0.83**	0.81	0.79	0.13
	Stratified Group K-Fold	0	0.89	0.88	0.90	0.85	0.99
		1	0.76	0.75	0.75	0.72	0.11
	Time Series Split	0	0.97	0.94	0.95	0.90	1.00
		1	0.34	0.42	0.43	0.42	0.03

- Azınlık sınıfı (1) için en yüksek Recall değeri %83 olarak Hold-Out validasyonu (CatBoost), Stratified K-Fold validasyonu (LGBM) ve Repeated Stratified K-Fold validasyonu (CatBoost ve LGBM) elde edilmiştir.
- Çoğunluk sınıfı (0) için en yüksek Recall değeri %100 olarak Time Series Split validasyonundan (Lojistik Regresyon) elde edilmiş olup, Lojistik Regresyon algoritması tüm validasyon tekniklerinde %99 ile %100 arasında çok yüksek duyarlılık değerleri sergilemiştir. Ayrıca, Time Series Split tekniğinde CatBoost %97, XGBoost %95 ile yüksek Recall değerleri yakalamıştır.

5.5.3 Kesinlik (Precision) Sonuçları

Tablo 5.3, farklı validasyon tekniklerine göre sınıflandırma algoritmalarından elde edilen Kesinlik (Precision) değerlerini ayrıntılı olarak sunmaktadır.

Tablo 5.3 Dava Açma Hedefinin Farklı Validasyon Teknikleri ve Sınıflandırma Algoritmaları Kullanılarak Elde Edilen Kesinlik (Precision) Değerleri

Kesinlik (Precision)		Hedef Sınıfı	Algoritmalar				
			CatBoost	LGBM	XGB	RF	LogReg
Validasyon Teknikleri	Hold-Out	0	0.99	0.99	0.99	0.99	0.95
		1	0.33	0.30	0.35	0.25	**0.49**
	K-Fold	0	0.99	0.99	0.99	0.99	0.95
		1	0.35	0.30	0.35	0.22	0.48
	Stratified K-Fold	0	0.99	0.99	0.99	0.99	0.95
		1	0.33	0.31	0.36	0.25	0.48
	Group K-Fold	0	0.98	0.99	0.98	0.98	0.95
		1	0.32	0.27	0.37	0.22	0.47
	Repeated K-Fold	0	0.99	0.99	0.99	0.99	0.93
		1	0.33	0.30	0.34	0.25	**0.49**
	Repeated Stratified K-Fold	0	0.99	0.99	0.99	0.99	0.95
		1	0.31	0.29	0.35	0.25	**0.49**
	Stratified Group K-Fold	0	0.98	0.98	0.98	0.98	0.95
		1	0.29	0.27	0.31	0.22	0.43
	Time Series Split	0	0.96	0.97	0.97	0.96	0.95
		1	0.42	0.30	0.32	0.19	0.29

- Azınlık sınıfı (1) için en yüksek Precision değeri %49 ile Hold-Out, Repeated K-Fold ve Repeated Stratified K-Fold validasyon tekniklerinden Lojistik Regresyon algoritmasıyla elde edilmiştir. Onu %48 ile K-Fold (LogReg) ve Stratified K-Fold (LogReg) izlemiştir. Time Series Split validasyonu azınlık sınıfı için CatBoost algoritmasında Precision değeri %42 ile belirgin bir iyileşme sağlasa da, genel olarak diğer validasyon tekniklerine kıyasla azınlık sınıfında daha düşük Precision değerleri göstermiştir Series Split tekniğinde, CatBoost %42 ile kendi içinde en yüksek azınlık sınıfı Precision değerine ulaşarak belirgin bir iyileşme göstermiştir.

- Çoğunluk sınıfı (0) için tüm validasyon tekniklerinde çoğunluk sınıfı için en yüksek Precision değeri genellikle %98–99 arasında değişmiş, bu da modellerin 0 sınıfını ayırt etmede oldukça başarılı olduğunu göstermiştir. Hold-Out, K-Fold, Stratified K-Fold, Repeated K-Fold ve Repeated

Stratified K-Fold tekniklerinde CatBoost, LGBM, XGBoost ve Random Forest algoritmaları %99 Precision değeri ile öne çıkmıştır.

5.5.4 F1-Skoru Sonuçları

F1-Skoru, özellikle dengesiz veri setlerinde modelin genel başarısını değerlendirmek için önemli bir metriktir. Bu metrik, Duyarlılık (Recall) ve Kesinlik (Precision) arasındaki harmonik ortalamayı temsil eder. Bu bölümde, validasyon tekniklerine göre sınıflama algoritmalarının F1-Skorları analiz edilmiştir. Tablo 5.4, uygulanan farklı validasyon teknikleri kapsamında algoritmaların F1-Skoru değerlerini göstermektedir.

Tablo 5.4 Dava Açma Hedefinin Farklı Validasyon Teknikleri göre Sınıflandırma Algoritmalarından Elde Edilen F1-Skoru Değerleri

F1-Skoru (F1-Score)		Hedef Sınıfı	Algoritmalar				
			CatBoost	LGBM	XGB	RF	LogReg
Validasyon Teknikleri	Hold-Out	0	0.94	0.94	0.95	0.92	0.97
		1	0.47	0.44	0.49	0.38	0.20
	K-Fold	0	0.95	0.93	0.95	0.91	0.97
		1	0.49	0.44	0.49	0.35	0.20
	Stratified K-Fold	0	0.94	0.94	0.95	0.92	0.97
		1	0.47	0.45	**0.50**	0.38	0.20
	Group K-Fold	0	0.94	0.93	0.95	0.91	0.97
		1	0.45	0.41	0.49	0.35	0.18
	Repeated K-Fold	0	0.94	0.94	0.94	0.92	0.96
		1	0.47	0.44	0.47	0.38	0.19
	Repeated Stratified K-Fold	0	0.94	0.93	0.94	0.92	0.97
		1	0.45	0.43	0.49	0.38	0.20
	Stratified Group K-Fold	0	0.93	0.93	0.94	0.91	0.97
		1	0.42	0.39	0.43	0.35	0.19
	Time Series Split	0	0.97	0.95	0.96	0.93	0.97
		1	0.38	0.35	0.36	0.26	0.05

- Azınlık sınıfı (1) için en yüksek F1-Skoru değeri %50 ile Stratified K-Fold validasyon tekniğinden XGBoost algoritması tarafından elde edilmiştir. Bunu %49 ile Hold-Out ve K-Fold validasyonunda XGBoost ile Repeated Stratified K-Fold ve Hold-Out validasyonunda CatBoost takip etmektedir.
- Çoğunluk sınıfı (0) için tüm validasyon tekniklerinde Lojistik Regresyon algoritması ve sadece Time Series Split validasyonunda hem Lojistik

Regresyon algoritması hem de CatBoost algoritması %97 ile en başarılı sonucu vermiştir.

- Time Series Split validasyon tekniği azınlık sınıfı (1) için genel olarak en düşük F1-Skoru değerlerini verirken (CatBoost için %38, XGBoost için %36), çoğunluk sınıfında ise en yüksek F1-Skoruna %97 ile ulaşmıştır.

5.5.5 Gini Katsayısı Sonuçları

Model performansını ölçmek için kullanılan bir diğer önemli gösterge, AUC (Area Under Curve) değeridir. Bu metrik, ROC eğrisi altında kalan alanı ifade eder ve modelin pozitif ve negatif sınıfları ayırt etme becerisini yansıtır. AUC'den türetilen Gini katsayısı ise AUC değerinin doğrusal bir dönüşümüdür ve benzer amaçla kullanılmaktadır.Tablo 5.5, validasyon tekniklerine göre sınıflandırma algoritmalarının elde edilen Gini değerlerini göstermektedir.

Tablo 5.5 Dava Açma Hedefinin Farklı Validasyon Tekniklerine göre Sınıflandırma Algoritmalarından Elde Edilen Gini Katsayısı Değerleri

Gini		Algoritmalar				
		CatBoost	LGBM	XGB	RF	LogReg
Validasyon Teknikleri	Hold-Out	0.70	0.66	0.70	0.6	0.4
	K-Fold	0.70	0.66	**0.72**	0.56	0.40
	Stratified K-Fold	0.70	0.66	0.68	0.60	0.40
	Group K-Fold	0.70	0.64	0.70	0.58	0.40
	Repeated K-Fold	**0.72**	0.66	0.70	0.60	0.40
	Repeated Stratified K-Fold	**0.72**	0.66	0.70	0.60	0.40
	Stratified Group K-Fold	**0.72**	0.64	0.70	0.56	0.40
	Time Series Split	0.48	0.50	0.56	0.34	0.34

- En yüksek Gini katsayısı %72 ile Repeated K-Fold, Repeated Stratified K-Fold ve Stratified Group K-Fold validasyon tekniklerinde CatBoost ile K-Fold validasyonunun XGBoost algoritmasından elde edilmiştir.
- Hold-Out, Group K-Fold, Repeated K-Fold, Repeated Stratified K-Fold ve Stratified Group K-Fold validasyon tekniklerinde XGBoost; Hold-Out, K-Fold ve Stratified K-Fold validasyonlarında ise CatBoost algoritmaları %70 ile en iyi ikinci Gini katsayısı değerleri elde edilmiştir.
- Time Series Split validasyonunda Gini katsayı değerleri düşük seviyelerde kalmış; CatBoost %48, XGBoost %56 ve LGBM %50 ile diğer validasyon tekniklerine kıyasla daha sınırlı bir performans sergilemiştir. Bu teknik, özellikle Random Forest (%34) ve Lojistik Regresyon (%34) için ayrım gücü bakımından en zayıf sonucu vermiştir.

6
SONUÇ

6.1 Genel Değerlendirmeler

Bu çalışma kapsamında farklı validasyon tekniklerinin sınıflandırma algoritmalarının performans metriklerine etkisi detaylı biçimde incelenmiştir. Beş farklı sınıflandırma algoritması (CatBoost, LGBM, XGB, RF, LogReg) ve sekiz farklı validasyon tekniği ile toplamda kırk senaryo oluşturularak analizler gerçekleştirilmiştir. Elde edilen sonuçlar, Doğruluk (Accuracy), Duyarlılık (Recall), Kesinlik (Precision), F1-Skoru ve Gini Katsayısı gibi çeşitli metrikler üzerinden değerlendirilmiştir.

Doğruluk (Accuracy) açısından, Lojistik Regresyon algoritması tüm validasyon tekniklerinde %95 doğruluk oranı ile dikkat çekerken, Time Series Split validasyonu hem Lojistik Regresyon hem de CatBoost algoritmaları ile %95 doğruluk sağlayarak genel doğrulukta en yüksek başarıyı sunmuştur.

Duyarlılık (Recall) açısından, azınlık sınıfı (1) için en yüksek değer %83 ile Hold-Out (CatBoost), Stratified K-Fold (LGBM) ve Repeated Stratified K-Fold (CatBoost, LGBM) validasyonlarıyla elde edilmiştir. Çoğunluk sınıfı (0) için ise Time Series Split validasyon tekniği %100'e varan değerlerle Lojistik Regresyon algoritmasında en yüksek duyarlılığı sağlamıştır. Ayrıca, bu teknikte CatBoost %97, XGBoost %95 oranları ile güçlü performans sergilemiştir.

Kesinlik (Precision) açısından, azınlık sınıfı (1) için en yüksek oran %49 ile Hold-Out, Repeated K-Fold ve Repeated Stratified K-Fold validasyonlarında Lojistik Regresyon algoritmasından elde edilmiştir. Çoğunluk sınıfı (0) için ise birçok algoritma ve validasyon tekniği %98–99 oranlarında yüksek doğruluk göstermiştir.

CatBoost, LGBM, XGBoost ve RF algoritmaları özellikle Hold-Out, K-Fold ve türevlerinde öne çıkmıştır.

F1-Skoru açısından, azınlık sınıfı (1) için en iyi sonuç %50 ile Stratified K-Fold (XGBoost) validasyon tekniğinden elde edilirken, onu %49 ile XGBoost (Hold-Out, K-Fold) ve CatBoost (Repeated Stratified K-Fold, Hold-Out) takip etmiştir. Çoğunluk sınıfı (0) için ise Lojistik Regresyon algoritması neredeyse tüm validasyonlarda %97 gibi çok yüksek skorlar elde etmiştir. Time Series Split, hem Lojistik Regresyon hem de CatBoost (%97) ve XGBoost (%96) ile bu sınıf için en yüksek F1-Skoru'nu üretmiştir. Ancak bu teknik, azınlık sınıfı için F1-Skorlarını önemli ölçüde düşürmüştür.

Gini katsayısı açısından, en yüksek ayrım gücü %72 ile Repeated K-Fold, Repeated Stratified K-Fold ve Stratified Group K-Fold validasyon tekniklerinde CatBoost ve K-Fold validasyonunda XGBoost algoritmasıyla elde edilmiştir. %70 ile ikinci sırada gelen sonuçlar ise yine XGBoost ve CatBoost algoritmalarından türeyen çeşitli validasyon teknikleriyle elde edilmiştir. Time Series Split, bu metrikte genel olarak düşük skorlar üretmiş ve ayrım gücü bakımından en zayıf teknik olmuştur. Tüm bu değerlendirmeler ışığında, özellikle dengesiz veri yapılarında F1-Skoru ve Duyarlılık gibi metrikler öne çıkmaktadır. Bu metriklere göre en iyi sonuç aldığımız validasyon teknikleri sıralamasız şekilde Repeated K-Fold, Repeated Stratified K-Fold ve Stratified Group K-Fold olmuştur. Bu teknikler, özellikle CatBoost ve XGBoost algoritmalarıyla birlikte kullanıldığında yüksek Gini katsayısı, F1-Skoru ve Duyarlılık değerleri sunmuştur. Öte yandan, Time Series Split validasyonu yalnızca çoğunluk sınıfında güçlü sonuçlar üretmiş, azınlık sınıfı performansında ise belirgin düşüşlerle sınırlı kalmıştır.

6.2 Sonuçlar ve Öneriler

Tüm performans metrikleri göz önünde bulundurulduğunda, **Repeated Stratified K-Fold** validasyon tekniği azınlık ve çoğunluk sınıfları için dengeli ve yüksek başarı sağladığı için en uygun teknik olarak öne çıkmaktadır.

Azınlık sınıfı özelinde bu başarının detayları şöyledir:

- **Recall:** %83 ile LGBM ve CatBoost algoritmalarında en yüksek azınlık sınıfı duyarlılığı sağlanmıştır. (Birincilik)
- **Precision:** Lojistik Regresyon algoritması %49 kesinlik oranı ile sınırlı veri yapısında yüksek bir performans göstermiştir. (Birincilik)
- **F1-Skoru:** CatBoost %49 ile en iyi ikinci sonucu almış, XGBoost da rekabetçi skorlarıyla ikinci sırayı paylaşmıştır. Böylece hem hatalı pozitif hem de hatalı negatiflerin dengeli minimize edilmesi sağlanmıştır. (İkincilik)
- **Gini Katsayısı (Ayrım Gücü):** CatBoost algoritması ile %72'ye kadar çıkan Gini değerleri, modelin sınıflar arasında güçlü ve güvenilir bir ayrım yapabildiğini göstererek birinci sırada yer almaktadır. (Birincilik)

Bu veriler, Repeated Stratified K-Fold validasyonunun azınlık sınıfında hem algılama başarısını hem de pozitif tahmin isabetini artırmada en etkili teknik olduğunu göstermektedir.

Öncelikle, Repeated Stratified K-Fold ile sınıf dağılımının veri setinde korunması sayesinde her katmanda azınlık ve çoğunluk sınıflarının dengeli şekilde temsil edilmesi sağlanmakta, bu da özellikle azınlık sınıfının doğru değerlendirilmesi açısından önemli bir avantaj oluşturmaktadır. Böylece, modelin azınlık sınıfını algılama başarısı yükselirken, aşırı çoğunluk sınıfına odaklanma riski azaltılmaktadır.

Tekrarlamalı yapı ise, veri örneklemesindeki rastgelelikten kaynaklanan varyasyonu azaltarak validasyon sonuçlarının kararlılığını ve güvenilirliğini yükseltir. Bu özellik, modelin farklı veri alt kümelerinde tutarlı performans göstermesine olanak tanır ve genel genelleme yeteneğini destekler.

Öte yandan, zaman serisi verisi yapısına uygun olduğu düşünülen Time Series Split validasyonunda çoğunluk sınıfında oldukça başarılı sonuçlar elde edilmesine rağmen, azınlık sınıfında düşük performans ve düşük Gini katsayısı nedeniyle genel model güvenilirliği ve dengesi olumsuz etkilenmiştir. Bu durum, sınıf dengesizliğinin yüksek olduğu problemlerde sadece çoğunluk sınıfını ön plana çıkaran validasyonların yetersiz kalabileceğini göstermektedir.

Sonuç olarak, bu tez kapsamında Repeated Stratified K-Fold validasyon tekniği, sınıf dengesizliği probleminin doğası ve çeşitli performans ölçütleri göz önünde bulundurulduğunda, en sağlıklı ve güvenilir değerlendirme yöntemi olarak önerilmektedir. Bu teknik, hem model performansını objektif olarak yansıtır hem de farklı sınıfların ayrımında dengeli ve yüksek başarı sağlaması sebebiyle uygulama alanlarında tercih edilmelidir.

Özetle, bu çalışma göstermektedir ki:

- Validasyon tekniklerinin seçimi, sınıflandırma algoritmalarının performansını ve model güvenilirliğini doğrudan etkileyen kritik bir faktördür.
- Sınıf dengesizliği içeren veri setlerinde, yalnızca Doğruluk (Accuracy) oranına dayalı değerlendirmeler yanıltıcı olabilir; bu nedenle F1-Skoru, Duyarlılık (Recall) ve Gini katsayısı gibi dengeli ve açıklayıcı metrikler öncelikli olarak kullanılmalıdır.
- Repeated Stratified K-Fold validasyon tekniği, hem azınlık hem de çoğunluk sınıflarında dengeli ve yüksek performans sağlayarak, en güvenilir ve sağlıklı değerlendirme yöntemi olarak öne çıkmaktadır. Bu teknik, model performansını objektif şekilde yansıtmakta ve farklı sınıfların ayrımında tutarlı başarı sağlamaktadır.
- Yalnızca yüksek doğruluk oranına sahip olmak, hatalı sınıf tahminlerini önlemeyebilir; bu da kritik karar destek sistemlerinde risk oluşturabilir. Dolayısıyla, özellikle dengesiz veri problemlerinde doğru validasyon tekniği kullanımı model güvenilirliği ve uygulanabilirliği için zorunludur.

Tüm çalışmalara ek olarak bu tezde sekiz farklı validasyon tekniği üzerinde çalışma yapılmıştır. Bu bağlamda, gelecekteki çalışmaların farklı veri yapıları ve sektör özelinde validasyon tekniklerinin etkisini derinlemesine incelemesi, hem akademik bilgi birikimine katkı sağlayacak hem de uygulamalı model geliştirme süreçlerinde daha isabetli stratejik kararlar alınmasını mümkün kılacaktır.

KAYNAKÇA

[1] Kohavi, R., "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. 14th Int. Joint Conf. Artificial Intelligence (IJCAI)*, vol. 2, 1995, pp. 1137–1143.

[2] Varma, S. and Simon, R., "Bias in error estimation when using cross-validation for model selection," *BMC Bioinformatics*, vol. 7, p. 91, 2006.

[3] Japkowicz, N. and Stephen, S., "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[4] Tukey, J. W., *Exploratory Data Analysis*. Reading, MA: Addison-Wesley, 1977, pp. 1–6, 18–25.

[5] Geisser, S., "The predictive sample reuse method with applications," *J. Amer. Stat. Assoc.*, vol. 70, no. 350, pp. 320–328, 1975. [Online]. Available: https://doi.org/10.2307/2285815

[6] Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B., "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 24, 2011. [Online]. Available: https://papers.nips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html

[7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://jmlr.org/papers/v12/pedregosa11a.html

[8] Varoquaux, G., "Cross-validation failure: Small sample sizes lead to large error bars," *NeuroImage*, vol. 180, pp. 68–77, 2018. [Online]. Available: https://doi.org/10.1016/j.neuroimage.2017.06.061

[9] Şahin, A., Ayvaz, S., and Çalımfidan, A., "Detection of fraudulent claims in automotive insurance using machine learning techniques," in *Proc. Int. Conf. Artif. Intell. Data Process.*, vol. 1, pp. 110–120, 2020.

[10] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY: Springer, 2009, pp. 222, 230.

[11] Bishop, C. M., *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006, pp. 32, 45.

[12] Arlot, S. and Celisse, A., "A survey of cross-validation procedures for model selection," *Stat. Surv.*, vol. 4, pp. 40–79, 2010.

[13] Sebastiani, F., "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.

[14] Stone, M., "Cross-validatory choice and assessment of statistical predictions," *J. R. Stat. Soc. Series B (Methodological)*, vol. 36, no. 2, pp. 111–147, 1974.

[15] James, G., Witten, D., Hastie, T., and Tibshirani, R., *An Introduction to Statistical Learning: With Applications in R*. New York, NY: Springer, 2013.

[16] Lemaitre, G., Nogueira, F., and Aridas, C. K., "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 559–563, 2017.

[17] Fernández, A., García, S., Herrera, F., and Chawla, N. V., "SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *J. Artif. Intell. Res.*, vol. 61, pp. 863–905, 2018.

[18] Bengio, Y. and Grandvalet, Y., "No unbiased estimator of the leave-one-out error," *Neural Netw.*, vol. 17, no. 5–6, pp. 1125–1133, 2004.

[19] Géron, A., *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed., O'Reilly Media, 2019.

[20] Scikit-learn Documentation, "RepeatedKFold." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RepeatedKFold.html

[21] Kuhn, M. and Johnson, K., *Applied Predictive Modeling*. New York, NY: Springer, 2013.

[22] Scikit-learn Documentation, "StratifiedGroupKFold." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedGroupKFold.html

[23] Raschka, S., "Model evaluation, model selection, and algorithm selection in machine learning," *arXiv preprint* arXiv:1811.12808, 2018.

[24] Hyndman, R. J. and Athanasopoulos, G., *Forecasting: Principles and Practice*, 2nd ed., OTexts, 2018.

[25] Islam, S. and Amin, S. H., "Prediction of probable backorder scenarios in the supply chain using distributed random forest and gradient boosting machine learning techniques," *J. Big Data*, vol. 7, no. 1, pp. 1–22, 2020.

[26] Kulkarni, A. D. and Lowe, B., "Random forest algorithm for land cover classification," unpublished.

[27] Dhaliwal, N., "Leadership in AI-driven data science: Fostering innovation and collaboration for advancing healthcare [Figure 4]," *ResearchGate*, 2024. [Online]. Available: https://www.researchgate.net/publication/381002112

[28] Friedman, J. H., "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[29] Ye, L., Jabbar, S. F., Abdul Zahra, M. M., and Tan, M. L., "Bayesian regularized neural network model development for predicting daily rainfall from sea level pressure data: Investigation on solving complex hydrology

problem," *Complexity*, vol. 2021, Article ID 6685311. [Online]. Available: https://doi.org/10.1155/2021/6685311

[30] Ye, L., Jabbar, S. F., Abdul Zahra, M. M., and Tan, M. L., "Bayesian regularized neural network model development for predicting daily rainfall from sea level pressure data: Investigation on solving complex hydrology problem [Figure 5]," *Comput. Intell. Neurosci.*, vol. 2023, Article ID 4243162.

[31] Chen, T. and Guestrin, C., "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 785–794.

[32] Dua, R., Wallace, G. R., Chotso, T., and Densil Raj, V. F., "Classifying pulmonary embolism cases in chest CT scans using VGG16 and XGBoost [Figure X]," in J. S. Pan, P. N. Mahalle, and P. M. Raj (Eds.), *Intell. Commun. Technol. Virtual Mobile Netw.*, pp. 273–292, 2022.

[33] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., and Ma, W., "LightGBM: A highly efficient gradient boosting decision tree," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, pp. 3146–3154, 2017.

[34] Qian, Y., Zhang, M., and Liu, T., "Optimization of gradient boosting machines: A review," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 1–32, 2019.

[35] An, Z., Jiang, K., and Zheng, J. R., "Features of realized volatility analysis and return predicting based on LGBM and RNN model [Figure 3]," *Appl. Comput. Eng.*, vol. 27, no. 1, pp. 38–48, 2023.

[36] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A., "CatBoost: Unbiased boosting with categorical features," in *Adv. Neural Inf. Process. Syst.*, vol. 31, pp. 6638–6648, 2018.

[37] Islam, M. M., Das, P., Rahman, M. M., Naz, F., Kashem, A., Nishat, M. H., and Tabassum, N., "Prediction of compressive strength of high-performance concrete using optimization machine learning approaches with SHAP analysis," *SN Appl. Sci.*, vol. 6, no. 6, p. 543, 2024.

[38] Islam, M. M., Das, P., Rahman, M. M., and Naz, F., "Prediction of compressive strength of high-performance concrete using optimization machine learning approaches with SHAP analysis [Figure 4]," *J. Build. Pathol. Rehabil.*, vol. 9, no. 2, 2024.

[39] Levy, J. J. and O'Malley, A. J., "Don't dismiss logistic regression: The case for sensible extraction of interactions in the era of machine learning," *BMC Med. Res. Methodol.*, vol. 20, no. 1, pp. 1–15, 2020.

[40] Westreich, D., Lessler, J., and Funk, M. J., "Propensity score estimation: Neural networks, support vector machines, decision trees (CART), and meta-classifiers as alternatives to logistic regression," *J. Clin. Epidemiol.*, vol. 63, no. 8, pp. 826–833, 2010.

[41] Zeng, G., "On the confusion matrix in credit scoring and its analytical properties," *Commun. Stat. Theory Methods*, vol. 49, no. 9, pp. 2080–2093, 2020.

[42] Metz, C. E., “Basic principles of ROC analysis,” *Semin. Nucl. Med.*, vol. 8, pp. 283–298, 1978.

[43] Faraggi, D. and Reiser, B., “Estimation of the area under the ROC curve,” *Stat. Med.*, vol. 21, pp. 3093–3106, 2002.

[44] Hand, D. J. and Till, R. J., “A simple generalisation of the area under the ROC curve for multiple class classification problems,” *Mach. Learn.*, vol. 45, no. 2, pp. 171–186, 2001.

[45] McKinney, W., *Python for Data Analysis*, 2nd ed., O’Reilly Media, 2018, pp. 57–91.

[46] Harris, C. R., Millman, K. J., van der Walt, S. J., et al., “Array programming with NumPy,” *Nature*, vol. 585, pp. 357–362, 2020.

[47] Hunter, J. D., “Matplotlib: A 2D graphics environment,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007.

[48] Waskom, M. L., “Seaborn: Statistical data visualization,” *J. Open Source Softw.*, vol. 6, no. 60, p. 3021, 2021.

[49] Chen, T. and Guestrin, C., “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2016, pp. 785–794.

[50] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., et al., “LightGBM: A highly efficient gradient boosting decision tree,” in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, pp. 3146–3154, 2017.

[51] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A., “CatBoost: Unbiased boosting with categorical features,” in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 31, pp. 6638–6648, 2018.

[52] Joblib Developers, *Joblib documentation*, 2024. [Online]. Available: https://joblib.readthedocs.io

[53] Python Software Foundation, “os — Miscellaneous operating system interfaces,” 2024. [Online]. Available: https://docs.python.org/3/library/os.html

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirileri

1. Şaylı, A. and Temel, G., "Validation techniques in machine learning," in *Proc. 5th Int. Conf. Contemp. Acad. Res. (ICCAR)*, Konya, Türkiye, May 30–31, 2025, p. 54.