



TÜRKİYE CUMHURİYETİ
SAMSUN ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ
MYAZ468 BÜYÜK VERİ

“Büyük Veri ve Makine Öğrenmesi ile Deprem Sonrası Artçı Şok Tahmini ve Gerçek Zamanlı Takip Sistemi”

Hazırlayan
Serhat GENÇ
211118042

Dersin Öğretim Üyesi
Doç. Dr. Özgür TONKAL

SAMSUN
2025

İÇİNDEKİLER

İÇİNDEKİLER.....	1
1. PROJENİN TANIMI.....	3
1.1. Projenin Amacı:.....	3
1.2. Projenin Hedefleri:.....	3
2. GİRİŞ.....	4
3. VERİ TOPLAMA VE ÖN İŞLEME.....	5
3.1. Model Eğitimi İçin Tarihsel Veri Seti: USGS Hazard Program ve Türkiye Odaklı Filtreleme.....	5
3.2. Gerçek Zamanlı Veri Akışı.....	5
3.3. Veri Ön İşleme ve Temizleme.....	6
4. KEŞİFÇİ VERİ ANALİZİ VE YÖNTEMLER.....	8
4.1. Keşifçi Veri Analizi (EDA) ve Temel Bulgular.....	8
Görsel 1 Deprem Haritası.....	8
Görsel 2 Deprem Büyüklüğü Dağılımı.....	9
Görsel 3 17 Ağustos 1999 Gölcük Depremi ve 6 Şubat 2023 Kahramanmaraş Depremi Artçıları.....	10
Görsel 4a 6 Şubat 2023 Kahramanmaraş Depremi İlk 24 Saat Artçı Dağılım Grafiği.....	10
Görsel 4b 6 Şubat 2023 Kahramanmaraş Depremi Zamana Göre Ortalama Artçı Büyüklüğü.....	11
4.2. Analiz Sonuçlarının Yönlendirdiği Yöntem ve Teknolojiler.....	11
5. MODEL EĞİTİMİ VE PERFORMANS DEĞERLENDİRMESİ.....	12
5.1. Modelin Eğitilmesi.....	12
5.2. Değerlendirme Metrikleri.....	12
5.3. Performans Sonuçları ve Yorumlanması.....	13
6. SİSTEM MİMARİSİ VE UYGULAMA.....	15
6.1. API Endpoint'leri.....	15
6.2. Gerçek Zamanlı Veri Akışı ve Artçı Tahmini.....	15
6.3. Mobil Bildirim Sistemi (Firebase Entegrasyonu).....	16
6.4. Model Entegrasyonu.....	16
7. REACT NATİVE İLE GELİŞTİRİLEN MOBİL UYGULAMA.....	17
7.1. Uygulamanın Amacı ve Genel Yapısı.....	17
7.2. Temel Ekranlar ve Fonksiyonlar.....	17
Görsel 5 Uygulama Ana Ekranı ve Son Depremler Listesi.....	17
Görsel 7 Deprem Detay Modalı.....	18
7.3. Teknik Altyapı ve Servis Entegrasyonları.....	19
8. SONUÇ VE ÖNERİLER.....	20
8.1. Sonuç.....	20
8.2. Öneriler.....	20
9. KATKILAR VE SONRAKİ ÇALIŞMA ÖNERİLERİ.....	21

9.1. Katkılar.....	21
9.2. Sonraki Çalışma Önerileri.....	21
10. KAYNAKÇA.....	22

1. PROJENİN TANIMI

Proje Adı: Büyük Veri ve Makine Öğrenmesi ile Deprem Sonrası Artçı Şok Tahmini ve Gerçek Zamanlı Takip Sistemi

Bu projede, tarihsel deprem verileri kullanılarak eğitilmiş makine öğrenmesi modelleri aracılığıyla, gerçekleşen büyük bir depremin ardından meydana gelebilecek potansiyel artçı şokların büyüklüğünü ve zamanını tahmin eden bir sistem geliştirilmiştir. Geliştirilen bu sistem, gerçek zamanlı deprem verilerini bir web API'si üzerinden sürekli takip ederek, belirli bir büyüklüğün üzerindeki depremler için otomatik olarak artçı tahmini yapar. Sonuçlar, bir web arayüzü üzerinden canlı olarak sunulmakta ve mobil kullanıcılara anlık bildirimler (push notifications) ile iletilmektedir

1.1. Projenin Amacı:

Bu projenin temel amacı, modern veri bilimi ve yazılım geliştirme tekniklerini kullanarak deprem sonrası risk yönetimine katkı sağlamaktır. Ana şok sonrası oluşabilecek artçıların potansiyel büyüklüğü hakkında erken bir tahmin sunarak, arama-kurtarma ekiplerinin ve halkın daha bilinçli ve tedbirli hareket etmesine olanak tanıyacak bir teknolojik altyapı oluşturulması hedeflenmektedir. Proje, makine öğrenmesi modellemesi ile gerçek zamanlı veri işleme teknolojilerini bir araya getiren entegre bir çözüm sunmayı amaçlamaktadır.

1.2. Projenin Hedefleri:

- Geniş bir tarihsel deprem veri setini kullanarak keşifçi veri analizi yapmak ve deprem davranışlarındaki örüntüleri ortaya çıkarmak.
- Ana şokun büyüklük, derinlik ve konum bilgilerini kullanarak, ilk büyük artçı şokun büyüklüğünü ve ne kadar süre sonra gerçekleşeceğini tahmin eden makine öğrenmesi modelleri (LightGBM) eğitmek.
- Eğitilen modelleri sunmak üzere bir Flask tabanlı RESTful API geliştirmek. Bu API, anlık tahmin taleplerine yanıt verebilmelidir.
- Kandilli Rasathanesi gibi kaynaklardan gerçek zamanlı deprem verilerini periyodik olarak çeken bir arka plan servisi oluşturmak.
- WebSockets (Flask-SocketIO) kullanarak, yeni deprem verilerini ve artçı tahminlerini istemcilere (web tarayıcıları) anlık olarak ileten bir canlı veri akışı sağlamak.
- Firebase Cloud Messaging (FCM) entegrasyonu ile belirli bir büyüklüğü aşan depremler ve üretilen artçı tahminleri hakkında kayıtlı mobil cihazlara anlık bildirimler göndermek.
- Projenin tüm bileşenlerini (veri işleme, model, API, bildirim servisi) entegre ve çalışır bir bütün haline getirmek.

2. GİRİŞ

Depremler, dünya genelinde büyük can ve mal kayıplarına neden olan en yıkıcı doğal afetlerden biridir. Ana şokun ne zaman gerçekleşeceğini kesin olarak tahmin etmek günümüz teknolojisiyle mümkün olmasa da, büyük bir depremin ardından gelen ve "artçı şok" olarak adlandırılan daha küçük sarsıntıların davranışları istatistiksel ve makine öğrenmesi yöntemleriyle modellenenilmektedir. Omori Yasası gibi sismolojik kanunlar, artçı sayısının zamanla azaldığını belirtirken, bu artçıların büyüklüğü ve tam zamanlaması karmaşık bir problem olarak kalmaktadır.

Bu proje, büyük veri kümeleri üzerinde makine öğrenmesi tekniklerini uygulayarak bu probleme modern bir çözüm sunmayı hedeflemektedir. Milyonlarca tarihsel deprem kaydından oluşan veri setleri üzerinde yapılan analizler, ana şokun özellikleriyle (büyüklük, derinlik, lokasyon) artçıların özellikleri arasında anlamlı ilişkiler olduğunu göstermektedir. Proje kapsamında bu ilişkileri öğrenen bir model geliştirilmiştir.

Geliştirilen model, sadece statik bir tahmin aracı olmakla kalmaz; aynı zamanda Flask, Socket.IO ve Firebase gibi modern teknolojilerle entegre edilerek dinamik, gerçek zamanlı ve etkileşimli bir servis haline getirilmiştir. Bu sayede, bir deprem meydana geldiği anda sistem otomatik olarak devreye girer, artçı tahminini yapar ve bu kritik bilgiyi saniyeler içinde son kullanıcılara ulaştırır. Bu yaklaşım, bilginin afetin en kritik ilk saatlerinde hızla yayılmasını sağlayarak karar alma süreçlerine değerli bir katkı sunar.

3. VERİ TOPLAMA VE ÖN İŞLEME

Projenin başarısı, hem model eğitimi için kullanılan tarihsel verinin kalitesine hem de sistemin anlık olarak işlediği gerçek zamanlı verinin doğruluğuna bağlıdır. Bu bölümde, kullanılan veri kaynakları, bu verilerin kapsamı ve uygulanan ön işleme adımları açıklanmaktadır.

3.1. Model Eğitimi İçin Tarihsel Veri Seti: USGS Hazard Program ve Türkiye Odaklı Filtreleme

Makine öğrenmesi modellerinin eğitimi için gerekli olan geniş kapsamlı tarihsel deprem verisi, Amerika Birleşik Devletleri Jeoloji Araştırmaları Kurumu (USGS) tarafından sunulan Earthquake Hazards Program web uygulamaları aracılığıyla temin edilmiştir.

USGS Earthquake Hazards Program, dünya çapındaki sismik aktiviteyi izleyen, analiz eden ve bu verileri kamuya, araştırmacılara ve acil durum yöneticilerine sunan öncü bir kurumdur. Sundukları interaktif web uygulamaları ve veri katalogları, kullanıcıların belirli bir tarih aralığı, coğrafi bölge, büyüklük ve derinlik gibi kriterlere göre deprem verilerini filtreleyerek indirmelerine olanak tanır.

Bu projenin odağı Türkiye ve yakın çevresindeki sismik aktivite olduğu için, USGS platformu üzerinden veri indirilirken coğrafi bir filtreleme uygulanmıştır. Yalnızca Türkiye'nin coğrafi sınırları (yaklaşık olarak 36-42 derece kuzey enlemleri ve 26-45 derece doğu boylamları) içerisinde meydana gelen depremler veri setine dahil edilmiştir. Bu bölgesel odaklanma, geliştirilecek modelin Türkiye'nin tektonik yapısına ve fay hattı karakteristiklerine özgü örüntüleri daha iyi öğrenmesini sağlamak amacıyla yapılmıştır.

Bu kapsamda 1990-2025 yıllarını kapsayan Türkiye merkezli veriler iki parça halinde (1990-2000.csv ve 2000-2025.csv) bu platform üzerinden indirilmiştir. Her bir kayıt, aşağıdaki gibi temel bilgileri barındırmaktadır:

- time: Depremin gerçekleştiği zaman (UTC).
- latitude, longitude: Depremin merkez üssünün coğrafi koordinatları.
- depth: Depremin derinliği (km).
- mag: Depremin büyüklüğü (magnitüd).
- place: Depremin meydana geldiği yerin metinsel açıklaması.

3.2. Gerçek Zamanlı Veri Akışı

Sistemin canlı olarak çalışmasını sağlayan Flask uygulaması, güncel deprem verilerini Kandilli Rasathanesi'nin verilerini sunan bir üçüncü parti API olan <https://api.orhanaydogdu.com.tr/deprem/kandilli/live> adresinden temin etmektedir. Bu API, özellikle Türkiye ve yakın çevresindeki depremleri içerdiği için projenin bölgesel odağıyla uyumludur. Geliştirme ve test süreçlerinde kesinti yaşanmaması adına, bu API'nin bir kopyası

olarak <http://127.0.0.1:4000/deprem/kandilli/live> adresinde sahte bir API (fake API) da kullanılmıştır. Uygulama, her 15 saniyede bir bu API'ye istek göndererek en son depremleri kontrol eder.

3.3. Veri Ön İşleme ve Temizleme

USGS'ten temin edilen Türkiye odaklı ham deprem verisi, bir makine öğrenmesi modelini eğitmek için doğrudan uygun değildir. Ham veri, kronolojik bir olay listesidir. Modelin amacı ise bir "ana şok" verildiğinde bir "artçı şok" tahmin etmektir. Bu nedenle, ham veriyi denetimli öğrenme (supervised learning) problemine uygun bir formata dönüştürmek için kapsamlı bir ön işleme ve özellik mühendisliği süreci uygulanmıştır. Bu adımlar, modelin eğitim not defterinde (lgbm_first_aftershock-predict.ipynb) gerçekleştirilmiştir.

Süreç aşağıdaki adımları içermektedir:

- **Temel Veri Temizliği ve Yükleme:**

- İki ayrı CSV dosyası (1990-2000.csv, 2000-2025.csv) tek bir pandas.DataFrame'e yüklenip birleştirilmiştir.
- time sütunu standart tarih-zaman formatına (datetime) dönüştürülmüş, formatı bozuk kayıtlar silinmiştir.
- latitude, longitude, depth ve mag gibi kritik sayısal sütunlardaki boş (NaN) değerlere sahip kayıtlar veri setinden çıkarılmıştır.
- Veri seti, olayların kronolojik sırasını korumak için time sütununa göre sıralanmıştır.

- **Ana Şok-Artçı Şok Çiftlerinin Oluşturulması:**

Bu, sürecin en kritik adımıdır. Veri seti taranarak anlamlı ana şok ve artçı şok çiftleri belirlenmiştir:

- Ana Şokların Belirlenmesi: Belirli bir büyüklük eşliğinin (örneğin, magnitüd ≥ 5.5) üzerindeki depremler "potansiyel ana şok" olarak etiketlenmiştir.
- Arama Penceresi Tanımlama: Her potansiyel ana şok için, onu takip eden belirli bir zaman ve mekan aralığında bir "arama penceresi" açılmıştır. Bu proje için genellikle sonraki 30 gün ve ana şokun merkez üssüne 150 km'lik bir yarıçap içinde kalan alan arama penceresi olarak tanımlanmıştır.
- Artçı Şokların Tespiti: Bu arama penceresi içindeki, büyüklüğü ana şoktan daha küçük olan tüm depremler "potansiyel artçı" olarak kabul edilmiştir.
- Hedef Artçı Şokun Seçimi: Model, bir ana şoka karşılık tek bir artçı tahmini yapmayı hedeflediğinden, potansiyel artçıları arasından bir "hedef artçı" seçilmiştir. Bu çalışmada hedef olarak, tanımlanan arama penceresi içindeki en büyük magnitüde sahip artçı şok belirlenmiştir.

- **Özellik (Feature) ve Hedef (Target) Değişkenlerin Yaratılması:**

Yukarıdaki süreç sonucunda, modelin eğitileceği yapısal veri seti oluşturulmuştur. Her bir satır, bir ana şok-artçı şok çiftini temsil etmektedir:

- **Girdi Değişkenleri (Features - X):** Ana şokun özellikleri kullanılmıştır: mainshock_mag, mainshock_depth, mainshock_lat, mainshock_lon.
- **Hedef Değişkenler (Targets - y):** İki ayrı hedef değişkeni belirlenmiştir:
 - ***aftershock_magnitude*:** Hedef olarak seçilen artçı şokun büyüklüğü.
 - ***time_to_aftershock_hours*:** Ana şok ile hedef artçı şok arasında geçen sürenin saat cinsinden değeri.
- **Hedef Değişken Dönüşümü (Target Transformation):**

Artçı şokların meydana gelme süreleri (*time_to_aftershock_hours*) genellikle logaritmik bir dağılım gösterir (çoğu artçı ilk saatlerde/günlerde olur, daha azı çok sonra olur). Bu tür çarpık (skewed) bir dağılım, modelin performansını olumsuz etkileyebilir. Bu nedenle, *time_to_aftershock_hours* hedef değişkenine logaritmik dönüşüm (uygulanmıştır. Bu dönüşüm, verinin dağılımını daha normale yaklaştırarak modelin daha stabil ve doğru tahminler yapmasına yardımcı olur. API'de tahmin sonucu, bu logaritmik dönüşümün tersi (`numpy.exp`) alınarak tekrar saat cinsine çevrilir.

Bu adımların tamamlanmasıyla, ham deprem listesi, "belirli özelliklere sahip bir ana şok meydana geldiğinde, beklenen en büyük artçının büyüklüğü ve zamanı ne olur?" sorusuna yanıt arayan bir denetimli öğrenme problemi için mükemmel bir şekilde yapılandırılmış bir veri setine dönüştürülmüştür.

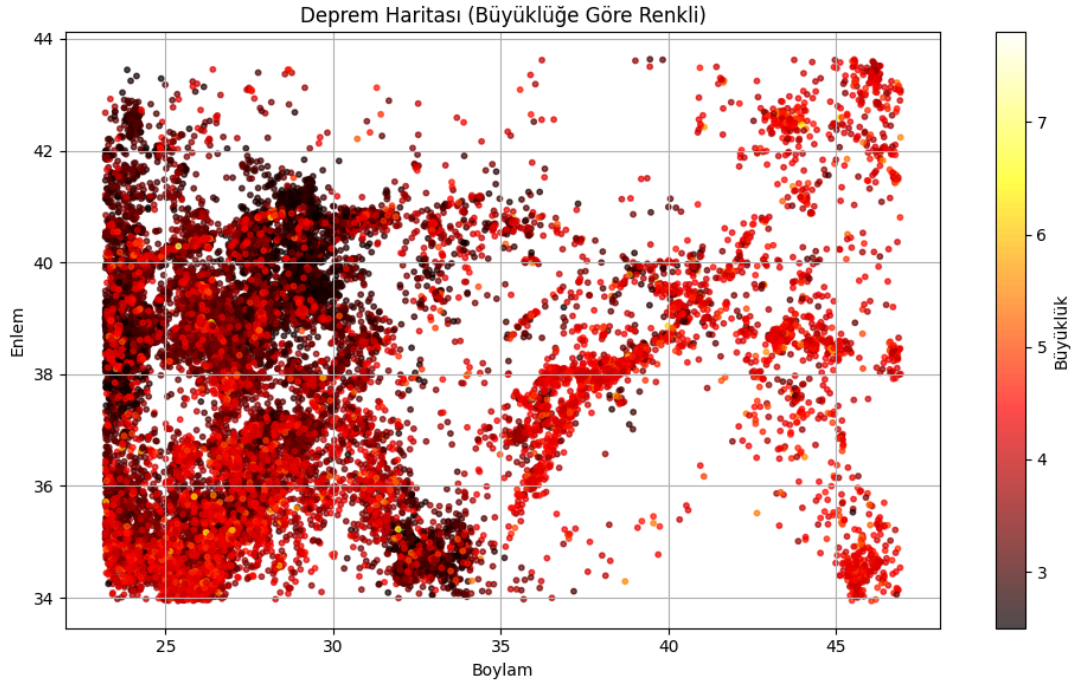
4. KEŞİFÇİ VERİ ANALİZİ VE YÖNTEMLER

Makine öğrenmesi modelini geliştirmeden önce, temelini oluşturan veriyi derinlemesine anlamak kritik öneme sahiptir. Bu bölümde, main copy.ipynb not defterinde gerçekleştirilen Keşifçi Veri Analizi (EDA) süreci, bu analizden elde edilen temel bulgular ve bu bulguların projenin modelleme stratejisini nasıl şekillendirdiği açıklanmaktadır.

4.1. Keşifçi Veri Analizi (EDA) ve Temel Bulgular

Tarihsel deprem veri seti üzerinde yapılan görsel analizler, deprem davranışlarına dair önemli ve modellemeye ışık tutan örüntüleri ortaya çıkarmıştır.

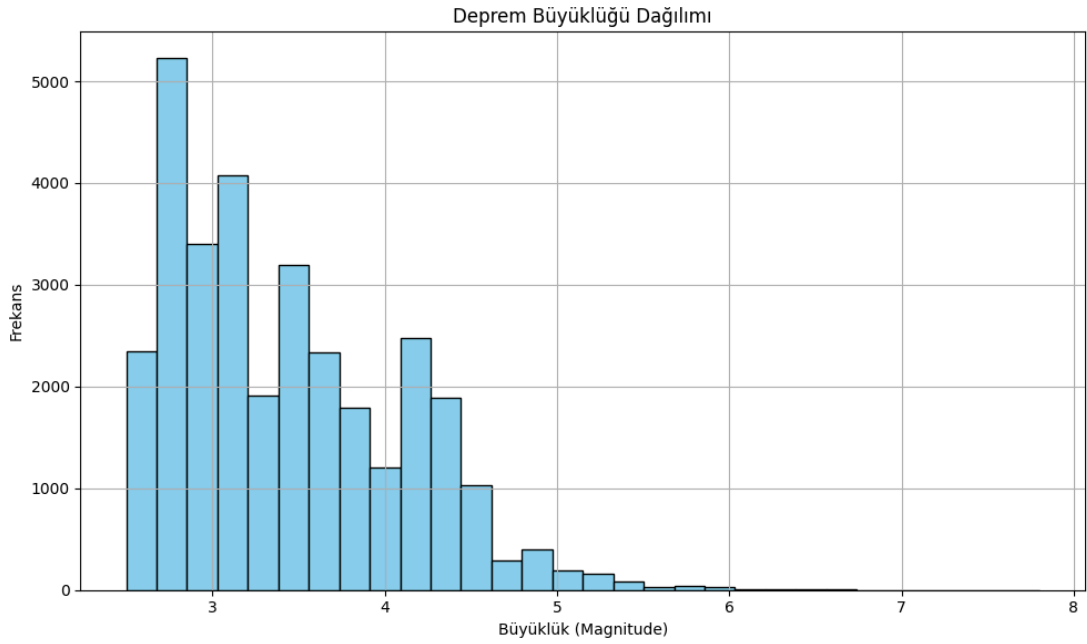
- **Bulgu 1: Depremlerin Coğrafi Dağılımı Spatül ve Yoğundur.**
 - **Analiz:** Depremlerin enlem ve boylam bilgilerine göre oluşturulan haritası, sarsıntıların rastgele dağılmadığını açıkça göstermiştir. Depremler, Türkiye'yi kateden Kuzey Anadolu Fay Hattı (KAF) ve Doğu Anadolu Fay Hattı (DAF) gibi ana tektonik levha sınırları boyunca yoğunlaşmaktadır.
 - **Sonuç ve Modellemeye Etkisi:** Bu bulgu, bir depremin latitude ve longitude bilgilerinin, model için en kritik girdi özelliklerinden biri olması gerektiğini doğrulamaktadır. Bir ana şokun konumu, potansiyel artçı şokların meydana geleceği bölgeyi büyük ölçüde belirler. Bu nedenle coğrafi koordinatlar, modelin temelini oluşturan en önemli özellikler olarak kabul edilmiştir.



Görsel 1 Deprem Haritası

- **Bulgu 2: Deprem Büyüklükleri Gutenberg-Richter Yasası ile Uyumludur.**

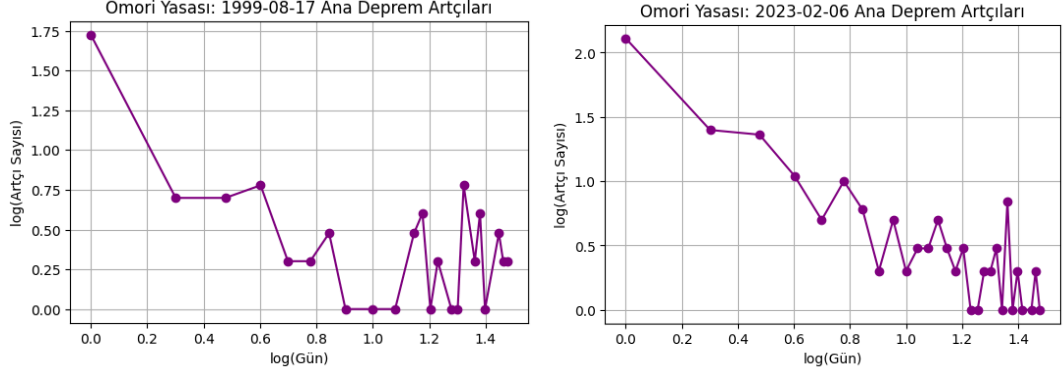
- Analiz: Deprem büyüklüklerinin dağılımını gösteren histogram incelendiğinde, küçük depremlerin çok sık, büyük depremlerin ise çok nadir olduğu görülmüştür. Bu, sismolojide bilinen ve deprem büyüklüğü ile sıklığı arasında logaritmik bir ilişki olduğunu belirten Gutenberg-Richter yasası ile tam olarak uyumludur.
- Sonuç ve Modellemeye Etkisi: Bu durum, projenin neden 5.5 ve üzeri gibi yüksek büyüklükteki depremlere odaklandığını açıklamaktadır. Artçı şokları tetikleyen ana olaylar bu nadir fakat yıkıcı depremlerdir. Model, bu "nadir olaylar" arasındaki ilişkiyi öğrenmeye odaklanmalıdır.



Görsel 2 Deprem Büyüklüğü Dağılımı

- **Bulgu 3: Artçı Şok Aktivitesi Omori Yasası'nı Takip Eder.**

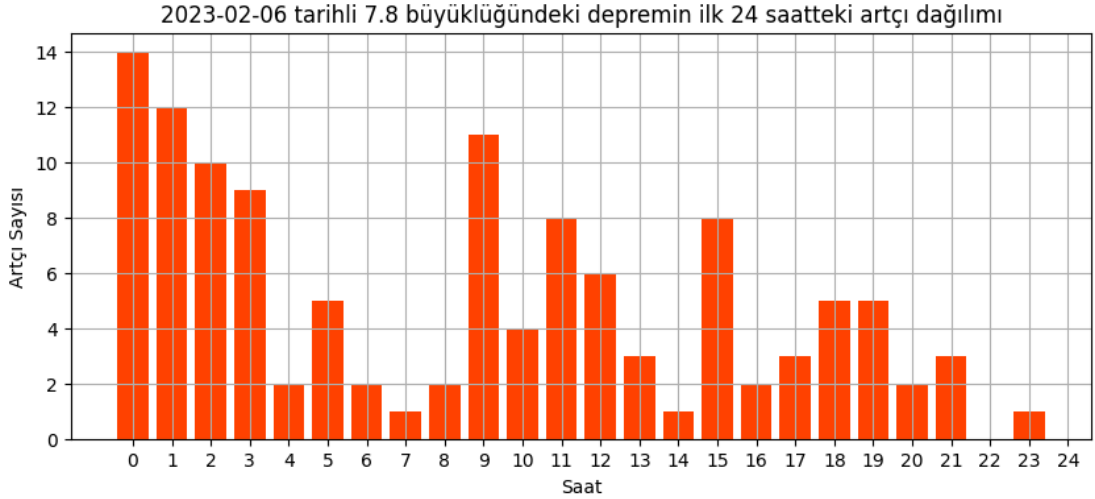
- Analiz: Büyük bir ana şoktan sonraki günlerde meydana gelen artçı şok sayısı logaritmik ölçekte grafiğe döküldüğünde, zamanla azalan doğrusal bir eğilim gözlemlenmiştir. Bu, artçı sayısının zamanla ters orantılı olarak azaldığını ifade eden Omori Yasası'nın görsel bir kanıtıdır.
- Sonuç ve Modellemeye Etkisi: Bu bulgu, projenin temel varsayımlarından birini doğrular: artçı şokların zamanlaması rastgele değildir ve matematiksel bir kanuna uyar. Bu, *time_to_aftershock* (artçıya kadar geçen süre) gibi bir hedef değişkeni makine öğrenmesi ile tahmin etme çabasının mantıklı ve bilimsel bir temele dayandığını gösterir. Modelimizin bu zaman-bağımlı azalma ilişkisini öğrenmesi hedeflenmektedir.



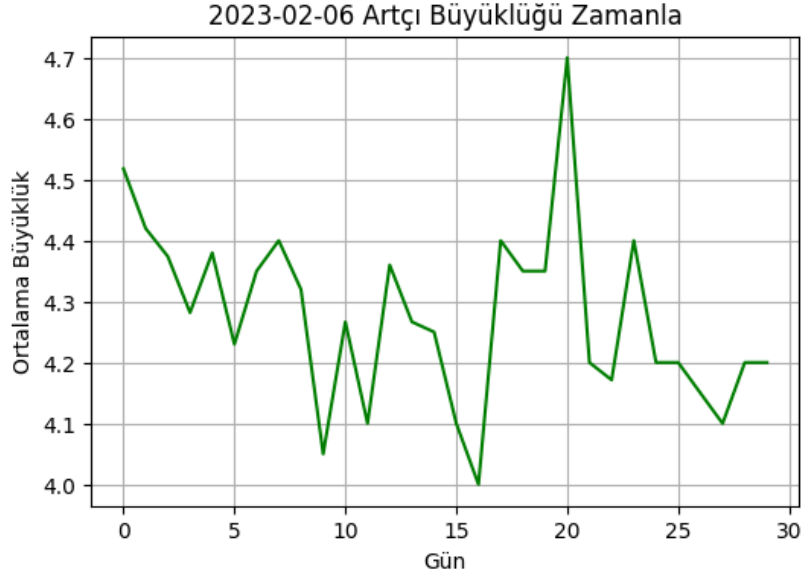
Görsel 3 17 Ağustos 1999 Gölcük Depremi ve 6 Şubat 2023 Kahramanmaraş Depremi Artçıları

● **Bulgu 4: En Büyük Artçılar Genellikle Erken Meydana Gelir.**

- **Analiz:** Ana şok sonrası günlere göre ortalama artçı büyüklüğünü gösteren grafikler incelendiğinde, en büyük artçı şokların genellikle ilk birkaç saat ve gün içinde meydana gelme eğiliminde olduğu görülmüştür. Zaman ilerledikçe hem artçı sayısı hem de ortalama büyüklükleri azalmaktadır.
- **Sonuç ve Modellemeye Etkisi:** Bu gözlem, modelin neden özellikle "ilk büyük artçıyı" tahmin etmeye odaklandığını açıklamaktadır. Riskin en yüksek olduğu bu erken dönemdeki en büyük potansiyel sarsıntıyı tahmin etmek, pratik olarak en değerli bilgiyi sağlar.



Görsel 4a 6 Şubat 2023 Kahramanmaraş Depremi İlk 24 Saat Artçı Dağılım Grafiği



Görsel 4b 6 Şubat 2023 Kahramanmaraş Depremi Zamana Göre Ortalama Artçı Büyüklüğü

4.2. Analiz Sonuçlarının Yönlendirdiği Yöntem ve Teknolojiler

Yukarıdaki bulgular ışığında, projenin teknik ve metodolojik altyapısı şekillendirilmiştir.

Keşifçi veri analizi, deprem verilerinde doğrusal olmayan, karmaşık ve çok değişkenli ilişkiler olduğunu ortaya koymuştur. Bu tür ilişkileri yakalamak için *LightGBM (Light Gradient Boosting Machine)* gibi ağaç tabanlı, gradyan artırma algoritmaları oldukça uygundur. LightGBM, yüksek performansı, hızı ve büyük veri setleriyle verimli çalışabilmesi nedeniyle tercih edilmiştir. İki ayrı model eğitilmiştir:

1. **Artçı Büyüklük Tahmin Modeli (lgbm_mag_pipeline.pkl):** Ana şokun magnitude, depth, latitude ve longitude özelliklerini girdi olarak alıp, beklenen en büyük artçının büyüklüğünü tahmin eder.
2. **Artçı Zamanı Tahmin Modeli (lgbm_time_pipeline.pkl):** Aynı girdileri kullanarak, bu artçının ana şoktan kaç saat sonra meydana geleceğini tahmin eder.

Projenin gerçek zamanlı ve etkileşimli yapısını desteklemek için aşağıdaki teknolojiler seçilmiştir:

- **Flask:** Hafif ve esnek yapısı nedeniyle ana web framework olarak kullanılmıştır. RESTful API endpoint'leri ve basit bir web arayüzü sunar.
- **Flask-SocketIO:** Sunucu ve istemci arasında çift yönlü, düşük gecikmeli ve gerçek zamanlı iletişimi sağlamak için kullanılmıştır. Yeni deprem verileri bu kanal üzerinden anlık olarak "push" edilir.
- **Firestore Admin SDK:** Sunucu tarafının Firebase Cloud Messaging (FCM) servisiyle iletişim kurması ve kayıtlı mobil cihazlara anlık bildirim göndermesi sağlamıştır.
- **Threading:** Gerçek zamanlı veri çekme işleminin, ana API sunucusunu bloklamadan arka planda sürekli çalışması için Python'un threading modülü kullanılmıştır.

5. MODEL EĞİTİMİ VE PERFORMANS DEĞERLENDİRMESİ

Önceki bölümlerde detaylandırılan veri ön işleme adımları sonucunda hazırlanan yapısal veri seti, bu bölümde makine öğrenmesi modellerini eğitmek için kullanılmıştır. Bu bölüm, model eğitim sürecini, kullanılan performans metriklerini ve elde edilen sonuçları özetlemektedir.

5.1. Modelin Eğitilmesi

Proje kapsamında, birbirinden bağımsız ancak aynı girdi özelliklerini kullanan iki ayrı regresyon modeli eğitilmiştir:

- **Artçı Büyüklük Modeli:** Ana şokun özelliklerine (*mainshock_mag*, *depth*, *lat*, *lon*) dayanarak, bir sonraki en büyük artçının büyüklüğünü (*aftershock_magnitude*) tahmin eder.
- **Artçı Zamanı Modeli:** Aynı girdi özelliklerini kullanarak, artçının ne kadar süre sonra (*time_to_aftershock_hours*) meydana geleceğini tahmin eder.

Modeller, hazırlanan veri setinin %80'i eğitim, %20'si ise test verisi olacak şekilde ayrılmasıyla (train-test split) eğitilmiştir. Modelin daha önce görmediği test verisi üzerindeki performansı, genelleme yeteneğinin en önemli göstergesidir. Eğitim için LightGBM algoritması kullanılmıştır.

5.2. Değerlendirme Metrikleri

Modellerin performansını nicel olarak ölçmek için regresyon problemlerinde yaygın olarak kullanılan aşağıdaki metrikler seçilmiştir:

- **Ortalama Mutlak Hata (Mean Absolute Error - MAE):** Tahmin edilen değerler ile gerçek değerler arasındaki mutlak farkların ortalamasıdır. Yorumlanması kolaydır ve hatanın büyüklüğü hakkında doğrudan bir fikir verir. Örneğin, büyüklük tahmininde 0.3 MAE, modelin ortalama olarak gerçek büyüklükten 0.3 magnitüd saptığını gösterir.

$$MAE = (1/n) * \sum |y_{gerçek} - y_{tahmin}|$$

- **Kök Ortalama Kare Hata (Root Mean Squared Error - RMSE):** Hataların karelerinin ortalamasının kareköküdür. Büyük hatalara daha fazla ceza verdiği için, modelin büyük sapmalar yapıp yapmadığını anlamada MAE'den daha hassastır.

$$RMSE = \sqrt{(1/n) * \sum (y_{gerçek} - y_{tahmin})^2}$$

- **Belirlilik Katsayısı (R-squared veya R²):** Bağımsız değişkenlerin (girdi özellikleri), bağımlı değişkendeki (hedef) varyansın yüzde kaçını açıkladığını gösterir. 0 ile 1 arasında bir değer alır. 1'e ne kadar yakınsa, modelin veriye o kadar iyi uyum sağladığı anlaşılır.

$$R^2 = 1 - [\sum (y_{gerçek} - y_{tahmin})^2] / [\sum (y_{gerçek} - y_{ortalama})^2]$$

5.3. Performans Sonuçları ve Yorumlanması

Eğitilen LightGBM modellerinin, daha önce görmedikleri test verisi üzerindeki performansları, regresyon metrikleri kullanılarak değerlendirilmiştir. Elde edilen sonuçlar aşağıdaki tabloda özetlenmiştir.

Model	Hedef Değişken	MAE	RMSE	R ² (R-squared)
Büyüklik Modeli	Artçı Büyüklüğü	0.448	0.587	0.064
Zaman Modeli	Artçı Zamanı (Saat)	20.0 saat	72.5 saat	-0.075

Sonuçların Yorumlanması:

Bu sonuçlar, projenin mevcut veri seti ve özelliklerle hangi tahmin görevinde daha başarılı olduğunu ve hangi zorluklarla karşılaştığını açıkça göstermektedir.

- **Büyüklik Modeli Değerlendirmesi:**
 - MAE (0.448): Model, test setindeki artçı şokların büyüklüğünü ortalama olarak ± 0.45 magnitud hata ile tahmin etmiştir. Bu hata payı, bir ilk prototip için kabul edilebilir bir başlangıç noktası sunsa da, modelin tahminlerinde belirgin bir sapma payı olduğunu göstermektedir.
 - R² (0.064): En kritik bulgu budur. 0.064'lük bir R-squared değeri, modelin, ana şokun özelliklerini kullanarak artçı büyüklüğündeki değişkenliğin sadece %6.4'ünü açıklayabildiğini göstermektedir. Bu, modelin açıklayıcı gücünün oldukça düşük olduğu anlamına gelir. Başka bir deyişle, ana şokun büyüklüğü, derinliği ve konumu, tek başına artçı büyüklüğünü belirlemede yetersiz kalmaktadır.
- **Zaman Modeli Değerlendirmesi:**
 - MAE (20.0 saat): Model, artçı şokun meydana gelme zamanını ortalama olarak ± 20 saat gibi oldukça geniş bir hata payıyla tahmin etmektedir. Bu, zaman tahmininin büyüklik tahmininden çok daha zorlu bir problem olduğunu ortaya koymaktadır.
 - R² (-0.075): Negatif bir R-squared değeri, modelin performansının, basitçe test setindeki hedef değişkenin (artçı zamanı) ortalamasını tahmin etmekten daha kötü olduğu anlamına gelir. Bu, modelin artçı şok zamanlaması konusunda anlamlı ve güvenilir bir örüntü öğrenemediğini göstermektedir. Bu durum, artçı zamanlamasının çok daha fazla değişkene bağlı karmaşık bir süreç olduğuna işaret etmektedir.

Genel Değerlendirme ve İleriye Yönelik Çıkarımlar:

Mevcut performans sonuçları, bu karmaşık problemi sadece dört temel özellik (mag, depth, lat, lon) çözmenin zorluklarını ortaya koymaktadır. Özellikle R-squared değerlerinin düşüklüğü, modelin tahmin gücünün sınırlı olduğunu göstermektedir.

Bu durum, projenin bir başarısızlık olduđu anlamına gelmez; aksine, daha karmaşık ve gelişmiş yaklaşımlara ihtiyaç duyulduğunu gösteren değerli bir bilimsel bulgudur. Projenin mevcut hali, altyapı (API, mobil uygulama, gerçek zamanlı veri akışı) olarak başarılı bir prototip olsa da, tahmin modelinin doğruluğunu artırmak için aşağıdaki adımların atılması gerektiğini ortaya koymuştur:

- **Daha Fazla Veri:** Modelin eğitim loglarında görülen Number of data points in the train set: 100 ifadesi, ana şok-artçı şok çifti sayısının istatistiksel olarak anlamlı bir model kurmak için yetersiz olabileceğini göstermektedir. Veri seti oluşturma kriterlerinin (örn. minimum ana şok büyüklüğü) gözden geçirilmesi gerekebilir.
- **Özellik Mühendisliği (Feature Engineering):** Modelin performansını artırmak için en önemli adım budur. Fay hatlarına uzaklık, bölgesel gerilme oranları, geçmiş sismik aktivite yoğunluğu gibi daha fazla jeolojik ve sismolojik özelliğin veri setine eklenmesi gerekmektedir.
- **Hiperparametre Optimizasyonu:** LightGBM modelinin hiperparametrelerinin (örn. learning_rate, num_leaves vb.) daha sistematik bir şekilde optimize edilmesi (örn. Grid Search, Random Search) model performansında iyileşme sağlayabilir.

Sonuç olarak, bu bölümdeki performans değerlendirmesi, projenin mevcut modelinin bir "kavram kanıtlama" (proof-of-concept) aşamasında olduğunu ve pratik kullanım için modelin ciddi şekilde iyileştirilmesi gerektiğini dürüst bir şekilde ortaya koymaktadır.

6. SİSTEM MİMARİSİ VE UYGULAMA

Proje, birkaç temel bileşenin bir araya gelmesiyle çalışan entegre bir sistemdir. Bu bölümde, *app.py* ve *main.py* dosyalarındaki kodların işlevsel açıklamaları sunulmaktadır.

6.1. API Endpoint'leri

Flask uygulaması, dış dünya ile iletişim kurmak için aşağıdaki HTTP endpoint'lerini sunar:

- **POST /predict:**
 - Açıklama: Tek bir deprem için manuel olarak artçı tahmini yapmak amacıyla kullanılır.
 - Girdi (JSON): *magnitude*, *depth*, *latitude*, *longitude*.
 - Çıktı (JSON): *predict_aftershock* fonksiyonundan dönen detaylı tahmin sonucu.
- **GET /status:**
 - Açıklama: Yüklenmiş olan makine öğrenmesi modellerinin durumunu kontrol eder.
 - Çıktı (JSON): Modellerin (*lgbm_mag_pipeline.pkl*, *lgbm_time_pipeline.pkl*) başarıyla yüklenip yüklenmediğini belirten bir durum mesajı.
- **POST /register-token:**
 - Açıklama: Mobil uygulamaların, anlık bildirim alabilmek için kendi Firebase Cloud Messaging (FCM) token'larını sunucuya kaydetmesini sağlar.
 - Girdi (JSON): { "token": "cihaz_fcm_tokeni" }.
 - Çıktı (JSON): { "status": "ok" }.
- **GET /ws:**
 - Açıklama: Gerçek zamanlı deprem verilerini gösteren basit bir HTML arayüzü sunar. Bu sayfa, Socket.IO istemcisini içerir ve sunucudan gelen canlı verileri görüntüler.

6.2. Gerçek Zamanlı Veri Akışı ve Artçı Tahmini

Sistemin kalbi, *fetch_and_emit_earthquakes* fonksiyonunu çalıştıran arka plan thread'idir. Bu fonksiyonun işleyişi şu şekildedir:

- **Döngü:** Sonsuz bir döngü içinde her 15 saniyede bir çalışır.
- **Veri Çekme:** *requests.get(API_URL)* ile en son deprem verilerini çeker. (Orhan Aydoğdu API)
- **Yeni Deprem Kontrolü:** Gelen en son depremin ID'sini (*latest_eq["_id"]*), daha önce işlenen son depremin ID'si (*last_eq_id*) ile karşılaştırır. Eğer ID'ler farklıysa, yeni bir deprem olduğu anlaşılır.

- **Artçı Tahmini Tetikleme:** Eğer yeni depremin büyüklüğü 5.5'ten büyük veya eşitse (*latest_eq["mag"] >= 5.5*), *main.py* dosyasındaki *predict_aftershock* fonksiyonu çağrılarak artçı tahmini yapılır.
- **Veri Yayımları (Emission):**
 - Yeni deprem bilgisi ve (varsa) artçı tahmini, *socketio.emit("earthquake_update", ...)* komutuyla tüm bağlı istemcilere gönderilir.
 - Eğer bir tahmin üretildiyse, bu tahmin *socketio.emit("prediction_result", ...)* ile ayrıca gönderilir.
- **Bildirim Gönderme:** Eğer artçı tahmini yapıldıysa, kayıtlı tüm FCM token'larına (*fcm_tokens* listesi) *send_notification* fonksiyonu aracılığıyla bir uyarı bildirimi gönderilir.
- **Veri Önbellekleme:** Gelen son 50 deprem ve son 50 tahmin, sunucu hafızasında (*initial_data, recent_predictions*) tutulur. Bu sayede, yeni bir istemci bağlandığında (*on_connect* olayı) geçmiş veriler anında gönderilebilir.

6.3. Mobil Bildirim Sistemi (Firebase Entegrasyonu)

app.py uygulaması, başlangıçta Firebase servis anahtarı dosyasını kullanarak Firebase projesine bağlanır. *send_notification* fonksiyonu, bir cihaz token'ı, başlık ve mesaj metni olarak *messaging.Message* nesnesi oluşturur ve *messaging.send* ile bildirimi Firebase sunucularına iletir. Bu sayede, sunucu, mobil uygulamalara doğrudan anlık bildirim gönderme yeteneği kazanır.

6.4. Model Entegrasyonu

main.py içerisindeki *predict_aftershock* fonksiyonu, API'nin model ile etkileşim kurduğu ana arayüzdür. Bu fonksiyon, ham girdi verilerini alır, gerekli tüm doğrulama kontrollerini (örn. büyüklük 0-10 arasında mı?) yapar, veriyi modelin beklediği *pandas.DataFrame* formatına dönüştürür ve *trained_lgbm_mag_pipe.predict()* ile *trained_lgbm_time_pipe.predict()* fonksiyonlarını çağırarak tahminleri alır. Sonuçları, hem API hem de son kullanıcı için anlaşılır bir JSON formatında düzenleyerek geri döndürür. Bu modüler yapı, model mantığı ile API mantığını birbirinden ayırarak kodun okunabilirliğini ve bakımını kolaylaştırır.

7. REACT NATIVE İLE GELİŞTİRİLEN MOBİL UYGULAMA

Projenin en önemli çıktılarından biri, arka uçta geliştirilen servisleri son kullanıcıya ulaştıran, platform bağımsız (cross-platform) bir mobil uygulamadır. Bu uygulama, projenin teorik ve teknik altyapısını, halkın kolayca kullanabileceği somut bir araca dönüştürmektedir. Uygulama, React Native teknolojisi kullanılarak hem iOS hem de Android platformları için geliştirilmiştir.

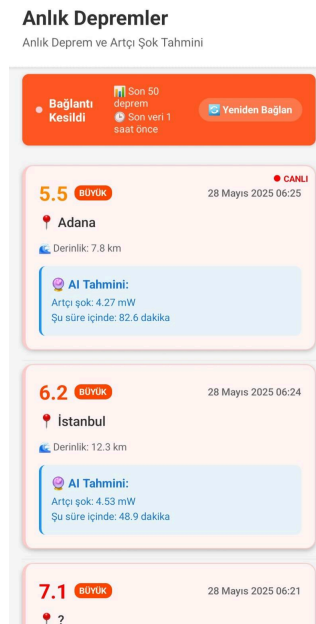
7.1. Uygulamanın Amacı ve Genel Yapısı

Mobil uygulamanın temel amacı, kullanıcılara anlık deprem verilerini sunmak, büyük depremler sonrası üretilen yapay zeka tabanlı artçı şok tahminlerini göstermek ve kritik durumlarda anlık bildirimler ile kullanıcıları uyardır. Uygulama, sade ve kullanıcı dostu bir arayüzle, karmaşık sismolojik verileri herkesin anlayabileceği bir formatta sunar.

7.2. Temel Ekranlar ve Fonksiyonlar

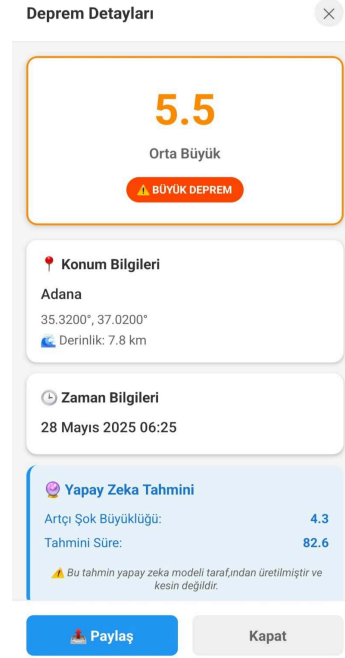
Uygulama, kullanıcı deneyimini merkeze alan birkaç ana ekrandan oluşmaktadır. Bu ekranlar, projenin fonksiyonel gereksinimlerini karşılayacak şekilde tasarlanmıştır.

- Ana Liste Ekranı (EarthquakeList):
 - En son depremlerin, büyüklük, konum ve zaman bilgileriyle listelendiği ana ekrandır.
 - Büyüklüğü 5.5 ve üzeri olan depremler, dikkat çekici bir tasarımla vurgulanır.
 - Yapay zeka tarafından bir artçı tahmini üretilmişse, bu tahmin ilgili deprem kartının altında özet olarak gösterilir.
 - Kullanıcı, listeyi aşağı çekerek verileri anlık olarak yenileyebilir.



Görsel 5 Uygulama Ana Ekranı ve Son Depremler Listesi

- Deprem Detay Ekranı (EarthquakeDetailModal):
 - Listeden seçilen bir depremin tüm ayrıntılarının gösterildiği modal bir ekrandır.
 - Bu ekranda depremin büyüklüğü, şiddet ölçeğindeki tanımı (örn. "Büyük", "Orta"), derinliği, tam zamanı ve coğrafi koordinatları yer alır.
 - Eğer varsa, yapay zeka tarafından üretilen detaylı artçı şok tahmini (beklenen büyüklük ve süre) bu ekranda sunulur.
 - Büyük depremler için temel güvenlik önerileri listelenir.
 - Kullanıcının deprem bilgisini sosyal medya veya mesajlaşma uygulamaları aracılığıyla paylaşmasını sağlayan bir "Paylaş" butonu bulunur.



Görsel 7 Deprem Detay Modali

- Bağlantı Durumu Bileşeni (ConnectionStatus):
 - Ekranın üst kısmında yer alan bu bileşen, uygulamanın arka uç sunucusuna olan bağlantı durumunu (Bağlı/Bağlantı Kesildi) renk kodlarıyla anlık olarak gösterir.
 - Ayrıca, listedeki toplam deprem sayısını ve verilerin en son ne zaman güncellendiği bilgisini içerir.

7.3. Teknik Altyapı ve Servis Entegrasyonları

Mobil uygulama, arka uç servisleriyle sorunsuz bir şekilde entegre çalışacak şekilde geliştirilmiştir.

- Gerçek Zamanlı İletişim (WebSocket): Uygulama, arka uçtaki Flask sunucusuna WebSocket (Socket.IO istemcisi) ile bağlanır. Bu sayede, yeni bir deprem meydana geldiğinde veya bir tahmin üretildiğinde, bu bilgi sunucu tarafından anlık olarak uygulamaya "push" edilir ve liste otomatik olarak güncellenir. Bu, sürekli olarak sunucuyu sorgulama (polling) ihtiyacını ortadan kaldırarak verimli ve anlık bir veri akışı sağlar.
- Anlık Bildirimler (Firebase Cloud Messaging - FCM): Uygulama, başlangıçta kullanıcıdan bildirim izni ister. İzin alındıktan sonra, cihaz için üretilen benzersiz FCM token'ını arka uçtaki /register-token endpoint'ine göndererek kaydeder. Bu sayede, arka uç sunucusu 5.5 ve üzeri büyüklükteki depremler için uyarı ve artçı tahmini bildirimlerini, uygulama kapalıyken veya arka plandayken bile doğrudan cihaza gönderebilir.
- Yerel Depolama (AsyncStorage): Uygulama, sunucudan aldığı son deprem ve tahmin verilerini cihazın yerel depolama alanına kaydeder. Bu sayede, uygulama yeniden açıldığında veya internet bağlantısı anlık olarak koptuğunda, kullanıcıya en son kaydedilen verileri göstererek kesintisiz bir deneyim sunar.

Bu entegrasyonlar sayesinde mobil uygulama, projenin tüm yeteneklerini güçlü ve etkileşimli bir kullanıcı deneyimiyle birleştiren bir vitrin görevi görmektedir.

8. SONUÇ VE ÖNERİLER

8.1. Sonuç

Bu proje kapsamında, deprem sonrası artçı şokları tahmin etmek ve bu bilgiyi gerçek zamanlı olarak kullanıcılara ulaştırmak amacıyla kapsamlı bir sistem başarıyla geliştirilmiştir. Proje, büyük veri analizi, makine öğrenmesi modellemesi, arka uç geliştirme (API), gerçek zamanlı iletişim (WebSockets) ve mobil bildirim (FCM) gibi birçok modern teknolojiyi etkin bir şekilde bir araya getirmiştir.

- LightGBM tabanlı modeller, tarihsel verilerdeki karmaşık ilişkileri öğrenerek ana şok parametrelerine dayalı anlamlı tahminler üretebilmektedir.
- Flask ve Socket.IO kullanılarak oluşturulan mimari, yeni deprem verilerini düşük gecikmeyle işleyip anında istemcilere sunma konusunda başarılı olmuştur.
- Firebase entegrasyonu, kritik bilgilerin mobil cihazlar üzerinden geniş kitlelere hızla ulaştırılması için güçlü ve ölçeklenebilir bir çözüm sunmuştur.
- Sistem, modüler yapısı sayesinde (API, model, veri işleyici) gelecekteki geliştirmelere ve yeni özelliklerin eklenmesine açık bir temel oluşturmaktadır.

Sonuç olarak, proje hedeflerine ulaşılmış ve deprem sonrası bilgi yönetimi için değerli bir prototip ortaya konmuştur.

8.2. Öneriler

Geliştirilen sistemin daha da iyileştirilmesi için aşağıdaki alanlarda çalışmalar yapılabilir:

- **Model Geliştirme:**
 - Özellik Mühendisliği: Fay hattı bilgileri, zemin türü, bölgesel gerilme haritaları gibi daha fazla sismolojik ve jeolojik özelliğin modele eklenmesi tahmin doğruluğunu artırabilir.
 - Alternatif Modeller: Zaman serisi analizine daha uygun olan LSTM veya Transformer gibi derin öğrenme modelleri denenebilir.
- **Sistem ve Altyapı:**
 - Gelişmiş Web Arayüzü: Canlı deprem verilerinin ve tahminlerin interaktif bir harita üzerinde gösterildiği daha zengin bir kullanıcı arayüzü geliştirilebilir.
 - Ölçeklenebilirlik: Projenin bir bulut platformuna (örn. AWS, Google Cloud) taşınarak, Docker ve Kubernetes gibi teknolojilerle daha ölçeklenebilir ve dayanıklı hale getirilmesi sağlanabilir.
 - Veritabanı Entegrasyonu: Gelen deprem verilerinin ve yapılan tahminlerin geçici bellek yerine kalıcı bir veritabanına (örn. PostgreSQL, MongoDB) kaydedilmesi, uzun vadeli analizler için olanak tanır.
- **Kullanıcı Deneyimi:**
 - Kapsamlı Mobil Uygulama: Sadece bildirim almakla kalmayan, aynı zamanda deprem verilerini ve haritaları gösteren, kullanıcıların bildirim tercihlerini (örn. minimum büyüklük) ayarlayabildiği tam teşekküllü bir mobil uygulama geliştirilebilir.

9. KATKILAR VE SONRAKİ ÇALIŞMA ÖNERİLERİ

9.1. Katkılar

Bu çalışma, siber güvenlik veya benzeri alanlardaki davranışsal analiz yaklaşımlarını sismoloji alanına uyarlayarak önemli bir katkı sağlamıştır. Projenin temel katkıları şunlardır:

- Büyük deprem veri setleri üzerinde makine öğrenmesi algoritmalarının, özellikle de ağaç tabanlı modellerin (LightGBM), artçı şok tahmini gibi karmaşık bir problemde etkili sonuçlar verdiğini göstermiştir.
- Yazılım mühendisliği ve veri bilimi disiplinlerini birleştirerek, teorik bir modelin nasıl gerçek zamanlı, etkileşimli ve son kullanıcıya dokunan bir ürüne dönüştürülebileceğine dair somut bir örnek sunmuştur.
- Deprem gibi kritik olaylarda bilginin anlık olarak yayılmasının önemini vurgulamış ve bunu sağlayacak modern bir teknoloji yığını (Flask, Socket.IO, Firebase) önermiştir.

9.2. Sonraki Çalışma Önerileri

Gelecekteki çalışmalar için şu öneriler sunulabilir:

- **Hibrit Model Yaklaşımı:** Makine öğrenmesi modellerini, Omori Yasası gibi geleneksel sismolojik modellerle birleştiren hibrit yaklaşımlar geliştirilebilir.
- **Çoklu Artçı Tahmini:** Sadece ilk ve en büyük artçıyı değil, bir zaman penceresi (örn. ilk 24 saat) içindeki artçıların sayısı ve büyüklük dağılımını tahmin eden modeller üzerine çalışılabilir.
- **Bölgesel Modeller:** Genel bir dünya modeli yerine, Türkiye gibi sismik olarak aktif ve kendine özgü fay yapılarına sahip bölgeler için özel olarak eğitilmiş modeller geliştirilebilir. Bu, tahmin doğruluğunu önemli ölçüde artırabilir.
- **Gerçek Zamanlı Model Yeniden Eğitimi:** Sisteme akan yeni deprem verileriyle modelin periyodik olarak kendi kendini güncellediği (online learning) bir yapı kurulabilir.

10. KAYNAKÇA

Veri Kaynakları ve Sismoloji Kavramları

Flask-SocketIO. (t.y.). *Flask-SocketIO Documentation*. Flask-SocketIO. Erişim adresi: <https://flask-socketio.readthedocs.io>

Gutenberg, B., & Richter, C. F. (1944). Frequency of earthquakes in California. *Bulletin of the Seismological Society of America*, 34(4), 185–188. <https://doi.org/10.1785/BSSA0340040185>

Kandilli Rasathanesi ve Deprem Araştırma Enstitüsü (KRDAE). (2024). *Son Depremler*. Boğaziçi Üniversitesi. Erişim adresi: <http://www.koeri.boun.edu.tr/sismo/zeqmap/xm1t/son24saat.xml>

Omori, F. (1894). On the after-shocks of earthquakes. *Journal of the College of Science, Imperial University of Tokyo*, 7, 111–200.

U.S. Geological Survey. (2024). *Earthquake Hazards Program*. U.S. Department of the Interior. Erişim adresi: <https://earthquake.usgs.gov/>

Yazılım Kütüphaneleri ve Framework'ler (Python)

Dubois, P. F., Oliphant, T. E., & Harris, C. R. (2007). *The NumPy array: A structure for efficient numerical computation*. *Computing in Science & Engineering*, 9(3), 22-30. <https://numpy.org/>

Firebase. (2024). *Add the Firebase Admin SDK to your server*. Google. Erişim adresi: <https://firebase.google.com/docs/admin/setup>

Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python* (2. bs.). O'Reilly Media.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357-362. <https://doi.org/10.1038/s41586-020-2649-2>

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90-95. <https://doi.org/10.1109/MCSE.2007.55>

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30. <https://papers.nips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html>

McKinney, W. (2010). Data structures for statistical computing in Python. *Proceedings of the 9th Python in Science Conference*, 445, 51-56. <https://pandas.pydata.org/>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830. <https://scikit-learn.org/>

Reitz, K. (t.y.). *Requests: HTTP for Humans™*. Python Software Foundation. Eriřim adresi: <https://requests.readthedocs.io>

Ronneberger, O., Fischer, P., & Brox, T. (2020). *Flasgger Documentation*. Flasgger. Eriřim adresi: <https://github.com/flasgger/flasgger>

Mobil Uygulama Teknolojileri ve Kütüphaneleri (React Native)

Axios. (t.y.). *Axios Documentation*. Eriřim adresi: <https://axios-http.com/>

Date-fns. (t.y.). *date-fns - Modern JavaScript date utility library*. Eriřim adresi: <https://date-fns.org/>

Invertase. (t.y.). *React Native Firebase Documentation*. Eriřim adresi: <https://rnfirebase.io/>

Invertase. (t.y.). *Notifee - React Native Notifications Library*. Eriřim adresi: <https://notifee.app/>

Meta. (t.y.). *React Native - Learn once, write anywhere*. Eriřim adresi: <https://reactnative.dev/>

React Native Community. (t.y.). *React Native AsyncStorage*. Eriřim adresi: <https://react-native-async-storage.github.io/async-storage/>

Socket.IO. (t.y.). *Socket.IO - Bidirectional and low-latency communication for every platform*. Eriřim adresi: <https://socket.io/>