Dün

bu projem için google engine den veri cetim vanın tif oalrak yıllara ayırdım simdi naoıcam

Harika bir başlangıç yapmışsın Lütfiye! Van Gölü için yıllara göre .tif formatında uydu görüntülerini ayırmak, zaman serisi analizine geçiş için mükemmel bir temel. Şimdi sıradaki adımları adım adım sadeleştirerek ilerleyelim:

**🧭 1. Dosya Yapısını Düzenle**

Her yıl için ayrı klasör oluştur ve içindeki .tif dosyalarını net şekilde adlandır:

Kod

Van\_Golu\_NDWI/

├── 2018/

│ ├── van\_2018\_05.tif

│ ├── van\_2018\_06.tif

│ └── ...

├── 2019/

│ ├── van\_2019\_05.tif

│ └── ...

└── 2020/

└── ...

Bu yapı, hem veri birleştirme hem de zaman serisi çıkarımı için işleri kolaylaştırır.

**🧪 2. NDWI Hesapla**

Eğer görüntüler Sentinel-2 ise, NDWI şöyle hesaplanır:

python

import rasterio

import numpy as np

def calculate\_ndwi(green\_path, nir\_path, output\_path):

with rasterio.open(green\_path) as green, rasterio.open(nir\_path) as nir:

green\_band = green.read(1).astype('float32')

nir\_band = nir.read(1).astype('float32')

ndwi = (green\_band - nir\_band) / (green\_band + nir\_band)

ndwi[np.isnan(ndwi)] = -9999 # NoData

profile = green.profile

profile.update(dtype=rasterio.float32, count=1)

with rasterio.open(output\_path, 'w', \*\*profile) as dst:

dst.write(ndwi, 1)

Eğer .tif dosyalarında band ayrımı yoksa, önce bandları ayırmamız gerekebilir. Bunu da birlikte yapabiliriz.

**📈 3. Zaman Serisi Oluştur**

Her NDWI çıktısı için su alanını hesapla:

python

def get\_water\_area(ndwi\_path, threshold=0.3):

with rasterio.open(ndwi\_path) as src:

ndwi = src.read(1)

pixel\_size = src.res[0] \* src.res[1] # m²

water\_pixels = np.sum(ndwi > threshold)

return water\_pixels \* pixel\_size / 1e6 # km²

Sonra bu değerleri bir pandas DataFrame'e aktar:

python

import pandas as pd

data = {

'date': ['2018-05', '2018-06', '2019-05', ...],

'water\_area\_km2': [123.4, 120.1, 118.7, ...]

}

df = pd.DataFrame(data)

df.to\_csv('van\_ndwi\_timeseries.csv', index=False)

**🌍 4. Görselleştirme**

Zaman serisini çiz:

python

import matplotlib.pyplot as plt

df['date'] = pd.to\_datetime(df['date'])

plt.plot(df['date'], df['water\_area\_km2'], marker='o')

plt.title('Van Gölü Su Alanı Zaman Serisi')

plt.xlabel('Tarih')

plt.ylabel('Su Alanı (km²)')

plt.grid(True)

plt.show()

**🔮 5. Sonraki Adım: Model Hazırlığı**

* NDWI zaman serisini normalize et
* Feature engineering ile mevsimsel etkileri çıkar
* Transfer learning için veri setini train/test/val olarak ayır

İstersen bu adımları birlikte kodlayabiliriz. Mock veriyle başlayıp gerçek veriye geçebiliriz. Hangi adımda destek istersin? Kod, dosya yapısı, görselleştirme, ya da model hazırlığı?

sence görüntülerden sayısal cıkarım yapıp ondan mı egitmeliyim regresyon filan

Bu soruyu sorman çok yerinde Lütfiye—çünkü AquaTrack gibi çevresel izleme projelerinde “görüntüden doğrudan çıkarım mı yapmalı, yoksa sayısal veriye dönüştürüp regresyon gibi modellerle mi ilerlemeli?” sorusu aslında stratejik bir karar noktasıdır. Hadi birlikte sadeleştirelim:

**🎯 İki Yaklaşımın Mantığı**

| **Yaklaşım** | **Ne Yapılır?** | **Avantaj** | **Zorluk** |
| --- | --- | --- | --- |
| **Sayısal çıkarım + regresyon** | Görüntüden NDWI gibi metrikler çıkarılır → zaman serisi oluşturulur → regresyon/ML ile tahmin yapılır | Daha az veriyle çalışır, yorumlanabilirlik yüksek | Görüntüdeki mekânsal detaylar kaybolabilir |
| **Görüntüden doğrudan modelleme (U-Net, CNN)** | Görüntü doğrudan modele verilir → segmentasyon veya sınıflandırma yapılır | Mekânsal örüntüleri öğrenir, daha güçlü genelleme | Daha fazla veri ve donanım ister, yorumlaması zor olabilir |

**💡 AquaTrack için Hangisi Daha Uygun?**

Senin projenin iki bileşeni var:

1. **Su miktarı takibi** → NDWI zaman serisiyle regresyon gayet mantıklı.
2. **Su kalitesi tahmini (klorofil-a, bulanıklık vs.)** → Görüntüden doğrudan çıkarım + transfer learning daha etkili olabilir.

Yani:

* **Su miktarı için**: NDWI → su alanı (km²) → zaman serisi → regresyon/temporal modelleme ✅
* **Su kalitesi için**: Görüntü → U-Net veya CNN ile doğrudan tahmin ✅

**🔄 Hibrit Yaklaşım Önerisi**

Senin gibi sistematik çalışan biri için hibrit bir yapı çok uygun olur:

text

Uydu Görüntüsü (.tif)

├── NDWI çıkar → su alanı → regresyon modeli (miktar tahmini)

├── Klorofil-a tahmini → CNN/U-Net ile kalite tahmini

└── Zaman serisi + mekânsal haritalar → web panelde görselleştirme

Bu yapı hem yorumlanabilir hem de güçlü tahminler sunar. Ayrıca kamu kurumlarına sunacağın karar destek sisteminde “sayısal çıktı + görsel harita” kombinasyonu çok etkileyici olur.

İstersen örnek bir regresyon modeli ya da U-Net mimarisi için mock veriyle birlikte adım adım ilerleyebiliriz. Hangisinden başlamak istersin: NDWI regresyon mu, yoksa kalite tahmini için görüntü tabanlı model mi?

görüntü yıl içinde images ve ndwı masks oalrak iki ayrımda

Süper Lütfiye! Görüntüleri yıl içinde **images** (ham uydu görüntüleri) ve **NDWI masks** (su alanı maskeleri) olarak ikiye ayırman, U-Net gibi segmentasyon modelleri için tam da olması gereken veri yapısı. Bu yapı sayesinde hem mekânsal hem zamansal analiz yapabiliriz. Şimdi bu yapıyı nasıl model eğitimine dönüştüreceğini adım adım sadeleştireyim:

**🗂️ 1. Dosya Yapısı Önerisi (U-Net için)**

bash

Van\_Golu\_Dataset/

├── 2018/

│ ├── images/

│ │ ├── van\_2018\_05.tif

│ │ └── ...

│ └── masks/

│ ├── van\_2018\_05\_mask.tif

│ └── ...

├── 2019/

│ └── ...

└── 2020/

└── ...

Her görüntü ile maskesi aynı isimde olmalı (van\_2018\_05.tif ↔ van\_2018\_05\_mask.tif) ki eşleştirme kolay olsun.

**🧠 2. Model Eğitimi için Hazırlık**

**✅ Veri Hazırlama**

* Görüntüleri normalize et (0–1 arası)
* Maskeleri binary hale getir (su: 1, diğer: 0)
* Boyutları sabitle (örneğin 256x256 veya 512x512)

**🧪 Train/Test/Val Ayrımı**

python

from sklearn.model\_selection import train\_test\_split

image\_paths = [...] # tüm image dosya yolları

mask\_paths = [...] # eşleşen mask yolları

train\_img, val\_img, train\_mask, val\_mask = train\_test\_split(

image\_paths, mask\_paths, test\_size=0.2, random\_state=42)

**🧬 3. U-Net Modeli (PyTorch örneği)**

python

import torch.nn as nn

class UNet(nn.Module):

def \_\_init\_\_(self):

super(UNet, self).\_\_init\_\_()

# Encoder

self.enc1 = nn.Sequential(nn.Conv2d(1, 64, 3, padding=1), nn.ReLU(), nn.Conv2d(64, 64, 3, padding=1), nn.ReLU())

# Decoder

self.dec1 = nn.Sequential(nn.Conv2d(64, 1, 1), nn.Sigmoid())

def forward(self, x):

x1 = self.enc1(x)

out = self.dec1(x1)

return out

Bu sade versiyon. İstersen ResNet backbone’lu veya Attention U-Net gibi gelişmiş versiyonlara da geçebiliriz.

**📊 4. Eğitim ve Değerlendirme**

* Loss: BCELoss veya DiceLoss
* Metrikler: IoU, Dice Coefficient
* Epoch başına görselleştirme: tahmin maskesi vs. gerçek maskeyi yan yana koy

**🔮 5. Tahmin Sonrası Kullanım**

* Her yeni görüntü için model → su maskesi üretir
* Bu maskeden su alanı hesaplanır → zaman serisi oluşturulur
* Web panelde hem harita hem grafik gösterilir

İstersen mock veriyle bir eğitim döngüsü kurabiliriz. Ya da senin elindeki .tif dosyalarıyla bir pipeline başlatabiliriz. Kod, görselleştirme, ya da veri eşleştirme kısmında hangisinden başlamak istersin?

kafam akrıstı bu görüntülerle nasıl bir yol ilerlemeliyim

Anlıyorum Lütfiye, bu kadar çok veri ve yöntem arasında kafa karışıklığı çok doğal. Senin gibi sistematik ve vizyoner biri için en iyisi, büyük resmi sadeleştirip adım adım ilerlemek. AquaTrack projesinin hem **su miktarı** hem de **su kalitesi** bileşeni var. Görüntülerin “images” ve “NDWI masks” olarak ayrılmış olması, seni doğrudan segmentasyon tabanlı bir modelleme sürecine götürüyor. Hadi bu süreci netleştirelim:

**🧭 Yol Haritası: Görüntülerle Ne Yapmalısın?**

**1. Veri Hazırlığı**

* images: Sentinel-2 ham görüntüler (RGB veya bandlar)
* NDWI masks: Su alanlarını gösteren binary maskeler (su: 1, diğer: 0)

Bu yapı, U-Net gibi bir segmentasyon modelini eğitmek için ideal. Her görüntüye karşılık gelen bir maske var → bu eşleşme üzerinden model öğrenebilir.

**2. Modelleme Seçimi**

| **Hedef** | **Yöntem** | **Veri Girişi** | **Çıktı** |
| --- | --- | --- | --- |
| Su alanı segmentasyonu | U-Net | Görüntü (image) | NDWI maskesi |
| Su miktarı tahmini | Regresyon | Görüntü + tarih | Alan (km²) |
| Su kalitesi tahmini | Transfer Learning | Görüntü | Klorofil-a, bulanıklık vs. |

Senin şu anki veri yapın, **U-Net ile su alanı segmentasyonu** için hazır. Bu model sana:

* Her görüntü için otomatik su maskesi üretir
* Bu maskeden su alanı hesaplanır
* Zaman serisi çıkarılır → kuraklık analizi yapılır

**3. Adım Adım Uygulama Planı**

**🔹 A. Veri Pipeline**

* Görüntüleri normalize et (0–1 arası)
* Maskeleri binary hale getir
* Boyutları sabitle (örneğin 256x256)

**🔹 B. Model Eğitimi**

* U-Net mimarisi kur
* BCELoss veya DiceLoss ile eğit
* Train/Test/Val ayrımı yap

**🔹 C. Tahmin ve Zaman Serisi**

* Yeni görüntü → model → tahmin maskesi
* Maskeden su alanı hesapla
* Tarih bazlı zaman serisi oluştur

**🔹 D. Görselleştirme**

* Harita üzerinde tahmin maskesini göster
* Zaman serisi grafiği ile birlikte sun

**🎯 Alternatif Yol: Sayısal NDWI ile Regresyon**

Eğer görüntüleri doğrudan modellemek istemezsen, NDWI değerlerini çıkarıp regresyon modeli kurabilirsin. Bu daha yorumlanabilir ama mekânsal detayları kaybettirir.

**🔄 Karar Noktan**

Senin veri yapın ve proje hedefin göz önüne alındığında:

✅ **U-Net ile segmentasyon → su alanı çıkarımı → zaman serisi → erken uyarı sistemi** ✅ **Transfer Learning ile kalite tahmini (ileriki aşama)**

İstersen mock veriyle bir U-Net pipeline’ı kurabiliriz. Ya da elindeki görüntüleri birlikte organize edip model eğitimine hazırlayabiliriz. Hangisinden başlamak seni rahatlatır: dosya yapısı mı, kod örneği mi, model mimarisi mi?

Bunun için resimlerle eğitmemize gerek yokmuş Resimlerden özellik çıkarıp regresyon modeli oluşturabilir mişiz Ama çıktı olarak sudaki değişimi görsel üzerinden gösteren bir model yapmak istersek bizim için daha zor olacak yapması Bence önce regresyonu deneyip sonrasında kalan zamanımızda da görsel çıktı olana uğraşırız Su kalitesi için de araştırıcam napabileceğimizi Veya yine bir kişi regresyona uğraşırken diğeri de görsel olana bakabilir bun e bununla

Bu yaklaşımın hem stratejik hem de gerçekçi Lütfiye 👏. AquaTrack’in iki ana bileşenini (su miktarı ve su kalitesi) düşününce, önce **sayısal çıkarım + regresyon** ile başlamak, sonra **görsel çıktı üreten model**e geçmek hem zaman yönetimi hem de ekip iş bölümü açısından çok mantıklı. Hadi bu planı netleştirelim:

**🧠 1. Regresyon Modeliyle Başlangıç**

**🎯 Amaç:**

NDWI gibi görüntüden çıkarılan metriklerle **su alanı (km²)** tahmini yapmak.

**🔧 Veri Yapısı:**

| **Tarih** | **NDWI Ortalaması** | **Su Alanı (km²)** |
| --- | --- | --- |
| 2018-05 | 0.42 | 123.4 |
| 2018-06 | 0.39 | 120.1 |
| ... | ... | ... |

**📈 Model Önerisi:**

* Basit regresyon (Linear, Ridge)
* Zaman serisi regresyon (ARIMA, Prophet)
* Feature engineering ile mevsimsel etkileri dahil etme

**✅ Avantaj:**

* Hızlı uygulanabilir
* Görselleştirmesi kolay
* Web panelde grafik olarak sunulabilir

**🎨 2. Görsel Çıktı Üreten Model (Segmentasyon)**

**🎯 Amaç:**

Görüntü → model → su maskesi → görsel çıktı

**🔧 Veri Yapısı:**

* images/: Sentinel-2 görüntüleri
* masks/: NDWI ile oluşturulmuş su maskeleri

**🧬 Model Önerisi:**

* U-Net (temel segmentasyon)
* Attention U-Net (daha hassas alanlar için)
* Transfer Learning ile önceden eğitilmiş backbone kullanımı

**❗ Zorluklar:**

* GPU ihtiyacı
* Veri boyutu ve etiketleme
* Eğitim süresi

**✅ Avantaj:**

* Harita üzerinde görsel çıktı sunar
* Mekânsal örüntüleri öğrenir
* Web panelde interaktif harita olarak gösterilebilir

**🔄 3. Ekip İş Bölümü Önerisi**

| **Kişi** | **Görev** |
| --- | --- |
| Lütfiye | Regresyon modeli → NDWI zaman serisi → grafik çıktı |
| Beyza | Görsel model → U-Net eğitimi → su maskesi üretimi |
| Serap | Su kalitesi araştırması → klorofil-a, bulanıklık tahmini için yöntem seçimi |

Bu yapı hem paralel ilerlemeyi sağlar hem de proje sonunda entegre bir sistem sunmanıza olanak verir.

**🔍 4. Su Kalitesi için İlk Adımlar**

* **Klorofil-a tahmini** için: GCI (Green Chlorophyll Index), FAI (Floating Algae Index)
* **Bulanıklık için**: Band oranları (örneğin B3/B11)
* **Modelleme**: Random Forest, XGBoost veya CNN tabanlı tahmin

İstersen bu kısmı birlikte araştırabiliriz. Literatür taraması, örnek indeks hesaplamaları ve veri kaynaklarıyla başlayabiliriz.

Hazırsan regresyon için mock veriyle bir örnek model kurabiliriz. Ya da görsel model için dosya yapısını birlikte organize edebiliriz. Hangisinden başlamak seni rahatlatır?

Bugün

bana sunum hazırlarmısın nasıl yollardan ilerledik ilerliyoruz bir hocaya sunsam

Tabii Lütfiye! Aşağıda AquaTrack projenizin ilerleyişini sade, etkileyici ve teknik olarak güçlü bir şekilde anlatan bir sunum taslağı hazırladım. Hem akademik hem proje odaklı bir hocaya sunulabilecek şekilde yapılandırdım. İstersen bunu PowerPoint’e dönüştürüp görsellerle de zenginleştirebiliriz.

**🎓 AquaTrack Proje Sunumu**

**Su Miktarı ve Kalitesi Takip Sistemi**

**🧭 1. Proje Amacı**

* Türkiye’deki göl ve barajların **su miktarı** ve **su kalitesi** değişimlerini izlemek
* Uydu görüntüleriyle **otomatik, düşük maliyetli ve geniş alanlı** izleme altyapısı kurmak
* Karar vericilere **erken uyarı sistemi** sunmak
* Tarım, içme suyu ve çevre sağlığı için **veri temelli yönetim** sağlamak

**🧩 2. Proje Bileşenleri**

| **Bileşen** | **Açıklama** |
| --- | --- |
| Su Miktarı Takibi | NDWI ile su alanı çıkarımı, zaman serisi analizi |
| Su Kalitesi Takibi | Klorofil-a, bulanıklık gibi parametrelerin tahmini |
| Web Panel | Harita ve grafiklerle görselleştirme, kullanıcı dostu arayüz |

**🛠️ 3. Teknik Yöntemler**

* **Uydu Verisi**: Sentinel-2 (Google Earth Engine üzerinden)
* **Öznitelik Çıkarımı**: NDWI, MNDWI, GCI, FAI
* **Modelleme**:
  + Regresyon (su miktarı tahmini)
  + U-Net (görsel su maskesi üretimi)
  + Transfer Learning (su kalitesi tahmini)
* **Veri Yönetimi**: Git, Python, Rasterio, Scikit-learn, PyTorch

**📈 4. Şu Ana Kadar İlerleme**

✅ Uydu görüntüleri yıllara göre indirildi ve organize edildi ✅ NDWI maskeleri oluşturuldu ✅ Görüntüler “images” ve “masks” olarak ayrıldı ✅ Regresyon modeli için veri seti hazırlandı ✅ Görsel model için dosya yapısı kuruldu ✅ Web panel için görselleştirme altyapısı planlandı

**🔄 5. Stratejik Yol Haritası**

**A. İlk Aşama:**

**Sayısal çıkarım + regresyon modeli** → NDWI → su alanı → zaman serisi → grafiksel çıktı

**B. İkinci Aşama:**

**Görsel çıktı üreten model (U-Net)** → Görüntü → su maskesi → harita üzerinde görselleştirme

**C. Üçüncü Aşama:**

**Su kalitesi tahmini** → Klorofil-a, bulanıklık gibi parametreler için indeks ve model araştırması

**👥 Ekip İş Bölümü**

| **Üye** | **Görev** |
| --- | --- |
| Lütfiye | Regresyon modeli, zaman serisi analizi |
| Beyza | Görsel model (U-Net), veri eşleştirme |
| Serap | Su kalitesi parametreleri araştırması |

**🌍 Hedef Kitle**

* DSİ, Çevre Bakanlığı, Belediyeler
* OSB’ler, turizm ve tarım işletmeleri
* Üniversiteler ve AR-GE merkezleri

**📊 Beklenen Çıktılar**

* Harita tabanlı su miktarı ve kalite görselleştirmesi
* Zaman serisi grafiklerle kuraklık analizi
* Web panel üzerinden kullanıcıya özel veri erişimi
* Karar vericilere erken uyarı ve planlama desteği

**🚀 Sonraki Adımlar**

* Regresyon modelinin eğitimi ve doğrulama
* Görsel modelin mock veriyle test edilmesi
* Su kalitesi için indeks ve literatür taraması
* Web panelin prototipinin oluşturulması

İstersen bu sunumu PowerPoint formatında görselleştirebiliriz: harita örnekleri, grafikler, veri akış şeması gibi öğelerle zenginleştirebilirim. Hangi formatta istersin: metin, slayt başlıkları, ya da doğrudan sunum dosyası önerisi?

evet

Harika Lütfiye! Elinizdeki başvuru dosyasını inceledim ve AquaTrack projesinin kapsamı, teknik derinliği ve toplumsal etkisi gerçekten çok güçlü. Şimdi bu içeriği bir hocaya sunulabilecek şekilde sade, etkileyici ve teknik olarak dengeli bir sunum haline getiriyorum. Aşağıda slayt başlıklarıyla birlikte içerik önerisini bulabilirsin:

**🎓 AquaTrack Sunumu**

**Su Miktarı ve Kalitesi Takip Sistemi**

**1️⃣ Giriş: Neden Bu Proje?**

* Türkiye’de artan **kuraklık ve kirlilik**
* Mevcut ölçüm yöntemleri: **manuel, noktasal, yetersiz**
* AquaTrack: **otomatik, geniş alanlı, düşük maliyetli** izleme çözümü

**2️⃣ Proje Amacı**

* Tatlı su kaynaklarının **miktar ve kalite** takibini sağlamak
* Karar vericilere **erken uyarı sistemi** sunmak
* **Çevresel sürdürülebilirlik** ve **iklim uyumu** kapasitesini artırmak

**3️⃣ Teknik Yaklaşım**

| **Bileşen** | **Yöntem** | **Araçlar** |
| --- | --- | --- |
| Su Miktarı | NDWI, MNDWI | Sentinel-2, GEE, Python |
| Su Kalitesi | Klorofil-a, bulanıklık | Transfer Learning, PyTorch |
| Görselleştirme | Harita & grafik | Web panel, CBS entegrasyonu |

**4️⃣ Modelleme Süreci**

* **U-Net** ile su alanı segmentasyonu
* **Transfer Learning** ile kalite tahmini
* **Veri kaynakları**: Copernicus, CMEMS, GEE
* **Metrikler**: IoU, Dice, Accuracy, F1-Score

**5️⃣ Web Panel Özellikleri**

* Harita üzerinden bölge seçimi
* Zaman serisi grafiklerle analiz
* Mobil uyumlu, kullanıcı dostu arayüz
* Geri bildirim modülü ile dinamik gelişim

**6️⃣ Hedef Kitle**

* DSİ, Çevre Bakanlığı, Belediyeler
* OSB’ler, turizm ve tarım işletmeleri
* Üniversiteler ve AR-GE merkezleri

**7️⃣ Rekabet Analizi**

| **Rakip** | **Odak** | **AquaTrack Farkı** |
| --- | --- | --- |
| Water Insight | Su kalitesi | AquaTrack: kalite + miktar |
| EOS Analytics | Tarım | AquaTrack: doğrudan su kaynakları |
| USBS | Veri arşivi | AquaTrack: karar destek + AI modeli |

**8️⃣ Ticari Potansiyel**

* Türkiye su hizmetleri pazarı: **4.5 milyar €**
* Küresel büyüme: **%7.2 CAGR**
* SaaS modeli: abonelik + lisans + ESG raporlama

**9️⃣ Proje Yönetimi**

* **Agile yaklaşım**: esnek ve hızlı müdahale
* Gantt şeması ile zaman yönetimi
* Haftalık görev takibi ve dış paydaş iş birlikleri

**🔟 Etki ve Sürdürülebilirlik**

* Su krizlerine karşı **erken müdahale**
* Tarım ve turizmde **verimli planlama**
* SKA hedefleriyle **doğrudan uyum**
* from sklearn.linear\_model import LinearRegression
* from sklearn.metrics import mean\_squared\_error, r2\_score
* import numpy as np
* # 1. Tarihi sayısal forma çevir (ordinal)
* df\_timeseries["date\_ordinal"] = df\_timeseries["date"].map(datetime.toordinal)
* # 2. Modeli eğit
* X = df\_timeseries[["date\_ordinal"]].values
* y = df\_timeseries["area\_m2"].values
* model = LinearRegression()
* model.fit(X, y)
* # 3. Tahmin ve performans
* y\_pred = model.predict(X)
* mse = mean\_squared\_error(y, y\_pred)
* r2 = r2\_score(y, y\_pred)
* print(f"Model MSE: {mse:.2f}")
* print(f"Model R²: {r2:.3f}")
* # 4. Görselleştirme
* plt.figure(figsize=(12, 6))
* plt.scatter(df\_timeseries["date"], y, label="Gerçek", color="teal")
* plt.plot(df\_timeseries["date"], y\_pred, label="Tahmin (Regresyon)", color="orange")
* plt.title("Su Alanı Regresyon Tahmini", fontsize=16)
* plt.xlabel("Tarih", fontsize=14)
* plt.ylabel("Su Alanı (m²)", fontsize=14)
* plt.legend()
* plt.grid(True)
* plt.tight\_layout()
* plt.show()
* # 5. Gelecek tarihler için tahmin üret (örnek: 2025 sonuna kadar her ay)
* future\_dates = pd.date\_range(start="2025-01-01", end="2025-12-01", freq="MS")
* future\_ordinals = future\_dates.map(datetime.toordinal).values.reshape(-1, 1)
* future\_preds = model.predict(future\_ordinals)
* # 6. Görselleştirme: geçmiş + tahmin
* plt.figure(figsize=(12, 6))
* plt.plot(df\_timeseries["date"], y, label="Gerçek", marker="o", color="teal")
* plt.plot(df\_timeseries["date"], y\_pred, label="Model Tahmini", color="orange")
* plt.plot(future\_dates, future\_preds, label="2025 Tahmini", linestyle="--", color="red")
* plt.title("Van Gölü Su Alanı Tahmini (Regresyon)", fontsize=16)
* plt.xlabel("Tarih", fontsize=14)
* plt.ylabel("Su Alanı (m²)", fontsize=14)
* plt.legend()
* plt.grid(True)
* plt.tight\_layout()
* plt.show()
* # Tahminleri CSV olarak dışa aktar
* future\_df = pd.DataFrame({
* "date": future\_dates,
* "predicted\_area\_m2": future\_preds
* })
* future\_df.to\_csv("van\_golu\_2025\_tahmin.csv", index=False)