

VERİ BİLİMİ DERSİ

# MLOps ve Capstone

*Hafta 9 · Modül 9*  
*Model Deployment, Streamlit ve Proje Yönetimi*

**Dr. Murat Altun**

Veri Bilimi ve Yapay Zekâ Eğitimi · 2026

6

Saat

3

Notebook

4

Proje Önerisi

# İçindekiler

01

## MLOps Temelleri

ML yaşam döngüsü · model serialization · joblib

Slayt 3-6

02

## Deployment & Araçlar

Streamlit · Gradio · Hugging Face Spaces · deploy adımları

Slayt 7-12

03

## Capstone Projesi

Proje yönetimi · portfolio · Airbnb · Trendyol · Spotify

Slayt 13-20

## ML + DevOps = MLOps

MLOps, makine öğrenmesi modellerinin geliştirme, eğitim, deploy ve izleme süreçlerini otomatikleştiren mühendislik disiplindir. DevOps prensiplerini ML yaşam döngüsüne uygular.

## Neden Önemli?

- 1 Modellerin %87'si üretime geçemiyor (Gartner)
- 2 Tekrarlanabilirlik ve versiyon kontrolü sağlar
- 3 Model bozulmasını (drift) erken tespit eder
- 4 Ekip iş birliğini standartlaştırır

**%87**

Model  
Üretime Geçemez

**2X**

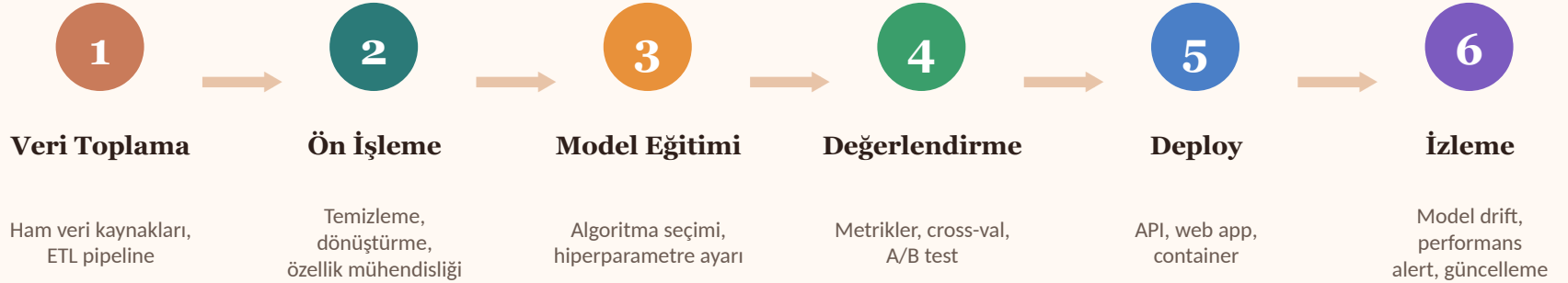
Hızlı  
Deploy Döngüsü

**10X**

Daha Az  
Manuel İş

**CI/CD**

Sürekli  
Entegrasyon



♻ *Sürekli Güncelleme Döngüsü* ♻

## MLOps'un Temel İlkesi

Bir model bir kez eğitilip bırakılmaz; sürekli izlenir, değerlendirilir ve güncellenir. Veri değiştiğinde model de değişmelidir. Bu döngüyü otomatikleştirmek MLOps'un özüdür.

## joblib ile Model Kaydetme

```
import joblib
# Modeli kaydet
joblib.dump(model, 'model.joblib')

# Modeli yükle
model = joblib.load('model.joblib')
```

## Alternatif: JSON / ONNX Formatı

```
# ONNX ile model export
import skl2onnx
onnx_model = skl2onnx.convert(
    model, initial_types=...
)
```

## Serialization Yöntemleri Karşılaştırma

Özellik	joblib	ONNX
NumPy Desteği	Mükemmel ✓	Var
Büyük Dosya	Sıkıştırma var	Kompakt
Hız	Hızlı	Çok Hızlı
Güvenlik	Güvenilir kaynak gerekli	Güvenli
Kullanım	ML modelleri	Cross-platform

## Python ile Web Uygulama Geliştirme

Streamlit, Python ile interaktif web uygulamaları oluşturmanızı sağlayan açık kaynak bir framework'tür. HTML/CSS/JS bilgisi gerektirmez. Tek bir Python dosyasıyla tam işlevsel bir dashboard oluşturabilirsiniz.

### Tek Dosya

Tüm uygulama tek bir  
Python dosyasında

### Hızlı Prototip

Dakikalar içinde  
çalışan demo

### Ücretsiz Deploy

Streamlit Cloud  
ile bedava yayınlama

### ML Entegrasyonu

scikit-learn, TF,  
PyTorch ile uyumlu

Bileşen	Açıklama	Kullanım
<code>st.title()</code>	Sayfa başlığı	<code>st.title('Başlık')</code>
<code>st.write()</code>	Metin, tablo, grafik yazdırma	<code>st.write(df)</code>
<code>st.slider()</code>	Kaydırıcı widget	<code>st.slider('Yaş', 0, 100)</code>
<code>st.selectbox()</code>	Açılır menü	<code>st.selectbox('Şehir', liste)</code>
<code>st.button()</code>	Buton widget	<code>st.button('Tahmin Et')</code>
<code>st.dataframe()</code>	DataFrame tablosu	<code>st.dataframe(df)</code>
<code>st.metric()</code>	KPI metrik kartı	<code>st.metric('Doğruluk', '%95')</code>

## Tam Bir Streamlit ML Uygulaması

```
import streamlit as st
import joblib
import numpy as np

# Modeli yükle
model = joblib.load('model.joblib')

st.title('Fiyat Tahmin Uygulaması')

# Kullanıcı girdileri
alan = st.slider('Alan (m2)', 30, 300, 100)
oda = st.selectbox('Oda Sayısı', [1,2,3,4,5])

if st.button('Tahmin Et'):
    X = np.array([[alan, oda]])
    fiyat = model.predict(X)[0]
    st.metric('Tahmini Fiyat', f'{fiyat:,.0f} TL')
```

## Streamlit

- ✓ Genel amaçlı dashboard
- ✓ Çoklu sayfa desteği
- ✓ Zengin widget kütüphanesi
- ✓ Streamlit Cloud ile deploy
- ✓ Daha fazla özelleştirme
- ✓ Büyük topluluk

## Gradio

- ✓ ML modeli odaklı arayüz
- ✓ 3 satır kod ile demo
- ✓ Otomatik API endpoint
- ✓ Hugging Face Spaces entegrasyonu
- ✓ Input/Output tabanlı tasarım
- ✓ Hızlı prototipleme

Dashboard istiyorsanız Streamlit, hızlı ML demo istiyorsanız Gradio tercih edin.

## HF Spaces Nedir?

Hugging Face Spaces, Streamlit veya Gradio uygulamalarınızı ücretsiz olarak barındıran bir platformdur. GitHub benzeri repo yapısıyla çalışır, GPU desteği sunar.

## Neden Kullanılır?

- 1 Ücretsiz hosting (2 vCPU, 16 GB RAM)
- 2 Git tabanlı versiyon kontrolü
- 3 Streamlit & Gradio native desteği
- 4 Docker container desteği
- 5 GPU erişimi (ücretli plan)

300K+

Space  
Paylaşıldı

0 €

Ücretsiz  
Hosting

GPU

A10G / T4 ile Hızlandırma

## İpucu

[huggingface.co/spaces](https://huggingface.co/spaces) adresinden binlerce uygulama inceleyerek ilham alabilirsiniz.

1

## Repo Oluştur

huggingface.co → New Space  
Streamlit veya Gradio seçin

2

## requirements.txt

Bağımlılıkları listeleyin:  
streamlit, joblib, scikit-learn, pandas

3

## app.py Hazırla

Streamlit/Gradio uygulamanızı  
tek dosyada yazın

4

## Git Push

git add . && git commit  
git push → Otomatik deploy!

## İyi Bir README Rehberi

- 1 Proje Başlığı & Açıklama** Ne yapar? Neden önemli?
- 2 Kurulum** pip install, gereksinimler
- 3 Kullanım** Örnek kod, ekran görüntüleri
- 4 Veri Seti** Kaynak, boyut, özellikler
- 5 Model & Metrikler** Algoritma, accuracy, F1 skoru
- 6 Demo Linki** HF Spaces / Streamlit Cloud URL
- 7 Lisans & İletişim** MIT, Apache, katkıda bulunma

## Portfolio Hazırlama İpuçları

- Her proje için ayrı repo açın
- README.md'de görsel kullanın
- Live demo linki mutlaka ekleyin
- Veri temizleme sürecini belgeleyin
- Model seçim gerekçesini açıklayın
- Sonuçları görsellerle destekleyin
- GitHub Profile README oluşturun

## Pinned Repositories

En iyi 6 projenizi pin'leyin. Çeşitlilik gösterin: EDA, ML, NLP, Deep Learning.

## README Kalitesi

Her repoda detaylı README: problem tanımı, veri seti, model, sonuçlar ve demo linki.

## Temiz Kod

PEP8 uyumlu, docstring'li, modüler yapıda kod. Jupyter Notebook + .py dosyaları birlikte.

## Contribution Graph

Düzenli commit'ler yeşil kare grafiğini doldurur. Süreklilik önemli, miktar değil.

## Proje Tanımı

Airbnb listelerinin özelliklerinden (konum, oda tipi, yorum sayısı, minimum gece) gecelik fiyatı tahmin eden bir regresyon modeli geliştirin.

48K+

Kayıt  
(İstanbul)

16+

Özellik  
Kolonu

## Veri Seti Özellikleri

Özellik	Açıklama
neighbourhood	Mahalle / bölge bilgisi
room_type	Tüm ev, özel oda, paylaşımlı
minimum_nights	Minimum konaklama süresi
number_of_reviews	Toplam yorum sayısı
availability_365	Yıllık müsaitlik günü
price (hedef)	Gecelik fiyat (USD)

## Beklenen Çıktılar

- 1 EDA + görselleştirme
- 2 Feature engineering
- 3 3+ model karşılaştırma
- 4 En iyi modeli deploy
- 5 Streamlit / Gradio demo
- 6 README + sunum

## Proje Tanımı: Sentiment Analizi + NLP + ML

Trendyol ürün yorumlarını pozitif/negatif olarak sınıflandıran bir NLP modeli. Metin ön işleme, TF-IDF vektörizasyonu ve makine öğrenmesi sınıflandırıcıları kullanarak duygu analizi yapın.

1

### Veri Toplama

Web scraping veya hazır veri seti

2

### Metin Temizleme

Küçük harf, noktalama, stopword çıkarma

3

### Vektörizasyon

TF-IDF veya CountVectorizer

4

### Model Eğitimi

Logistic Reg, SVM, Random Forest

5

### Değerlendirme

Confusion matrix, F1-score, ROC-AUC

**50K+**  
Yorum Verisi

**%90+**  
Hedef Doğruluk

**TF-IDF**  
Vektörizasyon

**NLP**  
Doğal Dil İşleme

## Şarkı Özellikleri → Popülerlik Skoru

Spotify API'dan alınan şarkı özellikleri (danceability, energy, tempo, loudness vb.) kullanarak popülerlik skorunu tahmin eden bir regresyon modeli geliştirin.

### danceability

Dans edilebilirlik  
(0.0 - 1.0)

### energy

Enerji seviyesi  
(0.0 - 1.0)

### tempo

BPM (beats per  
minute)

### loudness

Ses yüksekliği  
(dB)

### valence

Mutluluk skoru  
(0.0 - 1.0)

### speechiness

Konuşma oranı  
(0.0 - 1.0)

170K+

Şarkı  
Verisi

0-100

Popülerlik  
Skoru

## Proje Adımları

- 1 Spotify API ile veri çekme
- 2 Korelasyon analizi + EDA
- 3 Feature selection
- 4 Linear Reg, RF, XGBoost
- 5 Streamlit ile demo
- 6 HF Spaces'e deploy

%40

## Teknik Kalite

- Doğru algoritma seçimi
- Temiz ve modüler kod
- Model performans metrikleri
- Hata analizi ve iyileştirme

%20

## Sunum & Dokümantasyon

- README kalitesi
- Görselleştirme zenginliği
- Sunum akışı
- Sonuçların yorumlanması

%20

## Yenilikçilik

- Özgün yaklaşım
- Ek veri kaynakları
- Yaratıcı feature engineering
- Farklı model denemeleri

%20

## Deploy & Demo

- Çalışan web uygulaması
- Kullanıcı dostu arayüz
- HF Spaces veya Streamlit Cloud
- API endpoint (bonus)

## mlops\_pipeline.ipynb

Model eğitimi, joblib ile kaydetme/yükleme, basit pipeline oluşturma. sklearn Pipeline + ColumnTransformer kullanımı.

joblib

Pipeline

ColumnTransformer

cross\_val\_score

## streamlit\_app.py

Eğitilmiş modeli Streamlit ile web uygulamasına dönüştürme. Kullanıcıdan input alma ve tahmin gösterme.

st.slider

st.selectbox

st.metric

model.predict

## capstone\_sablon.ipynb

Capstone proje şablonu: veri yükleme, EDA, model eğitimi, değerlendirme ve raporlama bölümleri hazır.

Proje yapısı

EDA şablonu

Model karşılaştırma

Raporlama

## Capstone Proje Ödevi

### 1 Proje konusu seçin

Airbnb, Trendyol veya Spotify'dan birini seçin (veya kendi konunuzu önerin)

### 2 Proje planı hazırlayın

Veri kaynağı, hedef değişken, kullanılacak algoritmalar, zaman çizelgesi

### 3 Veri setini hazırlayın

Veriyi indirin, ilk EDA'yı yapın, temizleme stratejisi belirleyin

### 4 GitHub repo açın

README.md şablonunu doldurun, .gitignore ekleyin, ilk commit

**Teslim: Hafta 10 ders başlangıcı**

## Faydalı Kaynaklar

Streamlit Dokümantasyon

[docs.streamlit.io](https://docs.streamlit.io)

Gradio Dokümantasyon

[gradio.app/docs](https://gradio.app/docs)

Hugging Face Spaces

[huggingface.co/spaces](https://huggingface.co/spaces)

Kaggle Datasets

[kaggle.com/datasets](https://kaggle.com/datasets)

Inside Airbnb

[insideairbnb.com](https://insideairbnb.com)

Spotify Web API

[developer.spotify.com](https://developer.spotify.com)

MLOps Principles

[ml-ops.org](https://ml-ops.org)

# Hafta 9 Özet

1 MLOps, modelin üretim ortamına geçişini ve yaşam döngüsünü yönetir

2 joblib ile model serialization, Streamlit ile interaktif demo oluşturma

3 Hugging Face Spaces ücretsiz deploy platformu olarak güçlü bir seçenek

4 Capstone projesi tüm öğrenilenleri tek bir projede birleştirir

5 İyi bir GitHub portfolio, iş başvurularında fark yaratır

*“En iyi model üretimde çalışan modeldir.”*