Makine Öğrenmesi Sistemleri ve MLOps

Yapay Zekâ ve Veri Dersleri Rehberi · Kısım VI: Sistem ve Uygulama

Yaşam döngüsü | Sürümleme | Dağıtım | İzleme ve kayma | Otomasyon | Komşu dersler

Sercan KÜLCÜ · Giresun Üniversitesi · Bilgisayar Mühendisliği

© Sercan Külcü. Tüm hakları saklıdır. · Sunumun etkileşimli HTML sürümü: https://sercankulcu.github.io/files/ai_courses/sunumlar/Bolum_35_MLOps.html

GİRİŞ

Ders Hangi Soruyu Sorar?

> **Bir model laboratuvardan üretime nasıl taşınır ve orada nasıl ayakta tutulur?**

MLOps, makine öğrenmesi modellerini üretime taşıyıp orada güvenilir biçimde çalıştırmak için yazılım mühendisliği ve DevOps alışkanlıklarını MÖ'ye uygular. Ders veriyi, kodu ve modeli birlikte sürümlemeyi, **boru hattını** otomatikleştirmeyi, dağıtmayı ve üretimdeki modeli izlemeyi inceler.

Bir bakışta

YAKLAŞIM yazılım mühendisliği ve MÖ	**GİRDİ** eğitilmiş model, veri boru hattı	**ÇIKTI** çalışan ve izlenen hizmet
BAŞARI ÖLÇÜTÜ gecikme, kesintisizlik, veri kayması	**MATEMATİK YÜKÜ** düşük	**VERİ İHTİYACI** akan üretim verisi

Kısa Tarihçe

2009
İlk DevOpsDays Ghent'te yapılır; geliştirme ve işletmeyi birleştiren DevOps akımı yayılır.

2014
Docker 1.0 yayımlanır, Kubernetes duyurulur: kapsayıcılar dağıtımın ortak birimi olur.

2015
Sculley ve ark. MÖ sistemlerindeki gizli teknik borcu gösterir: model kodu sistemin küçük bir parçasıdır.

2017
Google TFX'i, Uber Michelangelo'yu tanıtır; şirket içi MÖ platformları görünür olur.

2018
Databricks MLflow'u açık kaynak olarak duyurur: deney takibi ve model kaydı.

2021
Andrew Ng veri merkezli YZ çağrısını yapar: modelden çok veri kalitesine odaklanmak.

DERS

Temel Kavramlar

Boru hattı
Veri hazırlamadan dağıtıma kadar otomatik adımlar.

Sürümleme
Veri, kod ve modelin birlikte izlenmesi ve geri alınması.

Deney takibi
Her eğitimin parametre, ölçüt ve çıktısını kaydetmek.

Model kaydı
Onaylı model sürümlerinin tek bir yerde tutulması.

Öznitelik deposu
Eğitim ve üretimde aynı özniteliklerin kullanılması.

CI/CD/CT
Sürekli tümleştirme, dağıtım ve sürekli yeniden eğitim.

Model sunma
Çevrimiçi API ya da toplu tahmin işi olarak çalıştırma.

Veri kayması
Üretim verisinin eğitimdekinden uzaklaşması; izlenmeli.

DERS

Tipik İş Akışı



Örnek: Bir bankanın kredi modeli her ay yeni başvurularla yeniden eğitilir. Kayda geçen en iyi sürüm bir kapsayıcıya paketlenir, önce trafiğin küçük bir kısmına açılır, sorun çıkmazsa herkese yayılır. Başvuru dağılımı kaydığında izleme panosu ekibi uyarır ve döngü yeniden başlar.

DERS

Derste Neler İşlenir?

1. Üretimdeki MÖ sistemi: gereksinimler ve teknik borç
2. Veri boru hatları, veri doğrulama ve sürümleme
3. Öznitelik mühendisliği ve öznitelik depoları
4. Deney takibi ve yeniden üretilebilir eğitim
5. Model değerlendirme: dilim analizi ve model testleri
6. Paketleme: kapsayıcılar ve model kaydı
7. Model sunma: çevrimiçi API ve toplu tahmin
8. Çıkarımı hızlandırma: gecikme, ölçek ve maliyet
9. Dağıtım stratejileri: kanarya, gölge, A/B testi
10. İzleme: veri kayması, kavram kayması, uyarılar
11. Sürekli eğitim ve CI/CD boru hatları
12. Sorumlu MÖ: denetim, belgeleme, model kartları

Kazandırdığı beceriler

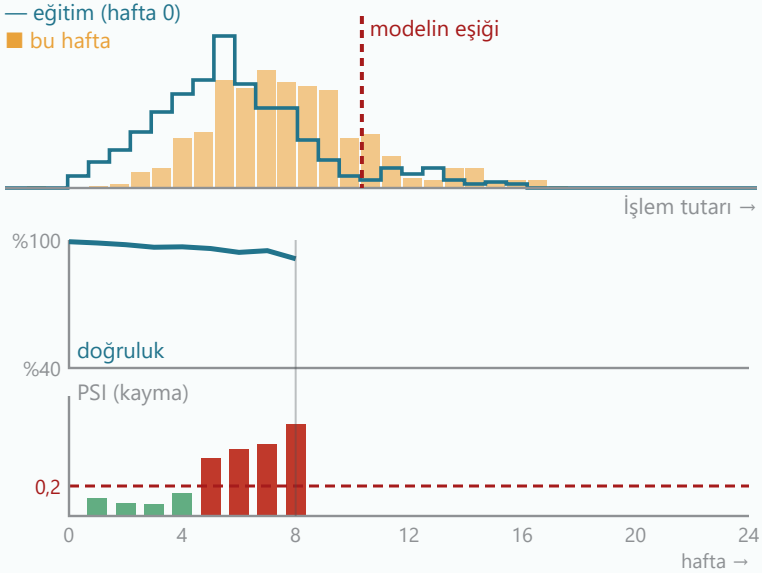
Git ve DVC ile sürümleme | MLflow ile deney takibi | Docker ile model paketlemek | modeli REST API olarak sunmak | Airflow ile boru hattı kurmak | Prometheus ile izleme panosu

Tipik bir dönemin akışıdır; sıra ve kapsam üniversiteden üniversiteye değişir.

DENE

Model Üretimde Sessizce Bozuluyor

Bankanın sahtecilik modeli olağan dışı büyük işlemleri işaretliyor; fiyatlar her hafta biraz artıyor. Haftayı ilerletin, kaymayı (PSI) ve doğruluğu izleyin; alarm çalınca modeli yeniden eğitin.



Üretimdeki hafta: **8**

Yeniden eğit | Baştan al

Hafta 8: PSI = **0,61** → **ALARM**
Doğruluk **%91,2** (eğitildiği hafta %99,2)
Model hafta 0 verisiyle eğitildi.

PSI, bu haftanın dağılımını eğitim verisiyle on dilimde karşılaştırır. **0,2** üstü alarm sayılır. Etiketler geç gelir; PSI ise hemen hesaplanır.

Model yazılım gibi bir kez kurulup bırakılmaz: MLOps **izleme**, alarm ve otomatik yeniden eğitim hattını kurar. Kayma, doğruluk düşmeden önce yakalanır.

KONUM

Ön Koşullar

Bu dersten önce bilinmesi yararlı olan dersler ve neden gerektikleri.

Ders	Neden gerekir
Makine Öğrenmesi	modeli dağıtma, izleme, yeniden eğitme
İşletim Sistemleri	süreçler, bellek ve kaynak yönetimi: kapsayıcıların dayandığı kavramlar
Nesneye Yönelik Programlama	modüler ve test edilebilir kod yazma alışkanlığı
Yazılım Mühendisliği	sürüm denetimi (Git), test ve sürekli tümleştirme

KONUM

Bu Ders Sana Göre mi?

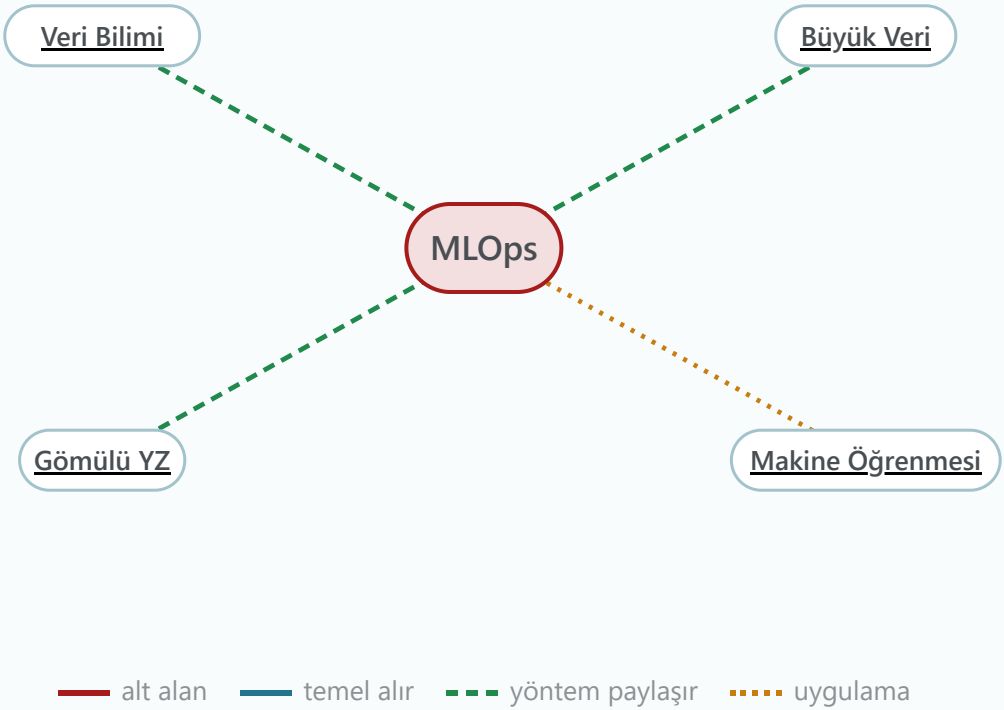
Kendine sor; "hayır" dediklerin için sağdaki sütun nereden başlayacağını gösterir.

Kendine sor	Hayırsa önce
Bir sınıflandırıcıyı eğitip çapraz doğrulamayla değerlendirebiliyor musun?	Makine Öğrenmesi
Git ile dal açıp birleştirebiliyor ve bir çakışmayı çözebiliyor musun?	Git'e kısa bir giriş ve bir yazılım mühendisliği dersi
Komut satırında süreçleri, ortam değişkenlerini ve izinleri yönetebiliyor musun?	İşletim sistemleri dersi ve Linux komut satırına giriş
Python'da modüllere ayrılmış, birim testleri olan bir proje yazabiliyor musun?	Nesneye yönelik programlama, ardından pytest'e kısa bir giriş

Çoğuna "evet" diyebiliyorsan derse hazırsın; bir ya da iki "hayır" dönem içinde kapanabilir.

KONUM

Dersin Komşuları



ÖRTÜŞME

Örtüştüğü Dersler

İlişki	Ders	Ortak konu
uygulama	Bu ders Makine Öğrenmesi yöntemlerini uygular	modeli dağıtma, izleme, yeniden eğitme
yöntem paylaşır	Veri Bilimi ile yöntem paylaşır	deney takibi, veri sürümleme
yöntem paylaşır	Büyük Veri Sistemleri ile yöntem paylaşır	veri boru hatları
yöntem paylaşır	Gömülü Yapay Zekâ ile yöntem paylaşır	modeli cihazlara dağıtma ve güncelleme

Ayrıştığı Noktalar (1/2)

Ölçüt	MLOps	Makine Öğrenmesi	Büyük Veri
Temel soru	Bir model laboratuvardan üretime nasıl taşınır ve orada nasıl ayakta tutulur?	Bir program örneklerden nasıl genelleme yapar?	Tek bir makineye sığmayan veri nasıl saklanır ve işlenir?
Yaklaşım	yazılım mühendisliği ve MÖ	istatistiksel öğrenme	dağıtık depolama ve hesaplama
Girdi	eğitilmiş model, veri boru hattı	etiketli ya da etiketsiz örnekler	çok büyük ve hızlı akan veri
Çıktı	çalışan ve izlenen hizmet	tahmin yapan model	işlenmiş veri, sorgu sonucu
Başarı ölçütü	gecikme, kesintisizlik, veri kayması	görülmemiş veride hata	ölçeklenme, gecikme, maliyet
Matematik yükü	düşük	yüksek: doğrusal cebir, olasılık, optimizasyon	düşük
Veri ihtiyacı	akan üretim verisi	orta ve çok	çok fazla

- **Makine Öğrenmesi:** MÖ iyi bir model kurmayı öğretir; bu ders o modelin üretimde aylarca güvenilir çalışmasını sağlamayı öğretir.
- **Büyük Veri:** Büyük veri sistemleri veriyi ölçekli saklar ve işler; bu ders o veriyle eğitilen modelin yaşam döngüsünü yönetir.

AYRIŞMA

Ayrıştığı Noktalar (2/2)

Ölçüt	MLOps	Veri Bilimi	Gömülü YZ
Temel soru	Bir model laboratuvardan üretime nasıl taşınır ve orada nasıl ayakta tutulur?	Veriden doğru soruları sorup karara dönüşen sonuçlar nasıl çıkarılır?	Yapay zekâ modelleri küçük, düşük güçlü cihazlarda nasıl çalışır?
Yaklaşım	yazılım mühendisliği ve MÖ	soru, veri, analiz, anlatı	kaynağı kısıtlı çıkarım
Girdi	eğitilmiş model, veri boru hattı	ham, dağınık veri	eğitilmiş model, cihaz kısıtları
Çıktı	çalışan ve izlenen hizmet	rapor, görselleştirme, model	küçük ve hızlı model
Başarı ölçütü	gecikme, kesintisizlik, veri kayması	kararlara katkı, doğruluk	gecikme, enerji, bellek
Matematik yükü	düşük	orta: istatistik	orta
Veri ihtiyacı	akan üretim verisi	orta ve çok	eğitim verisi zaten var

- **Veri Bilimi:** Veri bilimi bir soruyu yanıtlayan analiz yapar; bu ders sürekli çalışan, izlenen ve yenilenen bir hizmet kurar.
- **Gömülü YZ:** Gömülü YZ modeli küçük bir cihaza sığdırır; bu ders modeli sunucularda dağıtır, sürümler ve izler.

UYGULAMA

Uygulama Alanları

Öneri hizmetleri
Günde milyonlarca isteğe yanıt veren modeller.

Sahtekârlık tespiti
Milisaniyede karar, sık yeniden eğitim.

Talep tahmini
Her gece çalışan toplu tahmin işleri.

Arama sıralaması
A/B testiyle sürekli denenen modeller.

Kredi riski
Denetlenebilir, sürümlü ve izlenen modeller.

Dil modeli hizmetleri
Büyük modellerin ölçekli ve maliyetli sunumu.

UYGULAMA

Riskler ve Etik

Sessiz bozulma

Kayma izlenmezse model hata vermeden kötüleşir; COVID-19 döneminde birçok talep tahmin modeli bir anda isabetini yitirdi.

Eğitim-sunum çarpıklığı

Bir öznitelik eğitimde ve üretimde farklı hesaplanırsa testte iyi olan model canlıda yanlış tahmin üretir.

Geri besleme döngüsü

Model kendi kararlarının sonuçlarıyla yeniden eğitilir; öneri ya da kredi modeli kendi yanlılığını büyütür.

İzlenemeyen sürüm

Hangi veri ve kodla eğitildiği bilinmeyen model, bir şikâyette ya da denetimde açıklanamaz ve geri alınamaz.

Ayrıntılı tartışma: Açıklanabilir Yapay Zekâ ve Yapay Zekâ Etiği.

UYGULAMA

Araçlar ve Örnek Veriler

Araçlar ve kütüphaneler

MLflow | DVC | Docker | Kubernetes | Kubeflow | Apache Airflow | Prometheus

Klasik örnek senaryolar

Aylık yeniden eğitilen kredi modeli | Gece çalışan talep tahmini | Gerçek zamanlı sahtekârlık sunumu | A/B testli öneri modeli | Veri kayması benzetimi | Hatalı sürümü geri alma

Ders bir algoritma değil bir **süreç** öğretir: küçük bir model eğitimden izlemeye kadar uçtan uca bir boru hattına dönüştürülür.

UYGULAMA

Sık Karıştırılanlar

Yaygın düşünce	Doğrusu
Model eğitilince iş biter.	Üretimde veri değişir; izlenmeyen model zamanla sessizce kötüleşir.
MLOps yalnızca araç kurmaktır.	Araçlar süreci destekler; asıl iş sürümleme, test, sorumluluk ve geri alma düzenidir.
Test doğruluğu yüksekse üretimde de iyidir.	Üretim verisi eğitimdekinden farklı olabilir; gecikme ve maliyet de başarının parçasıdır.
Bir MÖ sisteminin çoğu modeldir.	Model kodu; veri, yapılandırma, sunum ve izleme altyapısı yanında küçük bir parçadır.

SONRASI

Bu Dersten Sonra

Bu dersi temel alan ya da genişleten dersler

Bu ders çoğunlukla uç noktadır; öteki dersleri temel alır.

İlgili meslekler

MLOps mühendisi | Makine öğrenmesi mühendisi | Veri mühendisi | Platform mühendisi | Site güvenilirlik mühendisi (SRE)

MLOps rehberin sistem kısmının omurgasıdır: büyük veri sistemleri eğitim verisini, GPU programlama eğitim hızını, gömülü YZ ise uçtaki dağıtımı ele alır. Bu dersten sonra bulut platformları, dağıtık eğitim ya da büyük dil modellerinin sunumu doğal sonraki adımlardır.

SONRASI

Kaynaklar

- Huyen, *Designing Machine Learning Systems*, O'Reilly, 2022.
- Burkov, *Machine Learning Engineering*, True Positive Inc., 2020.
- Lakshmanan, Robinson ve Munn, *Machine Learning Design Patterns*, O'Reilly, 2020.
- Sculley ve ark., "Hidden Technical Debt in Machine Learning Systems", *NeurIPS*, 2015.

Son

Bölüm 35: Makine Öğrenmesi Sistemleri ve MLOps

← Önceki bölüm · Alanın haritası · Sonraki bölüm →