Day 16 – ML Theory

Cross-Validation & Data Leakage

Amaç

Bu çalışmanın amacı, bir makine öğrenmesi modelinin performansını **doğru ve güvenilir şekilde değerlendirmeyi** öğrenmek ve gerçek hayatta sık yapılan **data leakage (veri sızıntısı)** hatalarını fark edip önlem alabilmektir.

Bias–Variance konusunun doğal devamı olarak, bu başlık **modelin gerçekten genellenebilir olup olmadığını** anlamaya odaklanır.

---

1. Train / Test Split Neden Yeterli Değil?

Basit bir train–test split şu sorunlara yol açabilir:

- Veri şans eseri iyi veya kötü bölünmüş olabilir
- Test seti küçükse performans **yanıltıcı** olabilir
- Modelin variance'ı doğru ölçülemez

📌 Örnek:

- Train accuracy: %92
- Test accuracy: %78

Bu fark:

- Gerçekten model kötü mü?
- Yoksa test seti mi şanssız?

Bu soruya tek bir split ile net cevap verilemez.

---

2. Cross-Validation (CV) Nedir?

Cross-validation, veriyi birden fazla kez farklı train–validation bölümlerine ayırarak model performansını daha **istikrarlı** ölçmeyi sağlar.

En yaygın yöntem: **K-Fold Cross Validation**

K-Fold Mantığı

- Veri K parçaya bölünür
- Her seferinde:
  - 1 parça validation
  - K-1 parça training
- K kez eğitim yapılır
- Performans skorları ortalaması alınır

📌 Sonuç:

- Tek bir split'e bağlı kalınmaz
- Performans dalgalanmaları (variance) daha net görülür

---

3. Cross-Validation Bias–Variance ile Nasıl İlişkilidir?

- CV, **variance ölçümünü daha güvenilir hale getirir**
- Özellikle:
  - Küçük datasetlerde
  - Karmaşık modellerde

📌 CV sayesinde şunu anlayabiliriz:

- Model gerçekten iyi mi?
- Yoksa sadece belirli bir veri bölünmesinde mi iyi?

---

4. Stratified K-Fold Ne Zaman Gerekir?

Sınıflar dengesizse (imbalanced data):

- Churn (0/1)
- Fraud (çok az 1)
- Hastalık tespiti

Normal K-Fold:

- Bazı fold'larda **hiç pozitif sınıf kalmayabilir**

Stratified K-Fold:

- Her fold'da sınıf oranlarını korur
- Classification problemlerinde tercih edilmelidir

---

5. Data Leakage Nedir?

Data leakage, modelin eğitim sırasında gerçekte erişememesi gereken bilgileri dolaylı veya doğrudan görmesidir.

Bu durumda:

- Train skoru aşırı yüksek çıkar
- Validation/Test skoru gerçek hayatta **tekrar edilemez**

📌 En tehlikeli ML hatalarından biridir.

---

6. Yaygın Data Leakage Örnekleri

❌ Yanlış

- StandardScaler'ı tüm dataset'e `fit` etmek
- Feature engineering'i train + test birlikte yapmak
- Target'a çok yakın zaman bilgisini feature olarak eklemek
- Train–test ayrımından önce imputasyon yapmak

☑ Doğru

- Scaler yalnızca **train set'te fit edilir**
- Test set sadece `transform` edilir
- Tüm preprocessing adımları train içinde öğrenilir

---

7. Pipeline Kullanımı (Leakage'ı Engellemenin En Sağlıklı Yolu)

Pipeline, preprocessing + model adımlarını tek yapı altında toplar.

Avantajları:

- Data leakage riskini minimize eder
- CV ile birlikte güvenle kullanılabilir
- Production ortamına daha yakındır

📌 Pipeline olmadan yapılan CV genellikle hatalıdır.

---

8. Cross-Validation Skorları Neden Oynar?

CV sonuçlarında:

- Fold'lar arası skor farkı yüksekse → **yüksek variance**
- Skorlar tutarlı ama düşükse → **yüksek bias**

Bu bilgiler:

- Model seçimi
- Hyperparameter tuning
- Feature engineering

kararlarını doğrudan etkiler.

---

9. Gerçek Projelerde Kullanımı

Bu konu özellikle şu soruların cevabıdır:

- "Model performansını nasıl değerlendirdin?"
- "Overfitting olup olmadığını nasıl anladın?"
- "Data leakage'ı nasıl önledin?"
- "Neden cross-validation kullandın?"

📌 Remote ve mid-level mülakatlarda **çok sık sorulur**.

---

10. Mülakat İçin Kısa Özet

- Train/test split tek başına güvenilir değildir
- Cross-validation performansı daha stabil ölçer
- Stratified CV, dengesiz sınıflarda gereklidir
- Data leakage, modeli sahte şekilde başarılı gösterir
- Pipeline kullanımı leakage riskini azaltır

> **İyi bir model, yalnızca yüksek skor değil,**
> **güvenilir ve tekrar edilebilir performans sunan modeldir.**