



# AI & Data Science Master Syllabus: "From Scratch to PyTorch"

## BÖLÜM 1: Temeller ve Matematiğin Kodlanması (Pure Python & NumPy)


### MODÜL 1: Makine Öğrenmesinin Dili (Lineer Cebir)

- **Teori:** Skaler, Vektör, Matris kavramları. Dot Product (İç Çarpım) mantığı. Broadcasting ve Vektörizasyonun performans farkı.
-  **Proje (Kendi Photoshop'un):** Hazır kütüphane kullanmadan, matris işlemleriyle bir resmi siyah-beyaza çeviren veya flulaştıran bir script yazmak.


### MODÜL 2: Öğrenmenin Motoru (Kalkülüs ve Optimizasyon)

- **Teori:** Fonksiyonlar, Minima/Maxima noktaları. Türev ne demektir? Gradient Descent (Gradyan İnişi) algoritmasının mantığı ve Learning Rate.
-  **Proje (Kör Dağcı):** 3 boyutlu bir matematiksel fonksiyonun ( $z = x^2 + y^2$ ) en dip noktasını türev alarak bulan algoritmayı yazmak.


### MODÜL 3: İlk Yapay Zeka (Lineer Regresyon)

- **Teori:**  $y = wx + b$  hipotezi. Loss Function (MSE) neden kareseldir? Analitik çözüm vs İteratif çözüm.
-  **Proje (Fiyat Tahmincisi):** Ev veya Araba özelliklerine göre (metrekare, yaş vb.) fiyat tahmini yapan modeli sıfırdan kodlamak.


### MODÜL 4: Karar Verme (Lojistik Regresyon ve Sınıflandırma)

- **Teori:** Sigmoid fonksiyonu ile olasılık hesabı. Binary Cross-Entropy Loss. Sınıflandırma metrikleri (Precision, Recall).
-  **Proje (Spam Dedektörü):** Basit metin analizi ile bir mesajın Spam olup olmadığını (0 veya 1) ayıran sistem.

### MODÜL 5: Yapay Sinir Ağları (Neural Networks - Forward Pass)

- **Teori:** İnsan beyni simülasyonu. Gizli Katmanlar (Hidden Layers). Aktivasyon Fonksiyonları (ReLU, Tanh) neden gereklidir?
-  **Proje (Mantık Kapıları):** Tek katmanlı modellerin çözemediği XOR problemini çözen çok katmanlı bir ağ kurmak.

### MODÜL 6: Büyü Bozumu (Backpropagation - Backward Pass)

- **Teori:** Zincir Kuralı (Chain Rule). Hatanın ağ boyunca geriye taşınması. Ağırlık güncellemeleri.
-  **CAPSTONE PROJE (Sıfırdan MNIST):** Hiçbir deep learning kütüphanesi kullanmadan, el yazısı rakamları tanıyan tam teşekküllü bir Neural Network Class'ı yazmak.

---

## BÖLÜM 2: PyTorch Framework'ü ve Modern Derin Öğrenme

## MODÜL 7: PyTorch Devrimi

- **Geçiş:** NumPy kodlarının PyTorch Tensor'larına çevrilmesi.
- **Autograd:** Elle yaptığımız türevlerin `loss.backward()` ile otomatikleşmesi.
- **GPU:** Kodların ekran kartında çalıştırılması.

## MODÜL 8: İleri Seviye Mimariler

- **CNN:** Görüntü İşleme (Convolutional Neural Networks).
- **RNN/Transformer:** Doğal Dil İşleme (NLP) temelleri.
- **Deployment:** Modelin kaydedilmesi ve kullanıma sunulması.