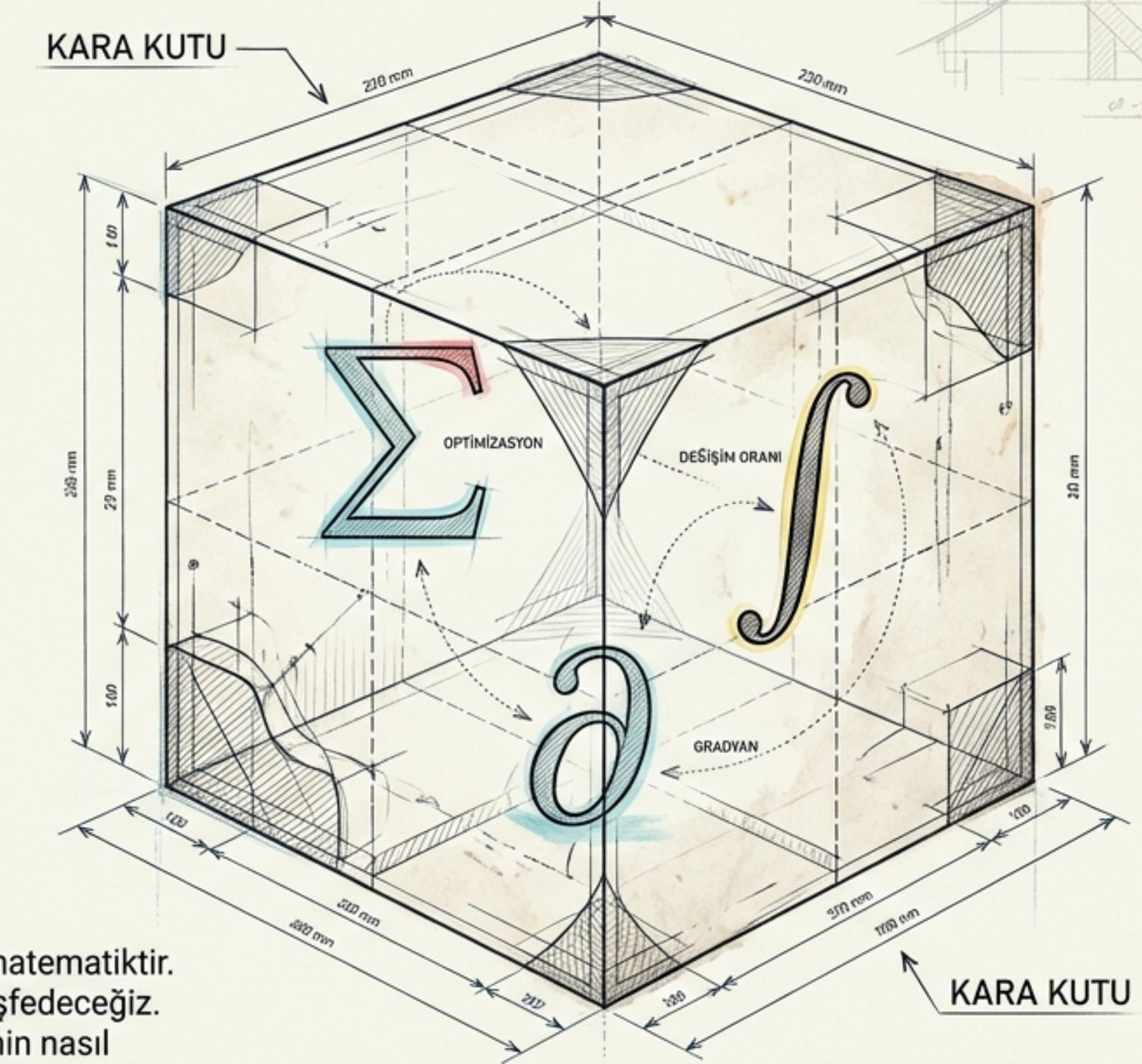


MAKİNE ÖĞRENMESİNİN MATEMATİĞİ KARA KUTUNUN İÇİNDE NE VAR?

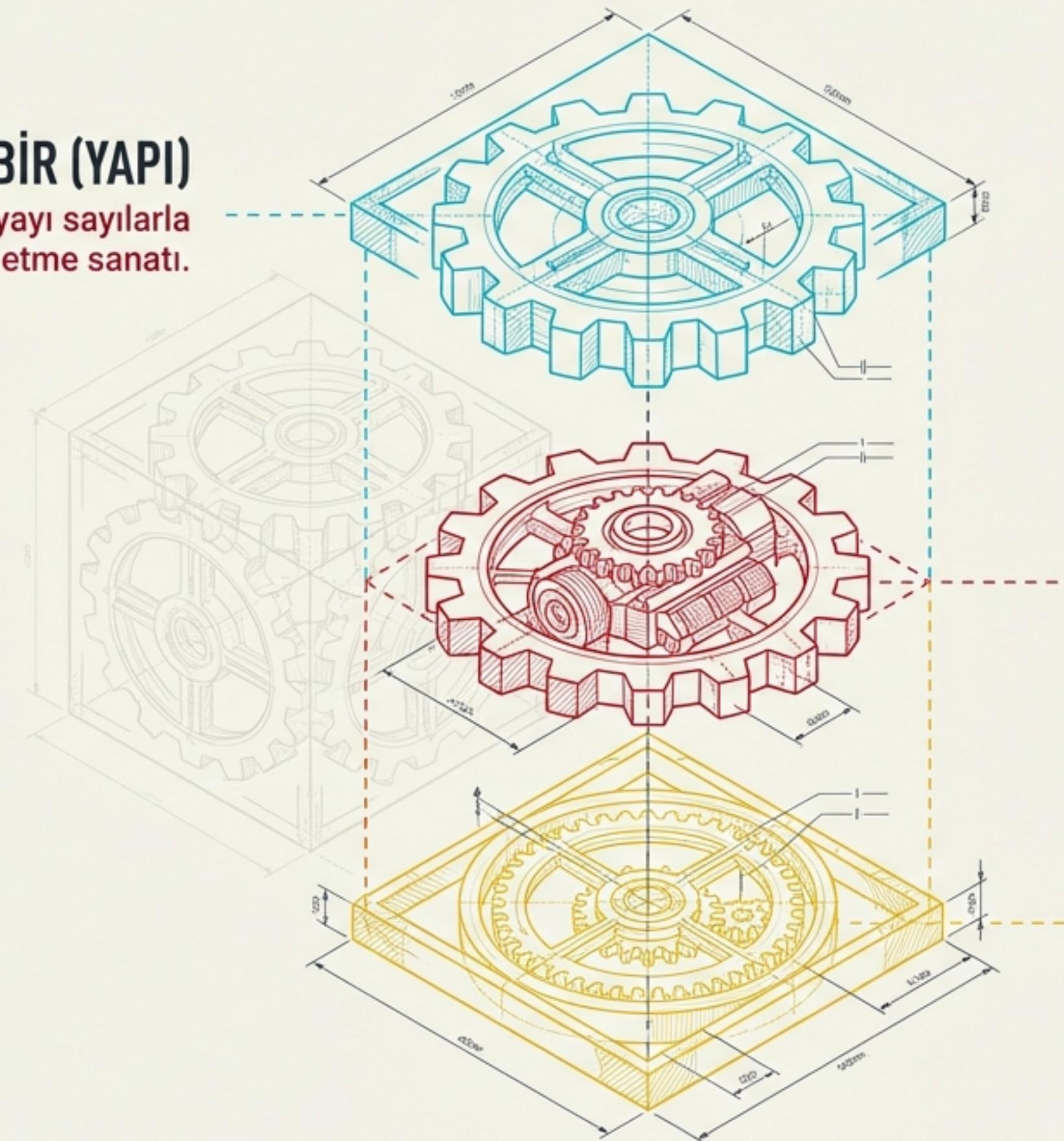
Yapay Zekayı Mümkürn Kilan 3 Temel
Sütun Üzerine Bir Yolculuk

Yapay zeka sıhir değildir; verinin üzerine inşa edilmiş zarif bir matematiktir. Bu sunumda, o matematiği yöneten dili, motoru ve pusulayı keşfedeceğiz. Bu sunum, karmaşık denklemleri ezberletmek için değil, sistemin nasıl çalıştığını 'hissetmeniz' için tasarlandı.



1. LINEER CEBİR (YAPI)

Verinin Dili. Dünyayı sayılarla temsil etme sanatı.



TEMEL MESAJ

Bir modelin öğrenmesi için önce dünyayı temsil etmesi, hatasını anlayıp düzeltmesi ve belirsizlikle başa çıkması gereklidir.

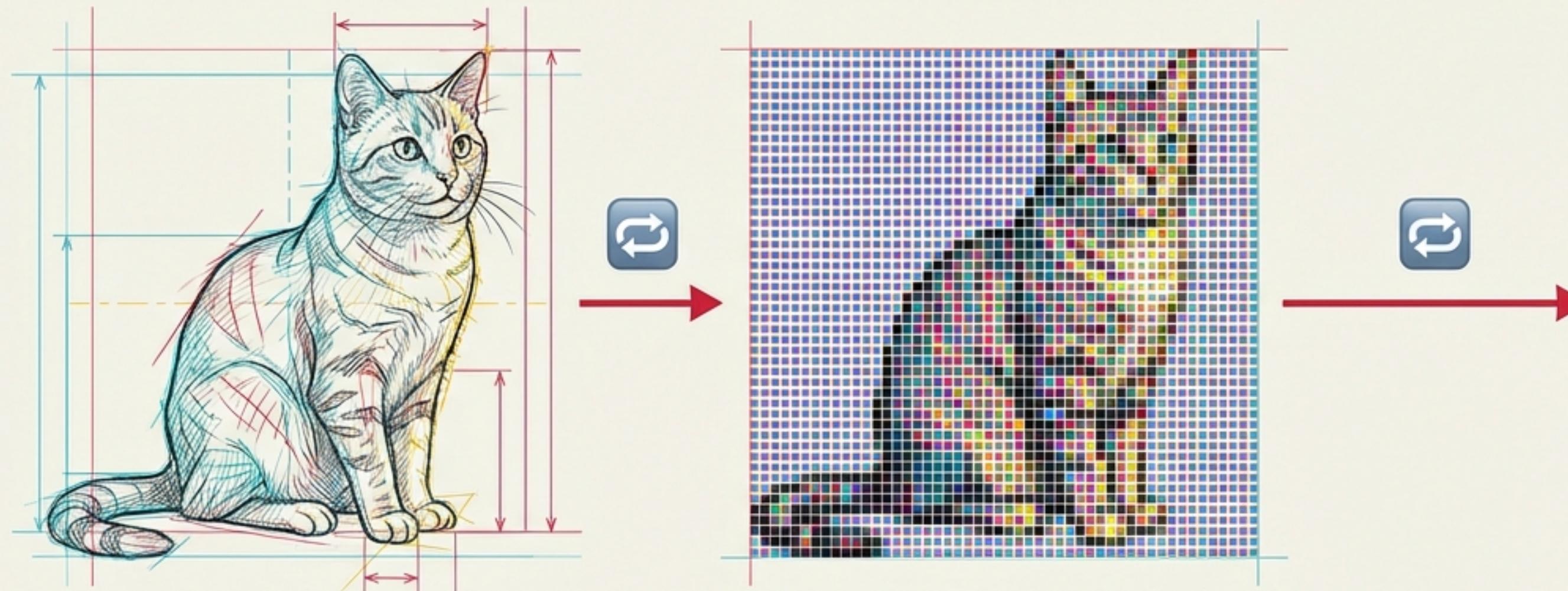
2. KALKÜLÜS (MOTOR)

Öğrenmenin Motoru. Hatalardan ders çıkarma ve optimizasyon.

3. OLASILIK (ORTAM)

Belirsizliğin Yönetimi. Kaotik ve gürültülü verilerle başa çıkma.

LINEER CEBİR: BİLGİSAYAR DÜNYAYI NASIL GÖRÜR?



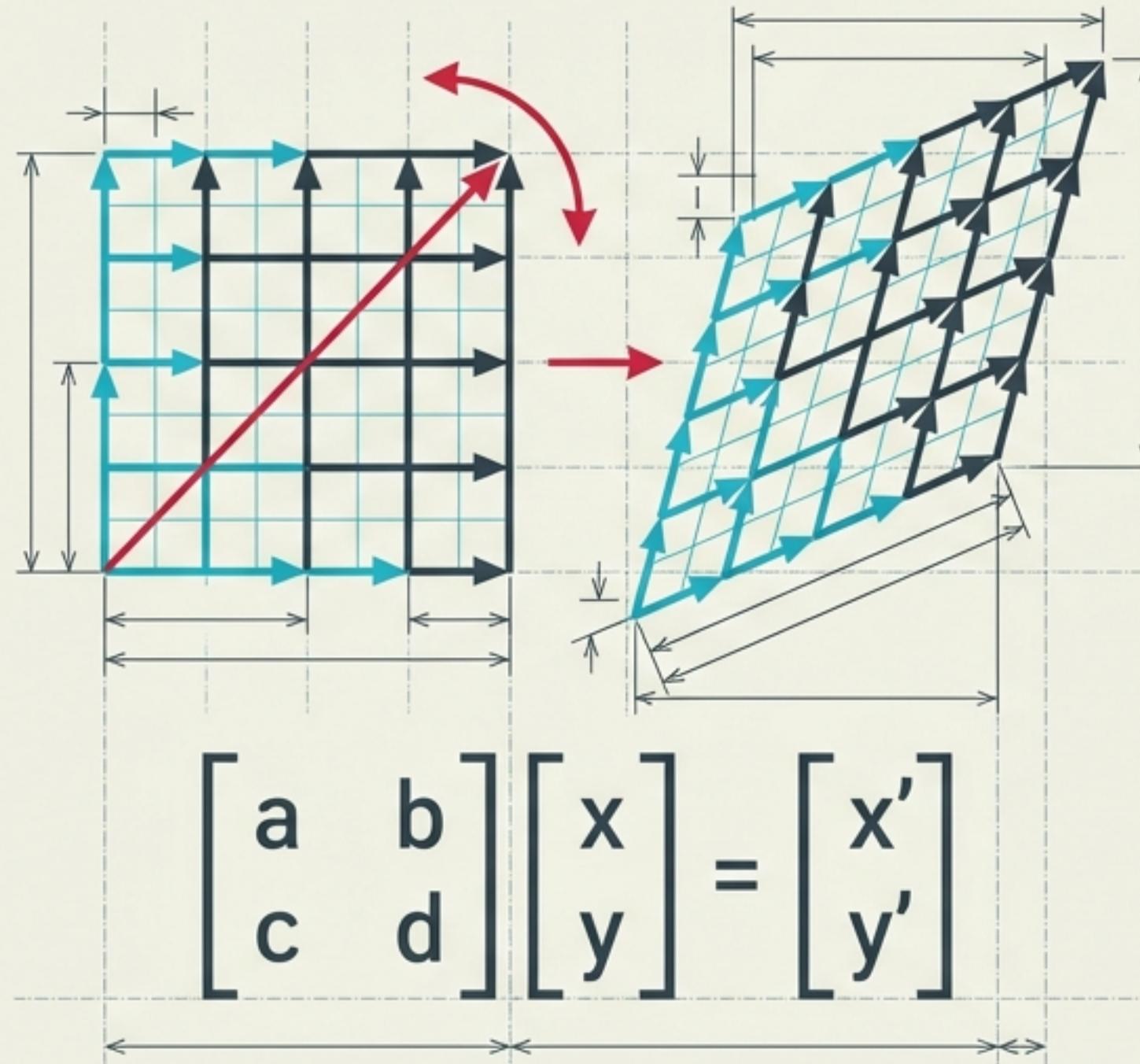
Vektörler: Verinin atomlarıdır. Bir resim, bir ses dosyası veya bir kelime; makine için sadece sıralı bir sayı listesidir.

Vektör Uzayı: Bu sayıların yaşadığı çok boyutlu evren. 'Boyut' kavramı korkutucu gelmesin; bir Excel tablosundaki her sütun yeni bir boyuttur. Modeller Modeller binlerce boyutlu uzayda desenler arar.

125
45
210
...
180
126
130
174
192
73
153
194
210
186
272
210
230
196
195
186
125
74
210
45
...
56
198
73

VEKTÖR:
Verinin atomları.

MATRİSLER VE DÖNÜŞÜMLER

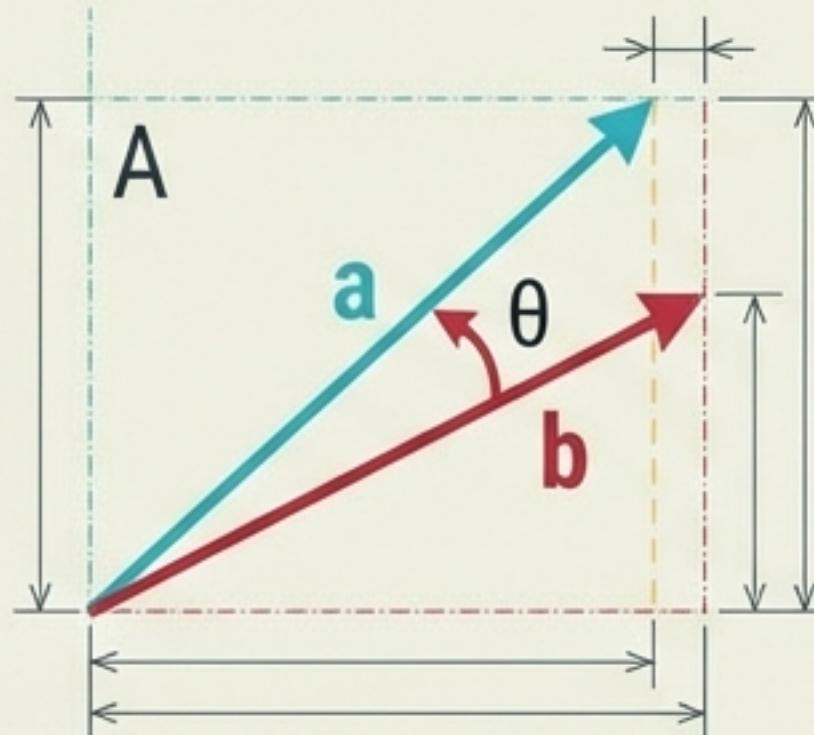


Matris: Veri tabloları veya sinir ağlarındaki “ağırlıklar” (weights) bütünüdür.

Dönüşüm: Bir vektörü bir matrisle çarpmak, o veriyi uzayda ‘bükmek’, ‘döndürmek’ veya ‘ölçeklendirmektir’. Sinir ağları, girdiyi doğru çıktıya dönüştürene kadar matris çarpımları yaparak veriyi işler.

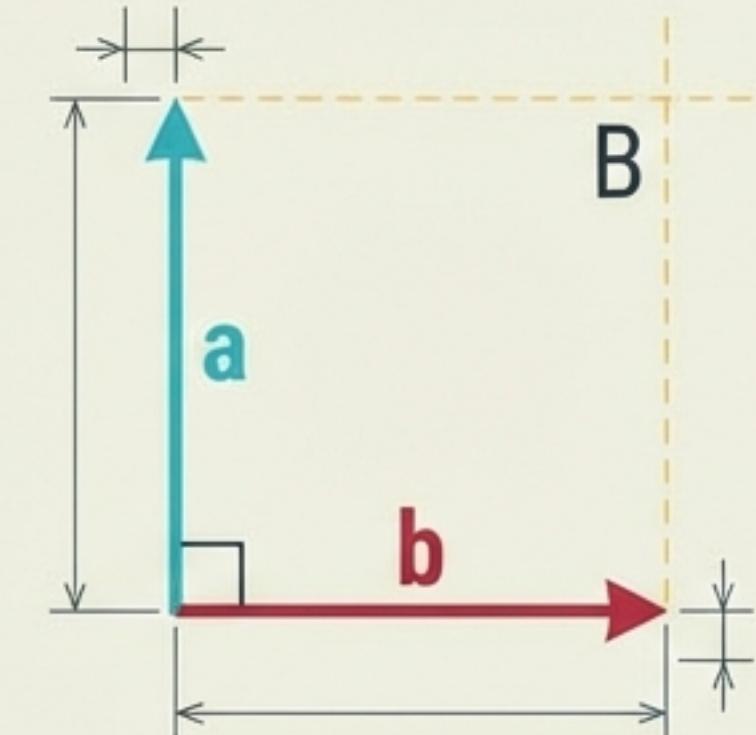
TENSÖRLER: Matrislerin daha yüksek boyutlu (3D (3D veya daha fazla) genelleştirilmiş halidir. Derin öğrenmenin (Deep Learning) temel yapı taşıdır.

DOT PRODUCT: BENZERLİK DEDEKTÖRÜ



$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos(\theta)$$

Yüksek Benzerlik (Büyük Çarpım).



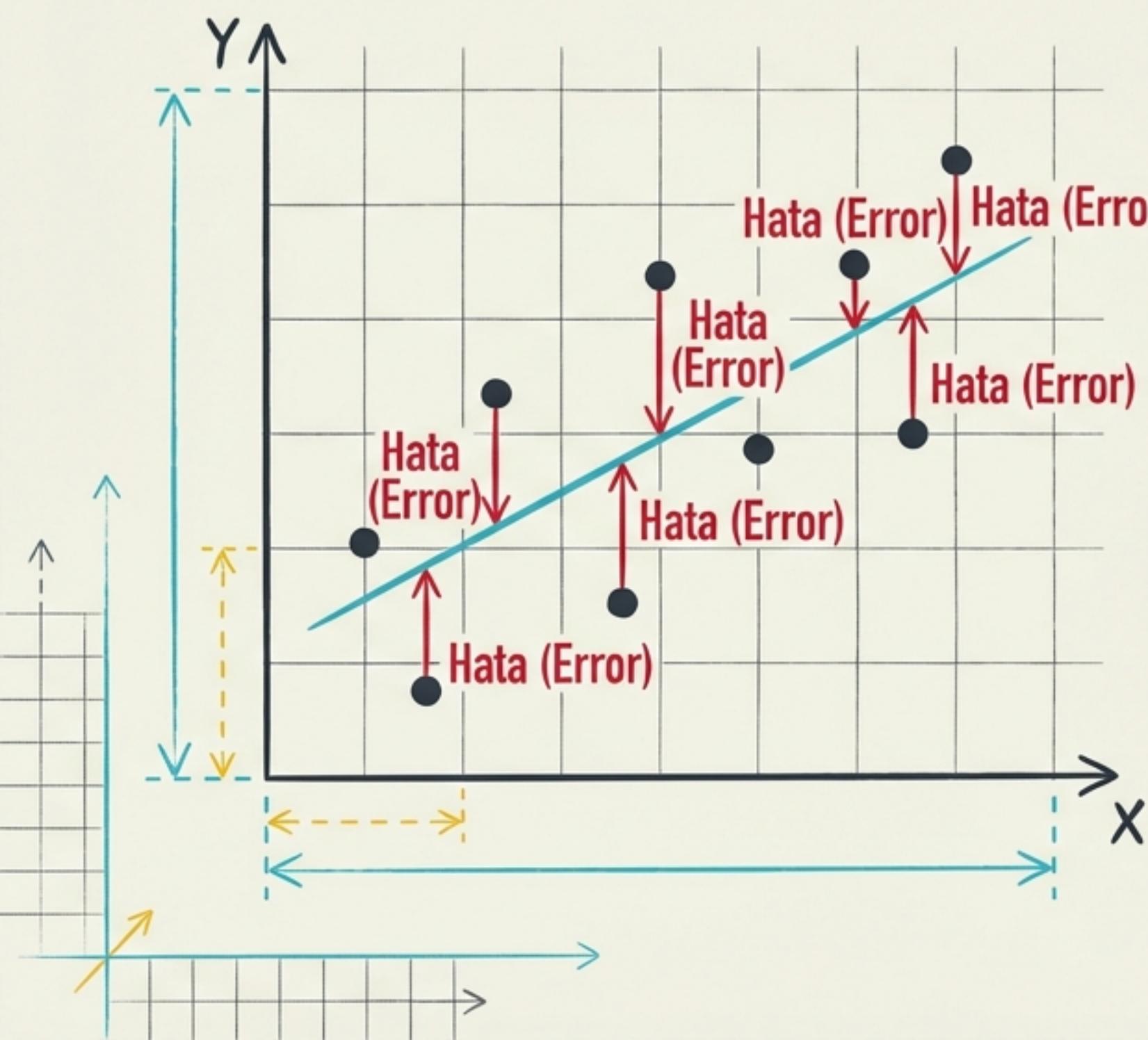
Benzerlik Yok (Sıfır).

Ne İşe Yarar?

İki vektörün ne kadar ‘uyumlu’ olduğunu ölçer. Netflix tavsiye sistemleri (sizin vektörünüz ile filmin vektörü arasındaki açı) ve NLP (kelime benzerlikleri) bu basit çarpım üzerine kuruludur.

KALKÜLÜS: HATANIN ÖLÇÜLMESİ (LOSS FUNCTION)

Hedef, 'Kayıp' Değerini Minimize Etmektir



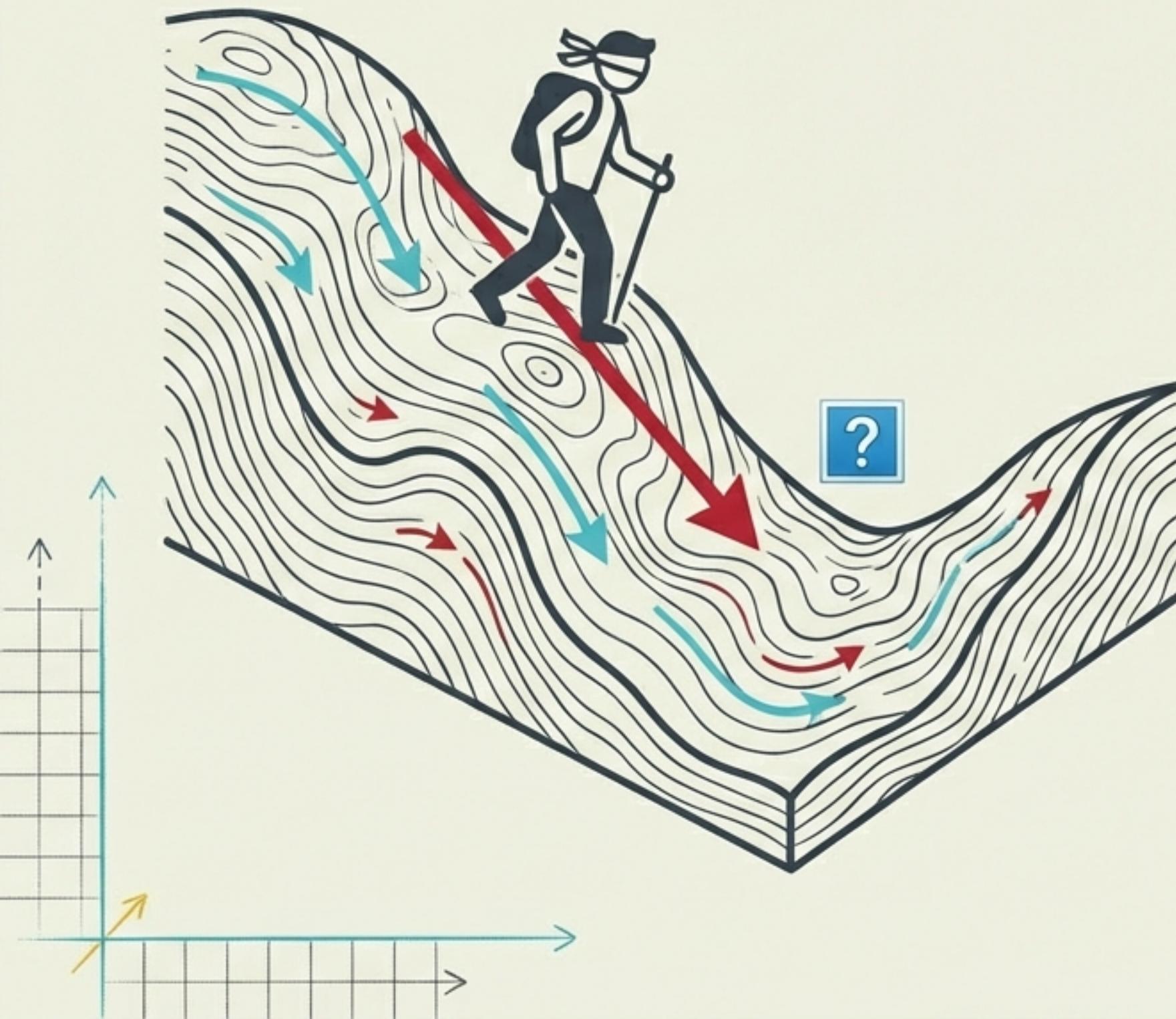
MSE (Mean Squared Error): Hataların karesini alır. Büyük hataları daha şiddetli cezalandırır. Regresyon problemlerinde yaygındır.

MAE (Mean Absolute Error): Hataların mutlak değerini alır. Aykırı verilere karşı daha dirençlidir.

Cross-Entropy: Sınıflandırma problemlerinde (Kedi mi Köpek mi?) modelin şaşkınlığını ölçer.

Öğrenmek demek, bu Loss fonksiyonunun ürettiği sayıyı 0'a yaklaşımak demektir.

TÜREV VE GRADYAN: YÖN BULMA ARACI



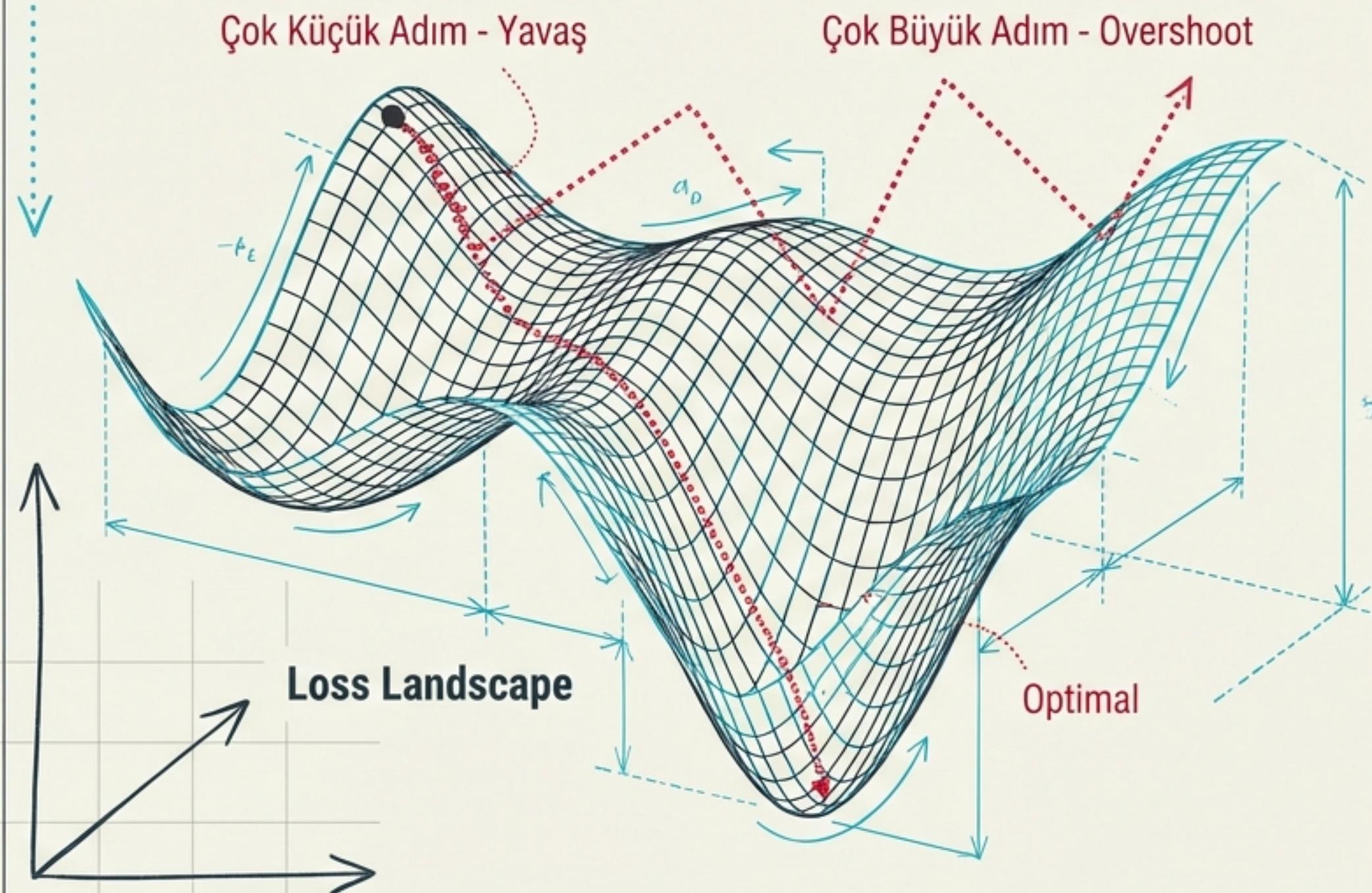
Türev: Bir fonksiyonun belirli bir noktadaki anlık değişim hızıdır (Eğim).

Gradyan: Çok boyutlu uzayda, hatanın en dik *arttığı* yönü gösteren vektördür.

Strateji: Gradyanın **tersini** alırsak, bizi hatanın en düşük olduğu yere (vadinin dibine) götürürecek yönü bulmuş oluruz.

"Gözleri bağlı bir dağcının, ayaklarıyla eğimi hissederek aşağı inmeye çalışması gibidir."

GRADIENT DESCENT VE LEARNING RATE



Algoritma:

1. Gradyanı hesapla.
2. O yönde bir adım at.
3. Tekrarla.

Learning Rate (Öğrenme Hızı - α):

Adımların büyüklüğünü belirler.

Çok Küçük: Öğrenme sonsuza kadar sürer.

Çok Büyük: Hedef ıskalanır, model ıraksar.

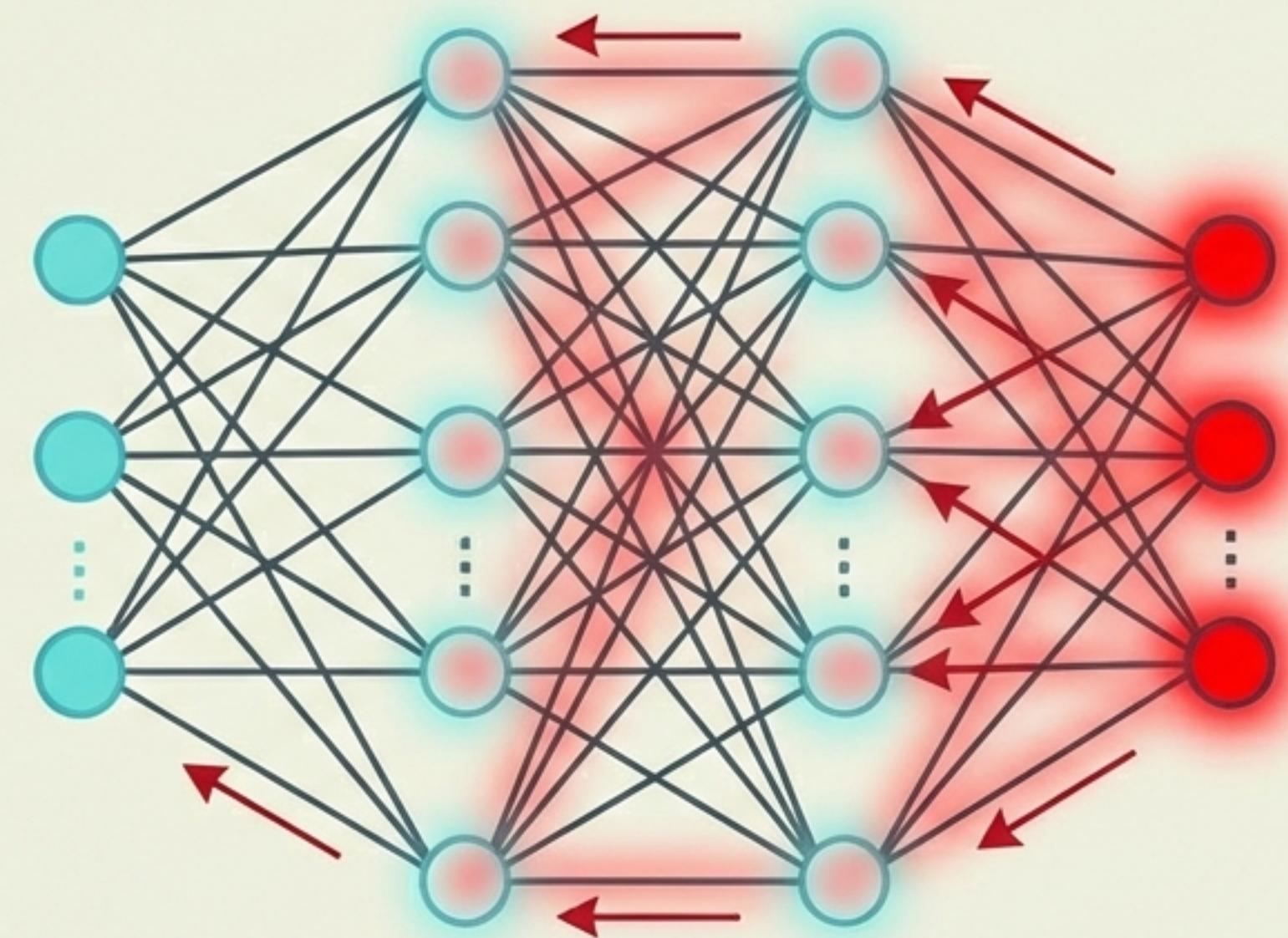
Adaptive (Örn: Adam): Adım büyüklüğünü duruma göre otomatik ayarlar.

BACKPROPAGATION: HATADAN KİM SORUMLU?

Sorun: Çıktıda bir hata var, ama buna binlerce parametreden hangisi sebep oldu?

Zincir Kuralı (Chain Rule): Kalkülüsün kalbi. Türevleri sondan başa doğru çarparak taşıriz.

Mekanizma: Hatayı ağını gerisine doğru yayarız ve her parametrenin hatadaki ‘suç payını’ hesaplarız. Ağırlıklar, hataya katkıları oranında güncellenir.



OLASILIK: BELİRSİZLİKLE DANS

2+2=4 Deterministik

Deterministik vs. Stokastik: Lineer Cebir ve Kalkülüs kesin sonuçlar verir. Ancak gerçek hayat kaotiktir. Veriler gürültülüdür ve eksiktir.

Güven Aralığı: Modellerimiz ‘Bu bir kedi’dir’ demez; ‘Bunun kedi olma ihtimali %98’dir’ der.

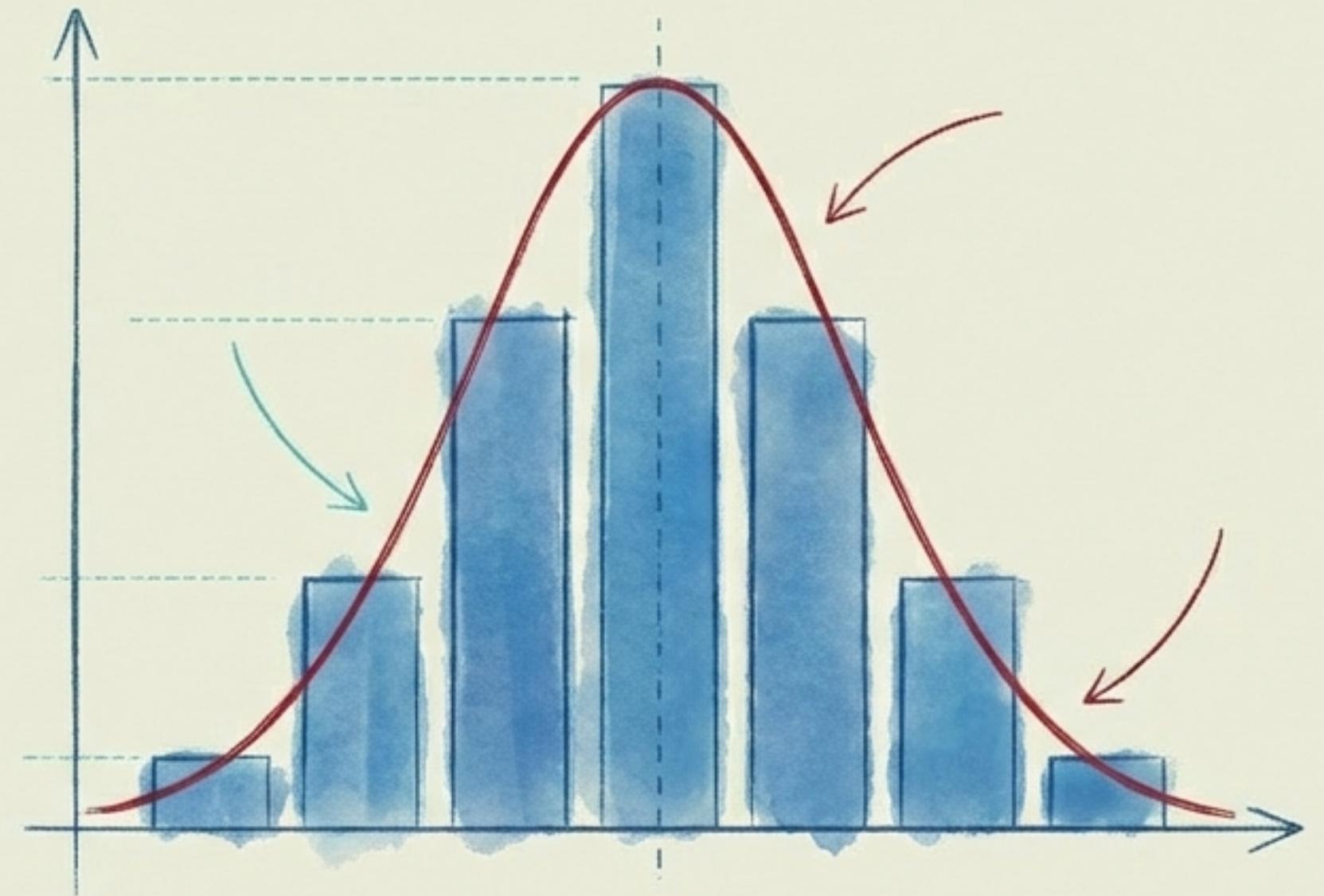
Olasılık, modelin **kendine olan güvenini** ölçmemizi sağlar.



VERİNİN ŞEKLİ VE DAĞILIMLAR

Normal Dağılım (Gaussian): Doğadaki pek çok olay (boy, hata oranları) çan eğrisi şeklindedir. Modelleri eğitirken parametreleri genellikle bu dağılımdan rastgele seçerek başlatırız.

Maximum Likelihood Estimation (MLE): Elimizdeki veriyi “en muhtemel” kıyan parametreleri bulma yöntemidir. ‘Bu veriyi hangi ayarlar üretmiş olabilir?’ sorusunun cevabıdır.



BAYES TEOREMİ: İNANCI GÜNCELLEMEK

Felsefe: Bir şeye dair mevcut inancımız (Prior), yeni kanıtlar (Evidence) geldikçe güncellenir (Posterior).

Uygulama: Spam filtreleri (Bu kelimeyi gördüm, spam olma ihtimali arttı) ve tıbbi teşhisler.

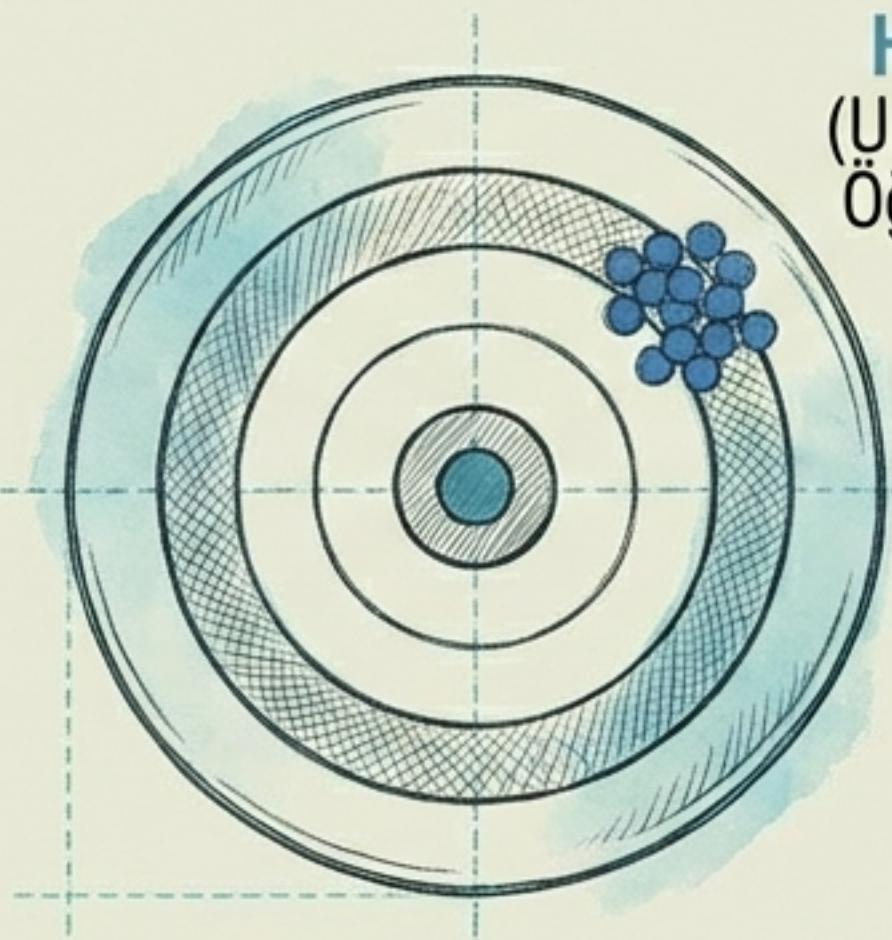
Prior
(Önsel Bilgi) Likelihood
(Yeni Kanıt)



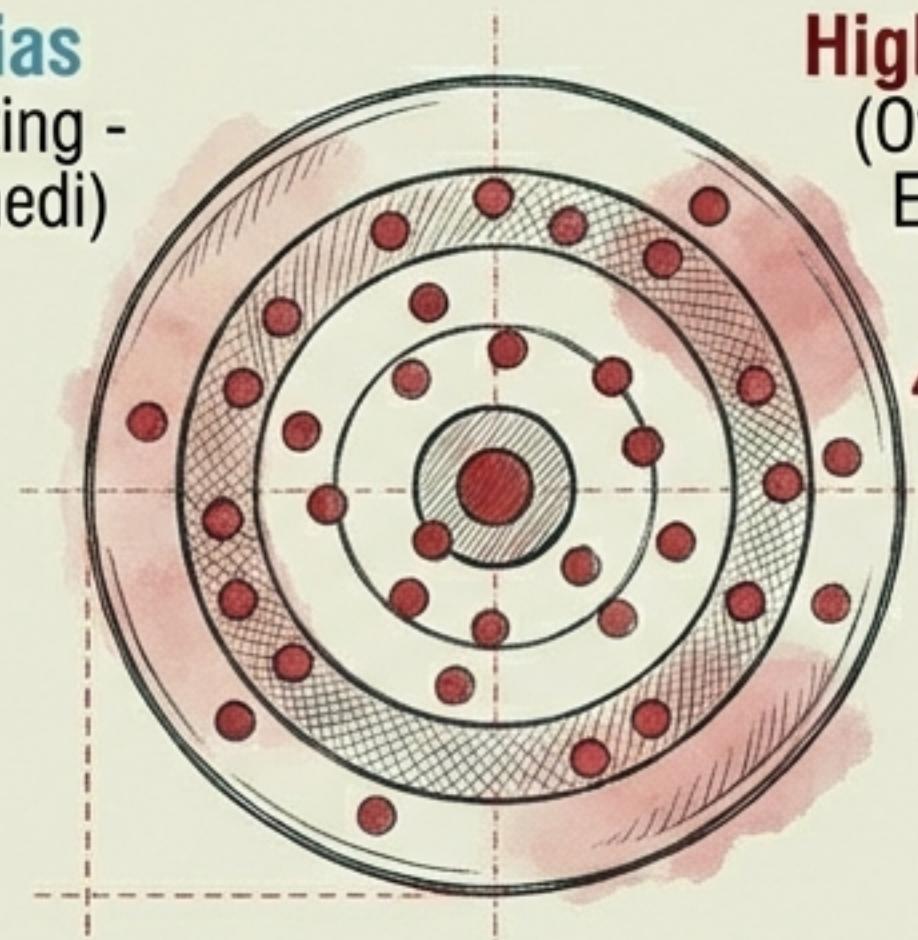
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

BIAS - VARIANCE DENGESİ

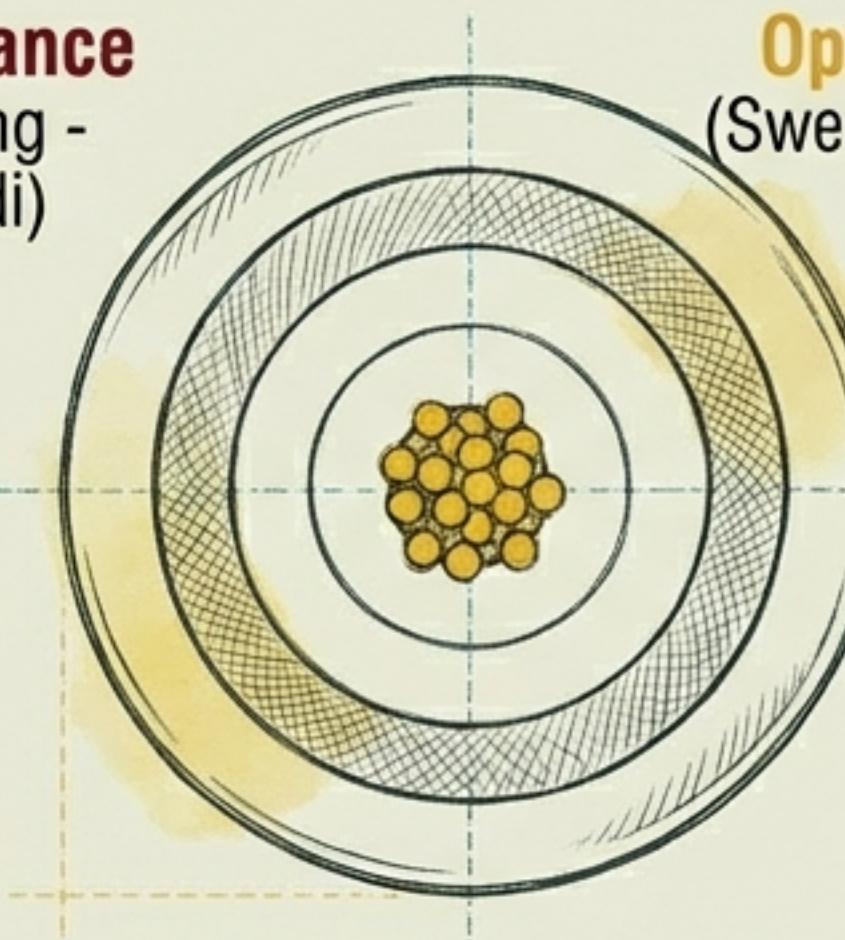
Ezberlemek mi, Öğrenmek mi?



High Bias
(Underfitting -
Öğrenemedi)



High Variance
(Overfitting -
Ezberledi)



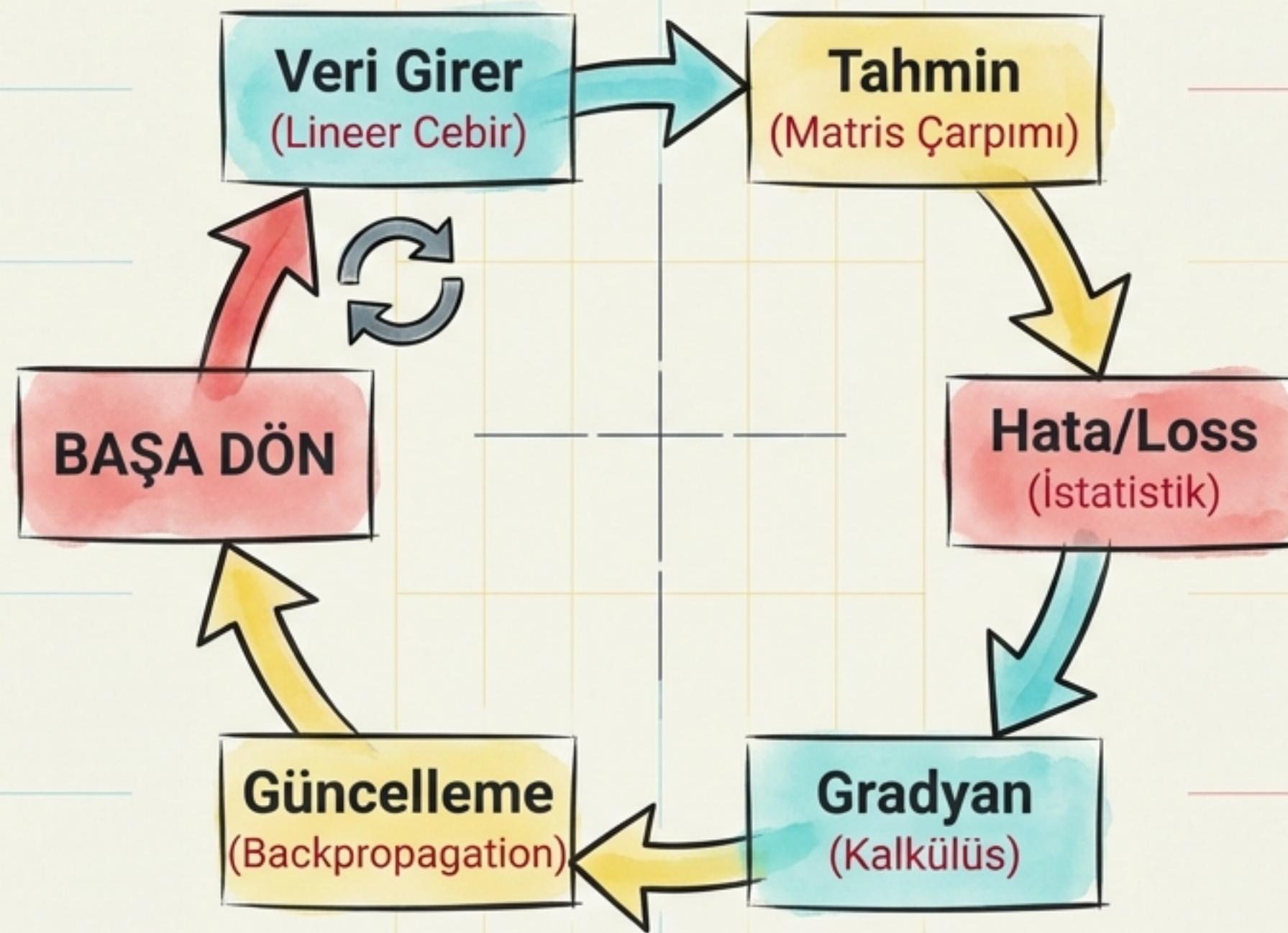
Optimal
(Sweet Spot)

Underfitting: Model çok basittir, verideki deseni yakalayamaz.

Overfitting: Model çok karmaşıktır, veriyi ezberler, gürültüyü bile öğrenir ama yeni veride başarısız olur.

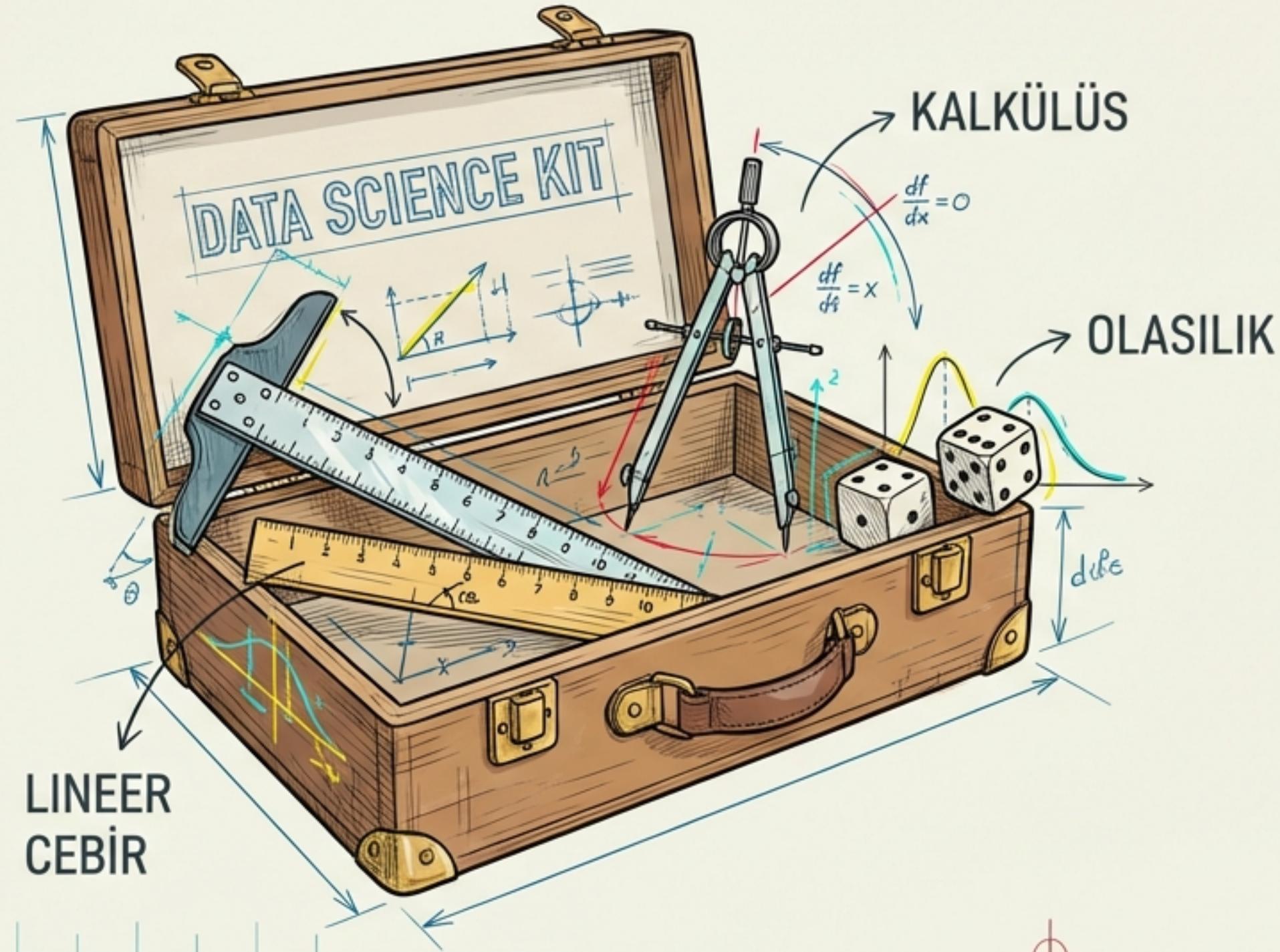
Amaç: Bu ikisi arasındaki dengeyi bulmaktır.

ÖĞRENMENİN ANATOMİSİ: BÜYÜK DÖNGÜ



Milyarlarca dolarlık yapay zeka ekonomisi, bu basit matematiksel döngünün saniyede trilyonlarca kez tekrarlanmasından ibarettir.

SIRADA NE VAR?



Matematik bir engel değil, bir araçtır. En karmaşık Derin Öğrenme modelleri bile **Vektörler, Türevler ve Olasılık** dağılımlarından oluşur.

Öneri: Bu kavramları kodlayarak (Python, NumPy, PyTorch) pekiştirebilirsiniz.

Kaynaklar:

- StatQuest with Josh Starmer
- 3Blue1Brown
- GeeksforGeeks & DataCamp