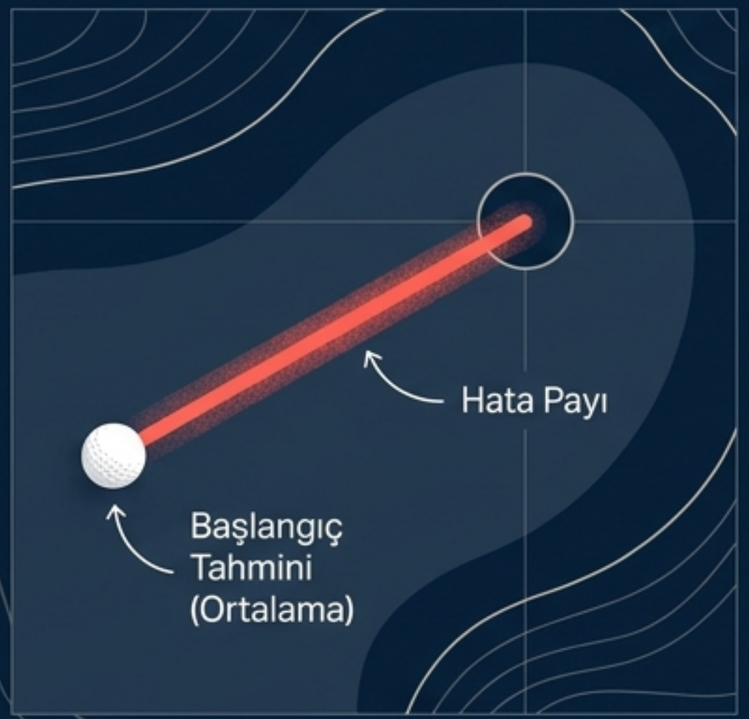
Gradient Boosting: Hatalardan Öğrenme Sanatı

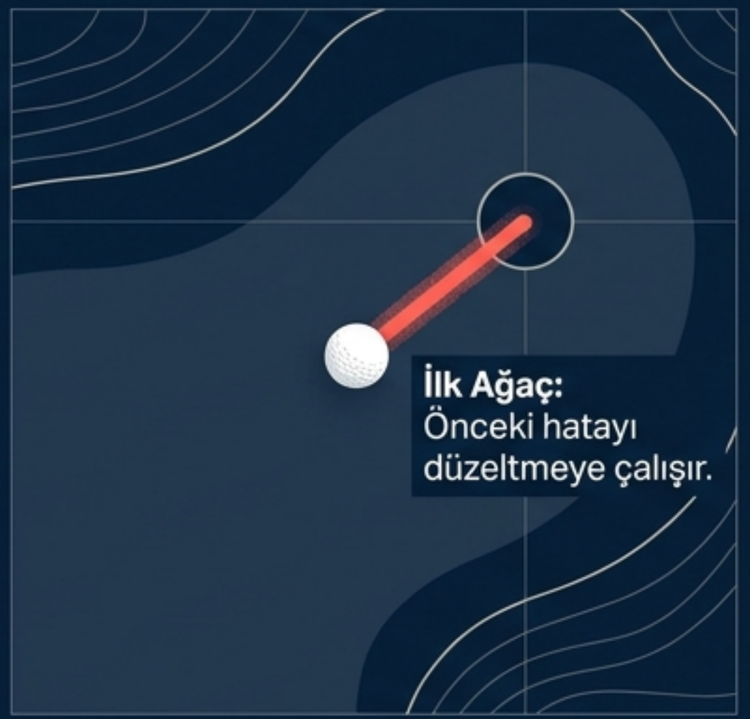
Makine öğrenmesinde kusuru mükemmelliğe dönüştüren algoritmanın anatomisi.

NotebookLM

Kusuru Düzeltme Felsefesi

Gradient Boosting, hedefe tek bir mükemmel hamlede ulaşmaya çalışmaz. Bunun yerine, her adımda sadece '**kalan hataya**' odaklanır.







NotebookLM

Ansambl Savaşları: Ağaçlar Nasıl Birlikte Çalışır?

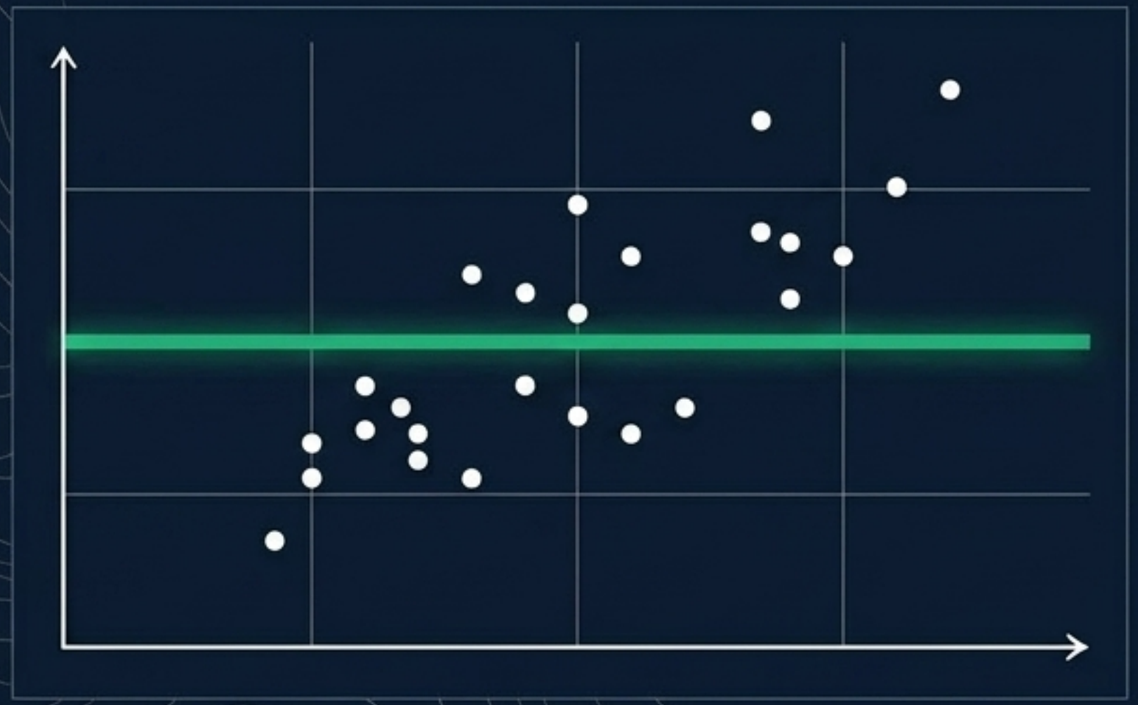
Zayıf öğrenicileri (weak learners) bir araya getirerek güçlü bir model oluşturmanın üç farklı yapısal yaklaşımı.

Random Forest	AdaBoost	Gradient Boosting (GBM)
Mimarisi: Bağımsız ve paralel ağaçlar.	**Mimarisi:** Sıralı. Her ağaç, önceki ağacın yanıldığı örneklerin ağırlığını artırır.	**Mimarisi:** Sıralı. Her ağaç, önceki ağacın hatasını (residual) tahmin eder.
Ağaç Boyutu: Tam derinlikli ağaçlar (Yüksek varyanslı).	**Ağaç Boyutu:** Sadece 'Stump' (Tek düğüm, iki yaprak).	**Ağaç Boyutu:** Kısıtlı derinlik (Genellikle 8-32 yaprak).
Birleştirme: Sonda, ortalama veya oylama (Majority rules) ile.	**Birleştirme:** Süreç boyunca, ağaçların başarılarına göre ölçeklendirilerek.	**Birleştirme:** Süreç boyunca, sabit bir 'Öğrenme Oranı' (Learning Rate) ile.

NotebookLM

Regresyon Mimarisi: İlk ve En Basit Tahmin

Gradient Boosting süreci, boş bir tuval ile değil, olası en güvenli varsayım ile başlar. Tüm örnekler için tek bir değer öngören bir **'Başlangıç Yaprağı'** (Initial Leaf). Sürekli bir değer tahmin ediliyorsa, bu değer tüm veri setinin ortalamasıdır.



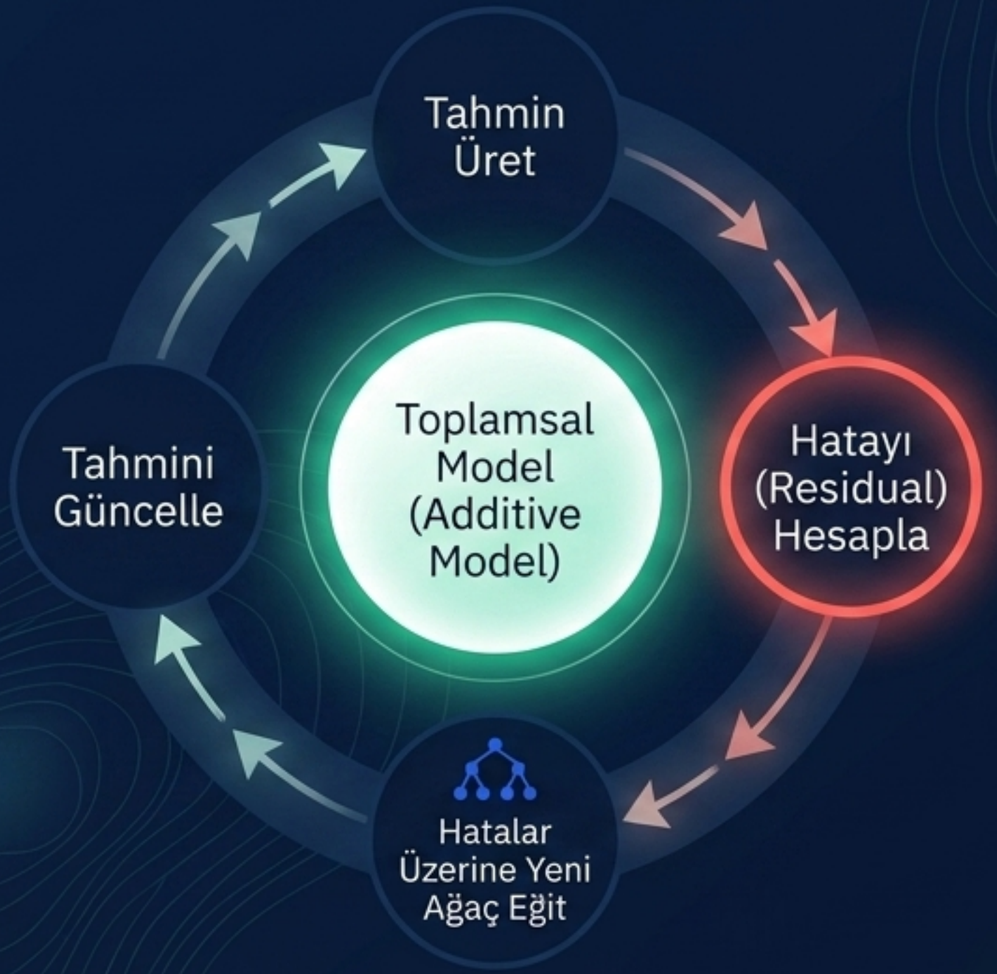
Hedefe yönelik ilk ve en basit varsayım.

$$F_0(x) = \bar{y}$$

NotebookLM

Sonsuz Geri Bildirim Döngüsü (Feedback Loop)

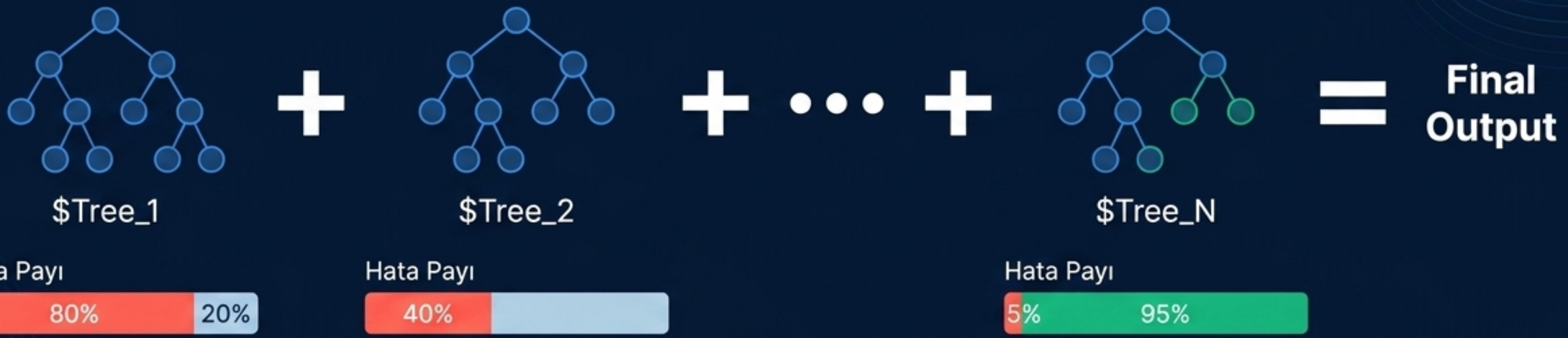
Algoritma, hedefe (Target) değil, hedeften ne kadar sapıldığına odaklanan kapalı bir sistemdir.



NotebookLM

Katmanlı Öğrenme: Toplamsal Model (Additive Assembly)

Her bir ağaç modeli sıfırdan kurmaz; önceki ağaçların bıraktığı eksiklikleri tamamlayan şeffaf bir katman olarak eklenir.



Statik bir yapı değil, ilerledikçe kendi hatalarından öğrenen dinamik bir sentez.

NotebookLM

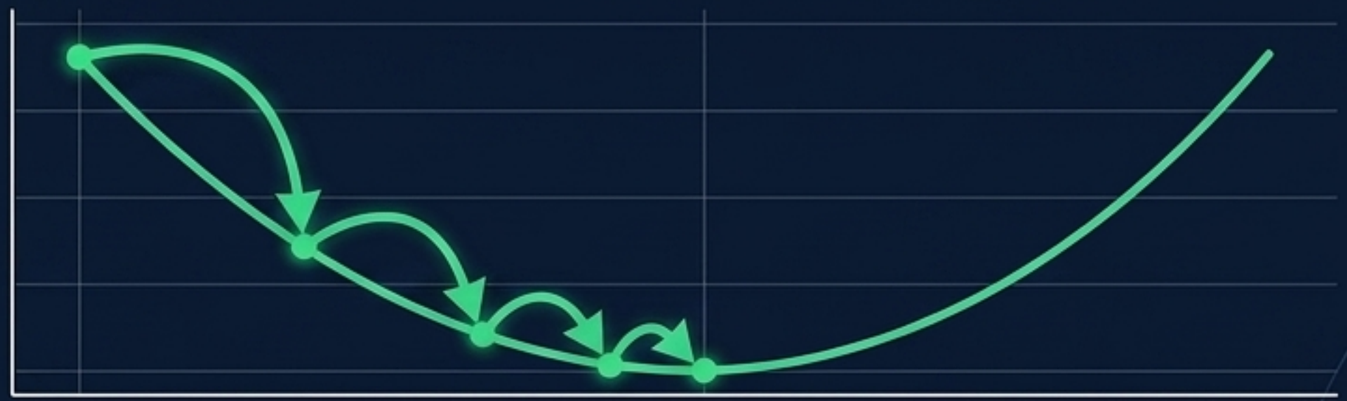
Modeli Evcilleştirmek: Öğrenme Oranı (α)

Yeni bir ağacın ürettiği düzeltmeyi modele doğrudan eklemek, hedefi aşarak (overfitting) yüksek varyansa neden olur. **Çözüm:** Her ağacın katkısını bir **Öğrenme Oranı (Learning Rate)** ile ölçeklendirmektir.

Hızlı ama Kontrolsüz:
Düşük Bias,
Yüksek Varyans



Küçük Ama
Emin Adımlar:
Güçlü Genelleme



NotebookLM

İkili Dünya: Regresyon vs. Sınıflandırma

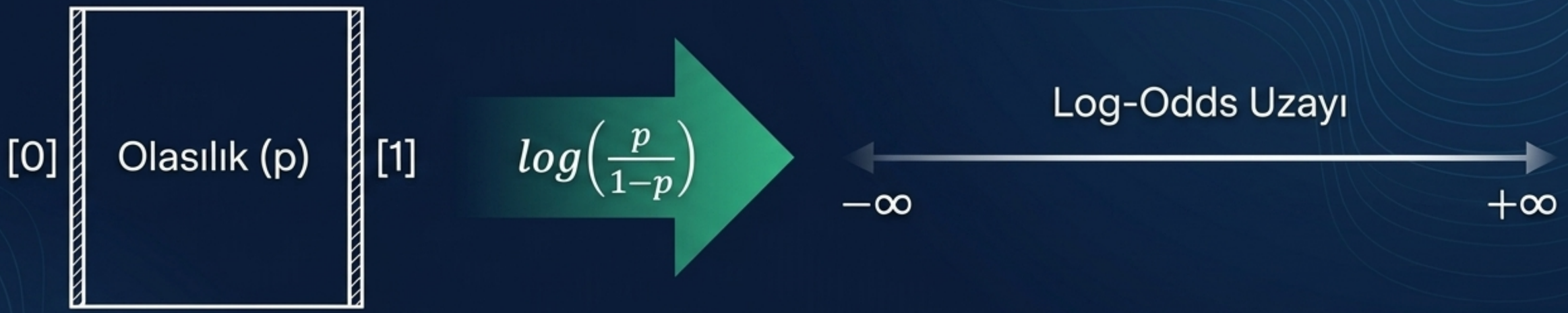
Gradient Boosting, sürekli bir değer tahmininden, olasılık tabanlı bir sınıflandırmaya geçtiğinde temel mimarisini korur, ancak metrikleri şekil değiştirir.

	Regresyon	Sınıflandırma
Problem Tipi	Sürekli Değerler	Ayrık Sınıflar
Başlangıç Tahmini (F_0)	Hedefin Ortalaması ($\bar{y}$)	Olasılıkların Log-Odds Değeri $\left(\log\left(\frac{p}{1-p}\right)\right)$
Kayıp Fonksiyonu (Loss Function)	Hata Kareler Ortalaması (MSE)	Log-Loss (Cross-Entropy)
Yaprak Çıktısı	Hataların Ortalaması	γ (Gelişmiş Formül)

NotebookLM

Sınıflandırma Kayması: Olasılıktan Log-Odds'a

Doğrudan 0 ve 1 arasına sıkışmış olasılık değerleri toplamsal (**additive**) bir model için uygun değildir. Ağaçları sonsuza kadar üst üste ekleyebilmek için, tahmini **serbestçe hareket** edebilen **Log-Odds uzayına** taşırız.



Tıpkı regresyonda olduğu gibi, ilk yaprağımız (Başlangıç Tahmini) tüm veri seti için bu Log-Odds değerini hesaplar.

NotebookLM

Hatanın Anatomisi: Log-Loss Eğrisi

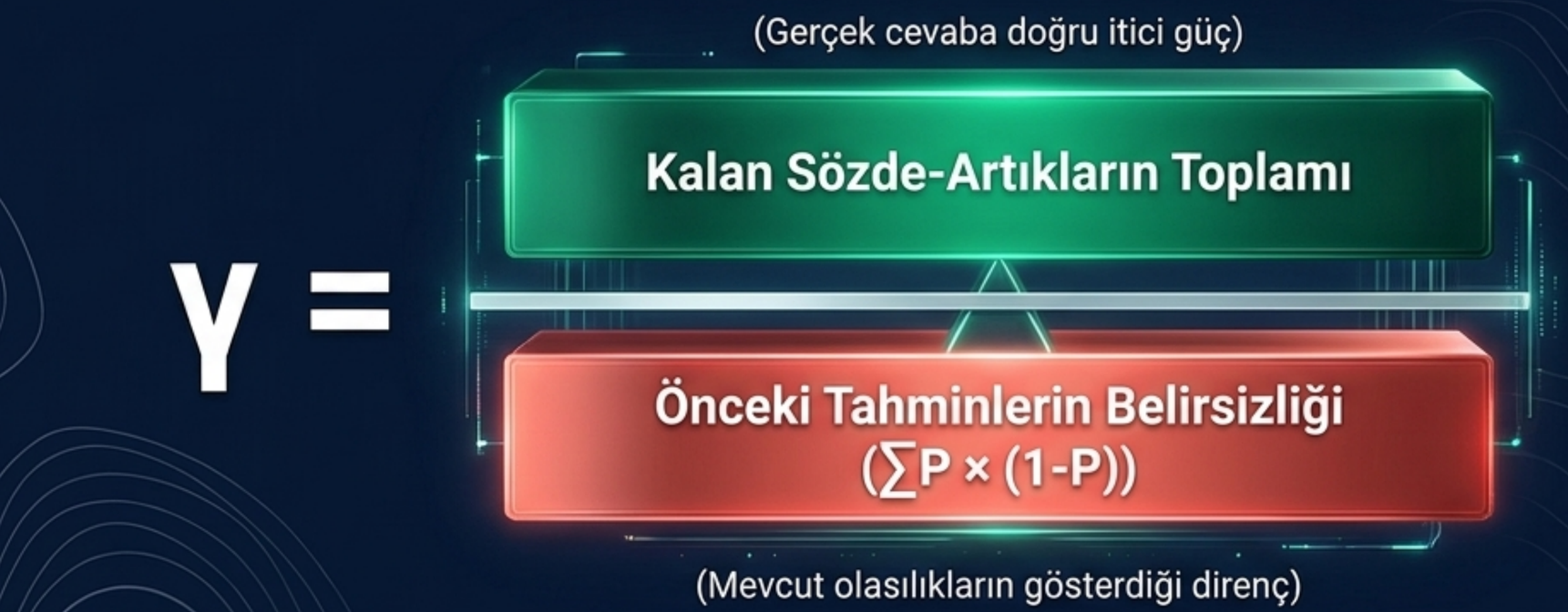
Sınıflandırmada hataları basitçe çıkaramayız. Neden? Çünkü modelin sadece yanlış olmasını değil, yanlış olup bundan **çok emin olmasını** şiddetle cezalandırmak isteriz.



NotebookLM

Matematiği Görselleştirmek: Ağaç Yapraklarının Çıktısı (γ)

Sınıflandırma ağaçlarında, her bir yaprağa düşen örnekler, karmaşık bir türev denklemi yerine basit bir denge mekanizması (tahterevalli) ile yeni çıktıyı belirler.



Payda, modelin bir örnek üzerinde ne kadar **kararsız** olduğunu ölçer. Eğer model çok eminse ($P \approx 1$ veya $P \approx 0$), **payda sıfıra yaklaşır** ve ağacın yapacağı **düzeltme katsayısı (γ) büyür**.

NotebookLM

Modeli Kontrol Paneli (Hyperparameters)

Gradient Boosting güçlü bir canavardır. Onu evcilleştirmek ve veri setine aşırı uyum sağlamasını (**overfitting**) engellemek için üç ana kontrol kadranı kullanılır.



n_estimators (Ağaç Sayısı)
"Kaç katman eklenecek?"

Not: Öğrenme oranı ile ters orantılı çalışır.



learning_rate (Öğrenme Oranı / α)
"Adımlar ne kadar büyük olacak?"

Not: Düşük oranlar (0.05 - 0.3) modeli genelleştirir ancak daha fazla ağaç gerektirir.



max_depth (Ağaç Derinliği)
"Her ağaç ne kadar karmaşık olacak?"

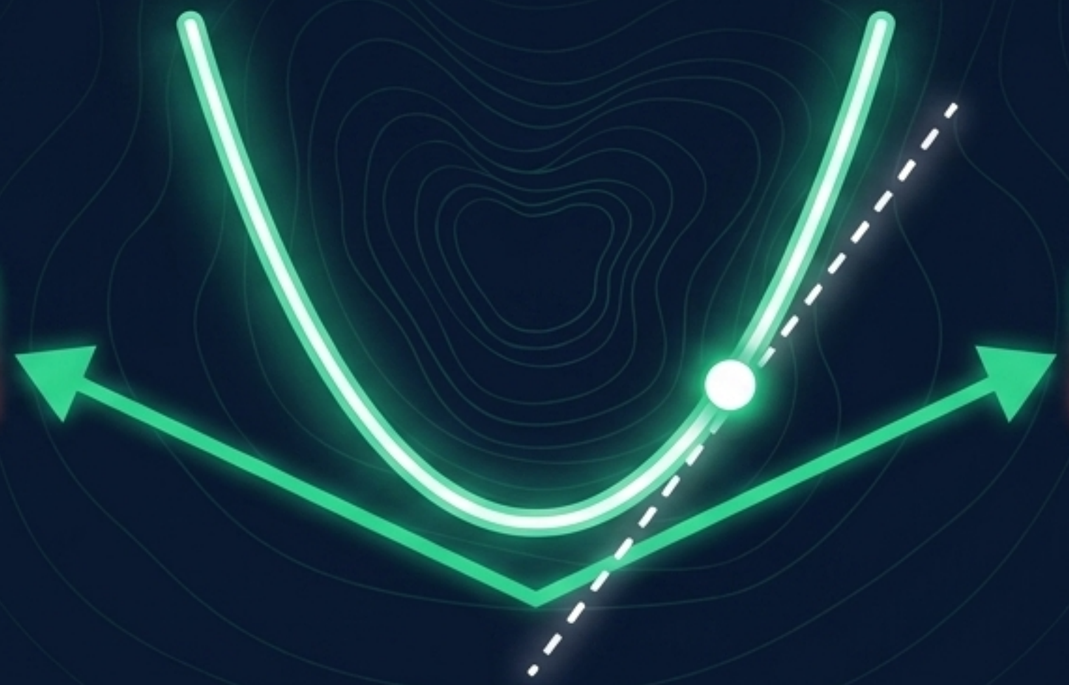
Not: Zayıf öğrenici mantığını korumak için genellikle düşük tutulur (3-8 arası).

NotebookLM

Büyük Sentez: Her Şey Türevden İbarettir

İster Regresyonun **basit hatası** olsun, ister Sınıflandırmanın **karmaşık Log-Odds denklemi** olsun...
Çizilen her ağaç, aslında tek bir **evrensel gerçekliği** hedefler: **Kayıp Fonksiyonunun Türevi**.

MSE -> Y - Ŷ
(Regresyon)



Log-Loss -> Karmaşık Artıklar
(Sınıflandırma)

Gradient Boosting, parametre uzayında (ağırlıklar) değil, fonksiyon uzayında (ağaçlar) gerçekleştirilen bir Gradient Descent işlemidir.

NotebookLM