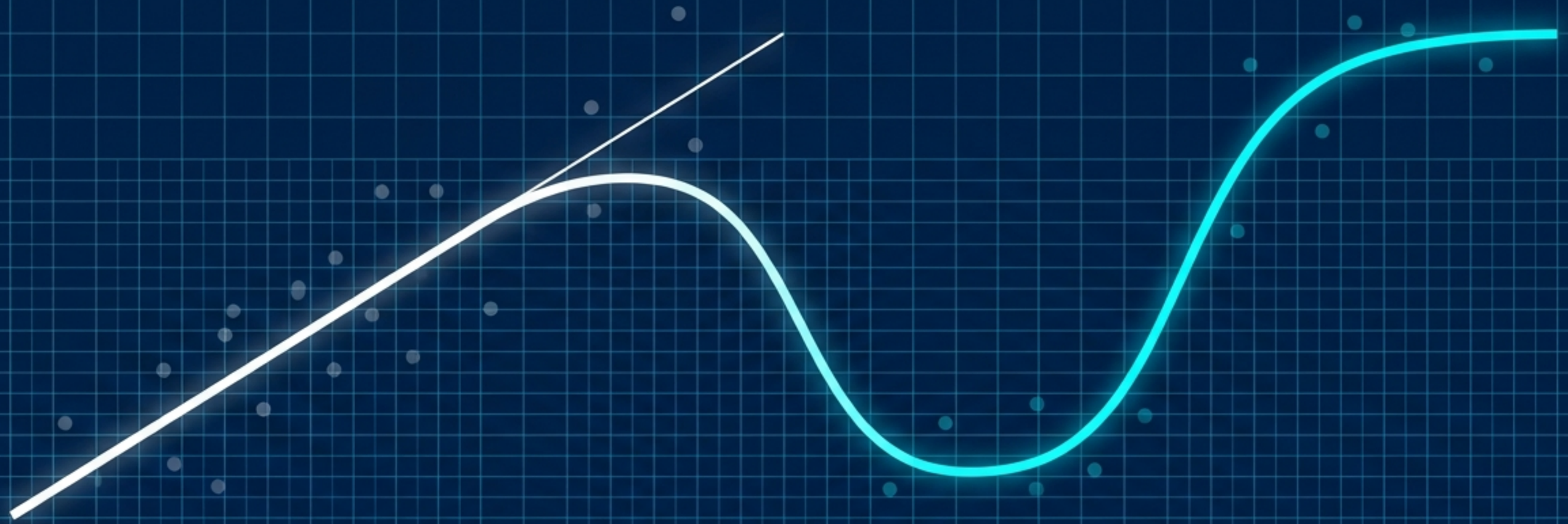


Supervised Learning & Regression Analysis

A Comprehensive Architectural Blueprint for Predictive Modeling

Covering Linear & Logistic Frameworks | Optimization Algorithms | Diagnostic Rigor | Model Regularization

The Foundation: Overview of Supervised Learning

Supervised learning is defined by the use of labeled data. The system learns a function ‘h’ (hypothesis) that maps Input (X) to Output (Y).

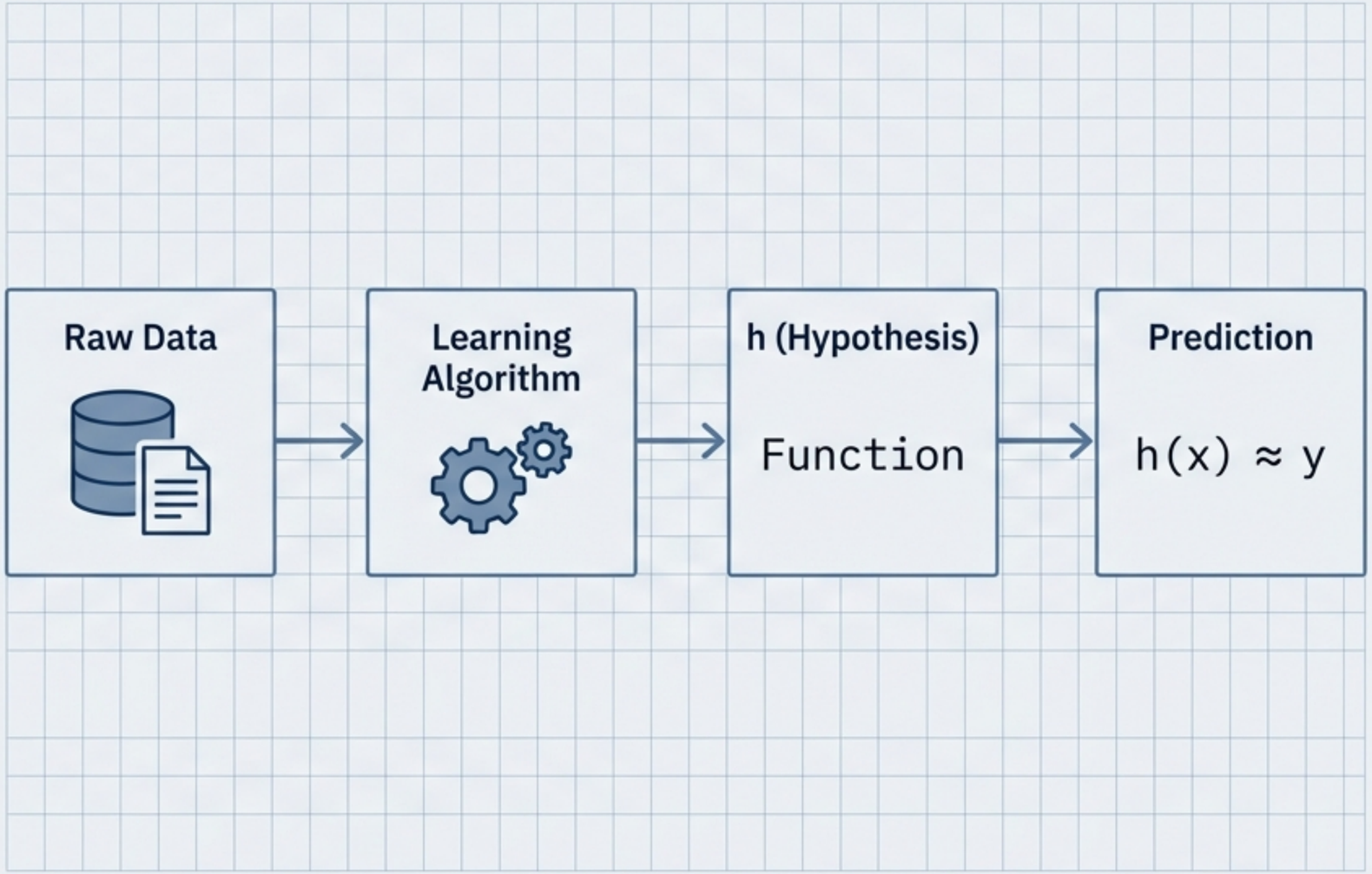
Visual Glossary

Input Variables (x):
Features or attributes.
Example: Living area (feet^2).

Target Variable (y):
The output to predict.
Example: Price ($1000s).

Training Example:
A specific pair ($x^{(i)}$, $y^{(i)}$).

Training Set:
The list of "m" training examples used to learn.

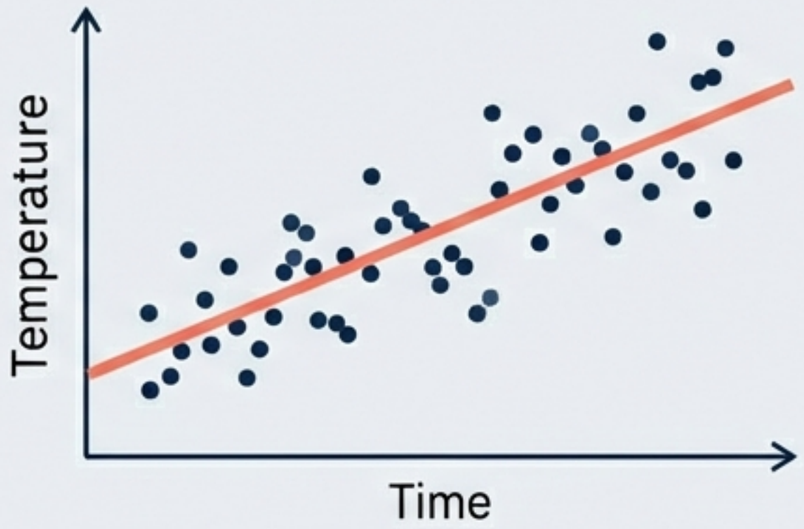


Context: Andrew Ng / CS229 Housing Price Example

The Fork in the Road: Regression vs. Classification

Predicting a Quantity

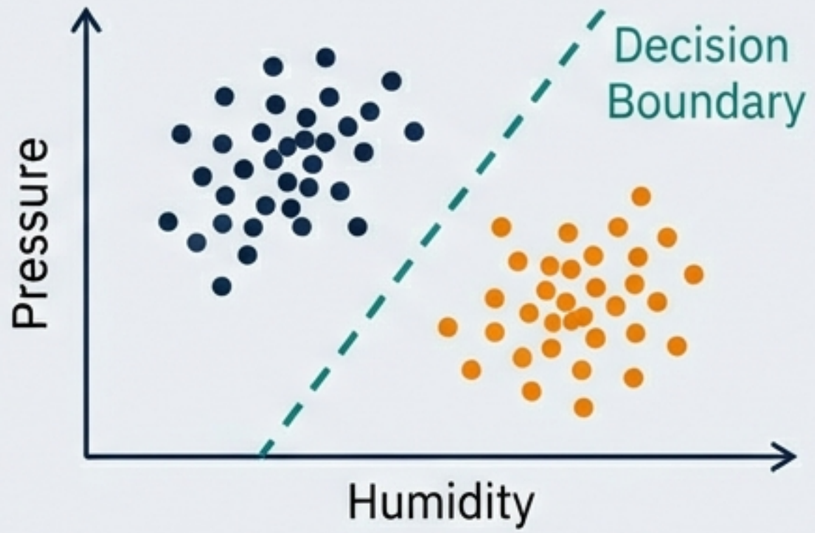
Target variable is continuous (e.g., Temperature, Revenue).



`Output: Numerical Value`

Predicting a Category

Target variable is discrete/categorical (e.g., Rain/No Rain, Benign/Malignant).



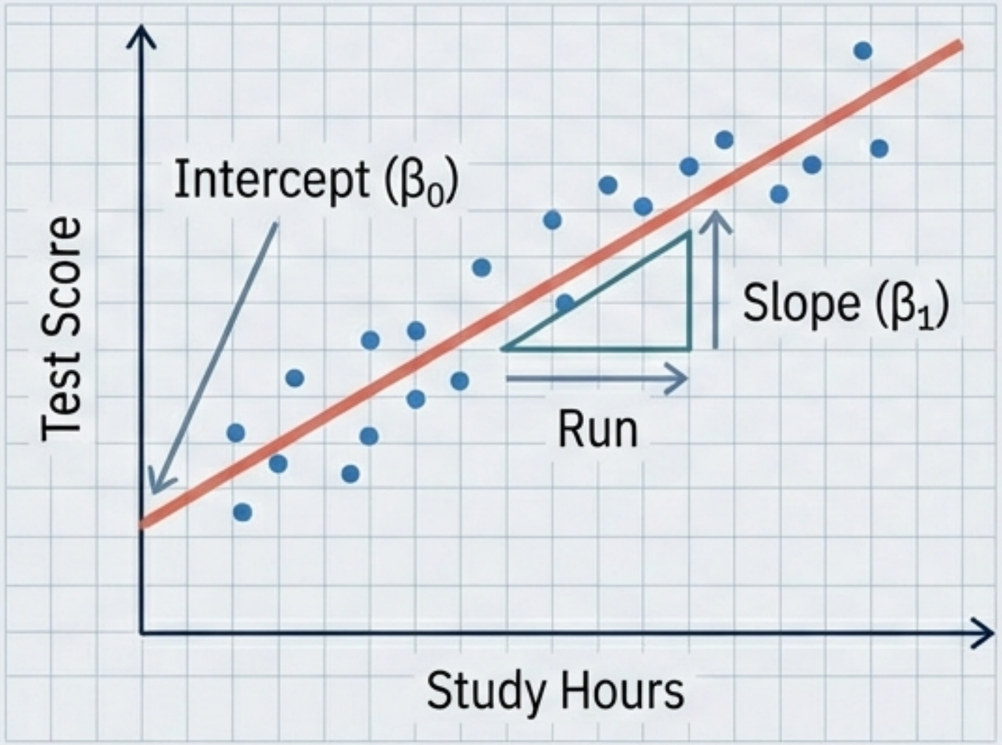
`Output: Class Label / Probability`

The nature of ‘y’ determines the mathematical path.

The Linear Approach: Modeling Continuous Relationships

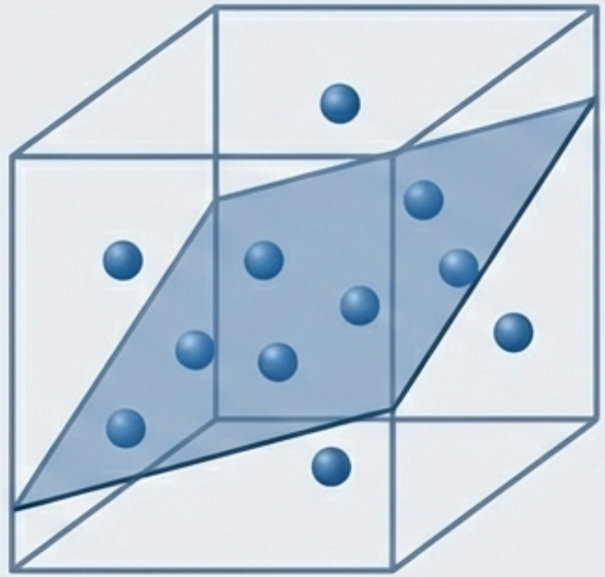
Simple Linear Regression (SLR)

$$y = \beta_0 + \beta_1 x + \varepsilon$$



Multiple Linear Regression (MLR)

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \varepsilon$$



****Ceteris Paribus*:** The coefficient reflects the change in y for a one-unit change in a specific predictor, holding all others constant.

The Logistic Approach: Modeling Probability

The Concept

Problem: Linear regression produces values > 1 or < 0, invalid for probability.

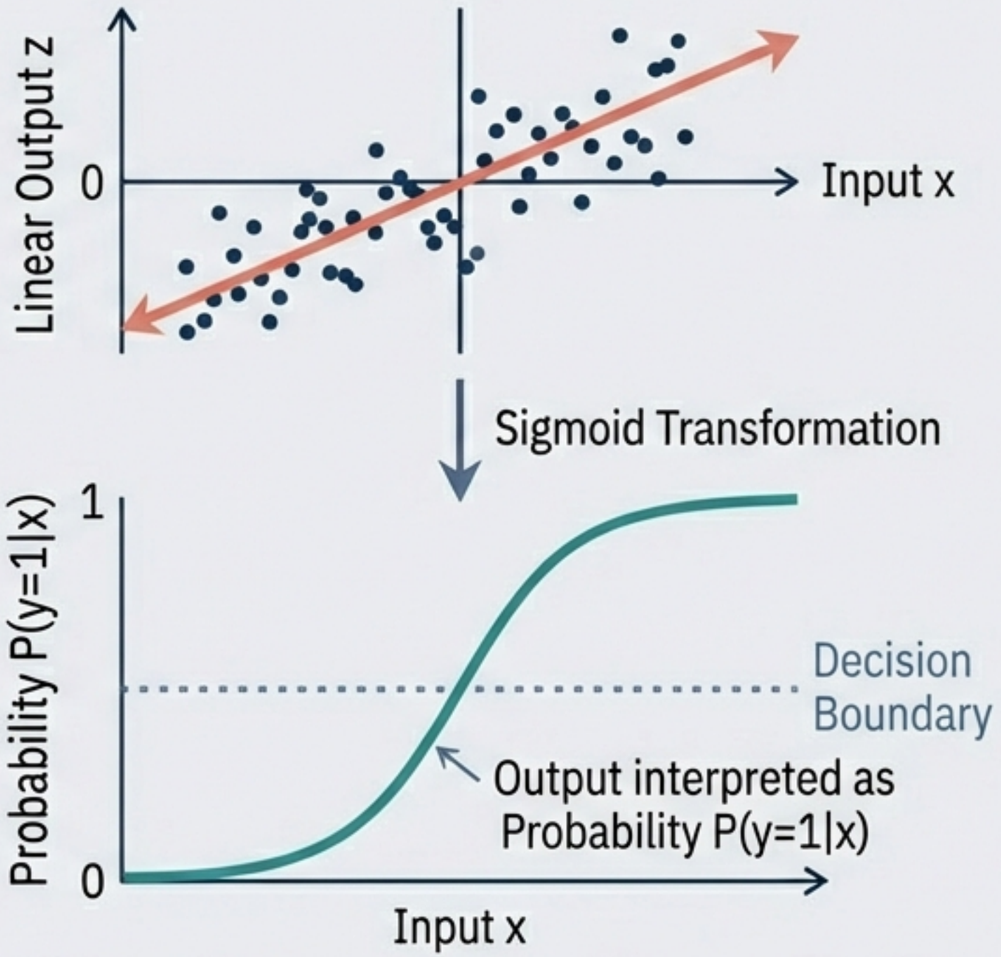
Solution: The Sigmoid Function "squashes" output between 0 and 1.

$$f(z) = \frac{1}{1 + e^{-z}}$$

The Logit: Models the Log-Odds of an event:

$$\ln\left(\frac{p}{1-p}\right) = \beta^T x$$

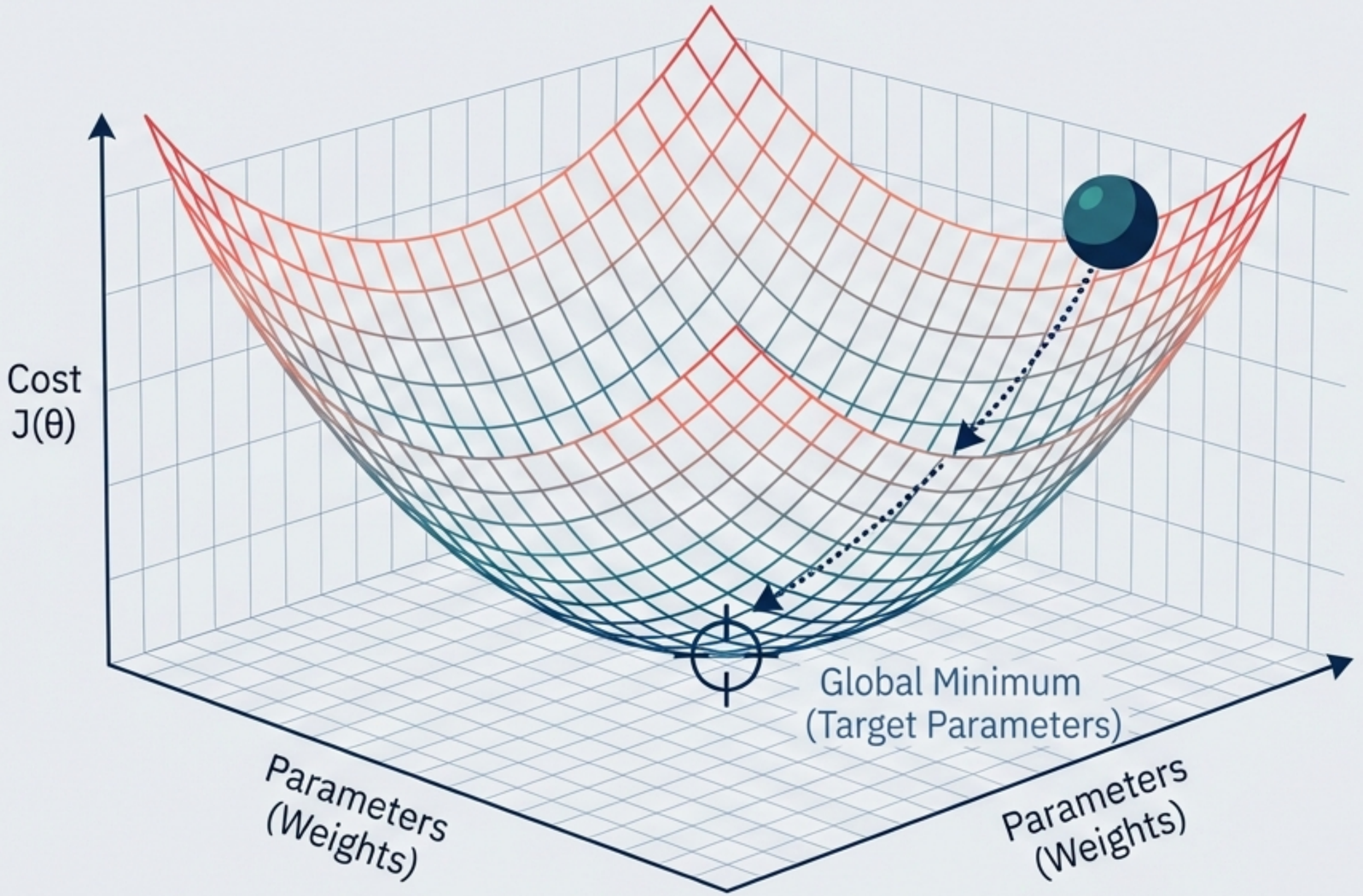
The Visual Transformation



Head-to-Head: Linear vs. Logistic Regression

Feature	Linear Regression	Logistic Regression
Dependent Variable	Continuous (Scale/Interval)	Categorical (Binary/Ordinal)
Visual Icon		
Output Format	Specific Value (e.g., Price $350k)	Probability (e.g., 76% Churn Risk)
Relationship Shape	Straight Line / Plane	S-Shaped (Sigmoid) Curve
Distribution	Gaussian (Normal)	Binomial
Optimization Target	Minimize Sum of Squared Errors (`Least Squares`)	Maximize Likelihood (MLE)

The Learning Engine: Gradient Descent & Optimization



- **The Goal:**
  Minimize the Cost Function (J) - the measure of prediction error.

- **Algorithm Steps:**
  1. Initialize weights randomly.
  2. Calculate gradient (slope).
  3. Update weights to move ‘downhill’.
  4. Repeat until convergence.

- **Key Parameter:**
  Learning Rate (α): Controls step size. Too small = slow. Too large = IBM. Ple = overshooting.

Evaluating Quality: Regression Performance Metrics

MAE (Mean Absolute Error)

IBM Plex Mono $\frac{\sum|y-\hat{y}|}{n}$

Average of absolute differences.

Pros:

- Robust to outliers.
- Highly interpretable (same units as Y).

MSE (Mean Squared Error)

IBM Plex Mono $\frac{\sum(y-\hat{y})^2}{n}$

Average of squared differences.

Pros:

- Heavily penalizes large errors.
- Good for mathematical optimization.

RMSE (Root Mean Squared Error)

IBM Plex Mono $\sqrt{MSE}$

Square root of MSE.

Pros:

- High penalty for large errors, but returns metric to original units of Y.

R-Squared (R^2)

IBM Plex Mono $1-\frac{SS_{\_res}}{SS_{\_tot}}$

Coefficient of Determination.

Pros:

- Explains proportion of variance (0 to 1). Note:
- Use Adjusted R^2 to penalize useless features.

Diagnostics: Assumptions & Multicollinearity

The L.I.N.E. Assumptions

- **Linearity:** “Relationship between X and Y is linear.”
- **Independence:** “Residuals are independent (no autocorrelation).”
- **Normality:** “Residuals are normally distributed (Bell curve).”
- **Homoscedasticity:** “Residuals have constant variance (no ‘cone’ patterns).”

Deep Dive on Multicollinearity

When predictors are highly correlated, distorting coefficients.

Visual Tool:

	Var A	Var B	Var C	Var D	Var E
Var A	1.0	0.87	0.97	0.26	0.20
Var B	0.97	1.0	0.95		0.82
Var C	0.94	0.91	1.0	0.95	0.97
Var D	0.25		0.95	1.0	0.20
Var E	0.20	0.97	0.85	0.27	1.0

High Correlation Warning

Metric:
VIF (Variance Inflation Factor): Values > 5 or 10 indicate trouble.

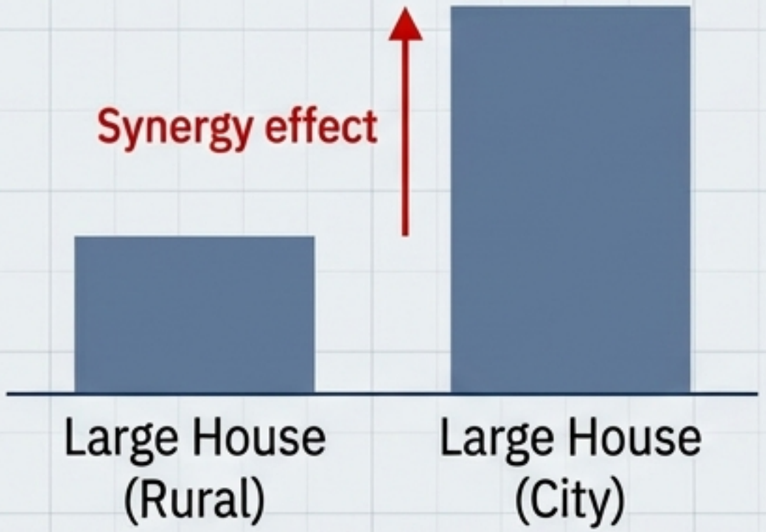
Enhancing the Model: Feature Engineering

Creating new features from existing data to capture complexity and non-linearity.

A: Interaction Terms

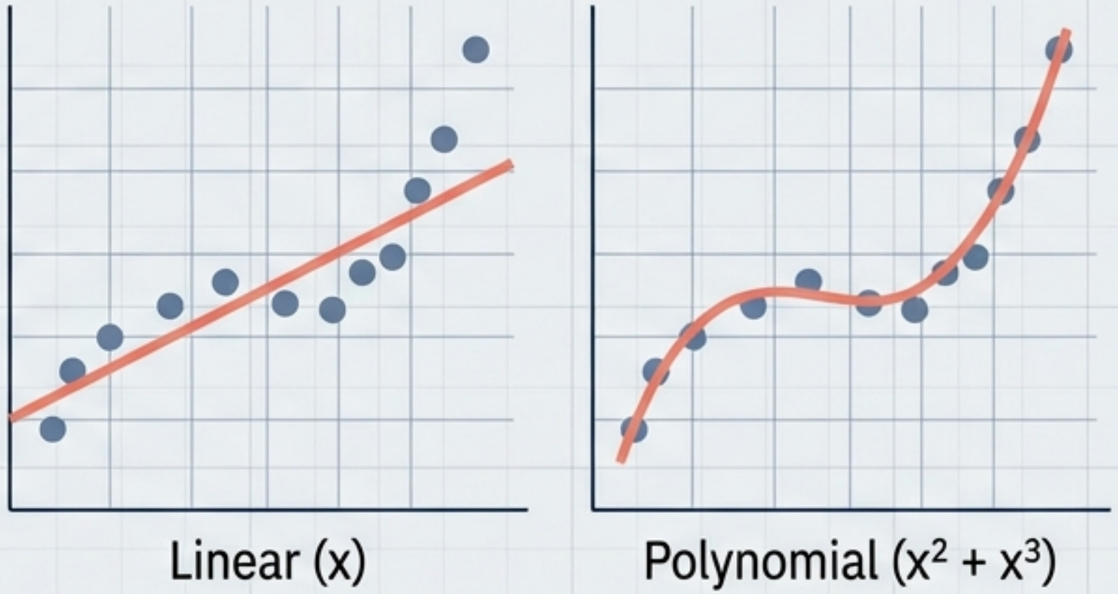
$$\text{Price} = \beta_1\text{Size} + \beta_2\text{Location} + \beta_3(\text{Size} \times \text{Location})$$

Captures when the impact of one variable depends on another.



B: Polynomial Features

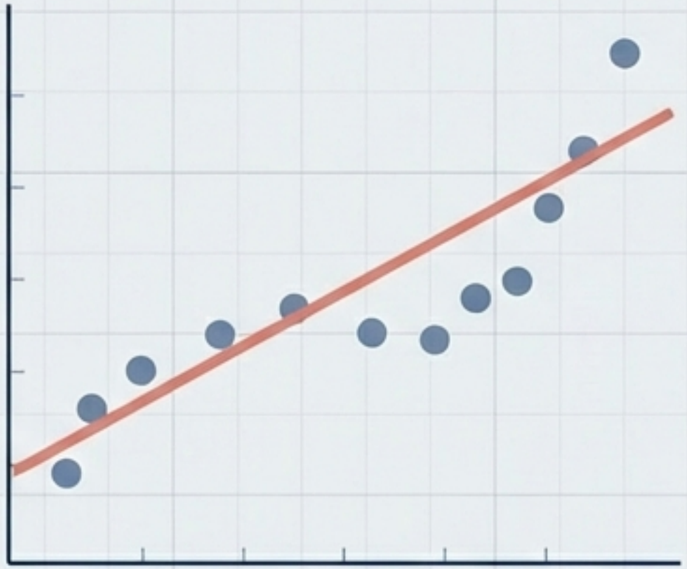
$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$



The Balancing Act: Complexity vs. Accuracy

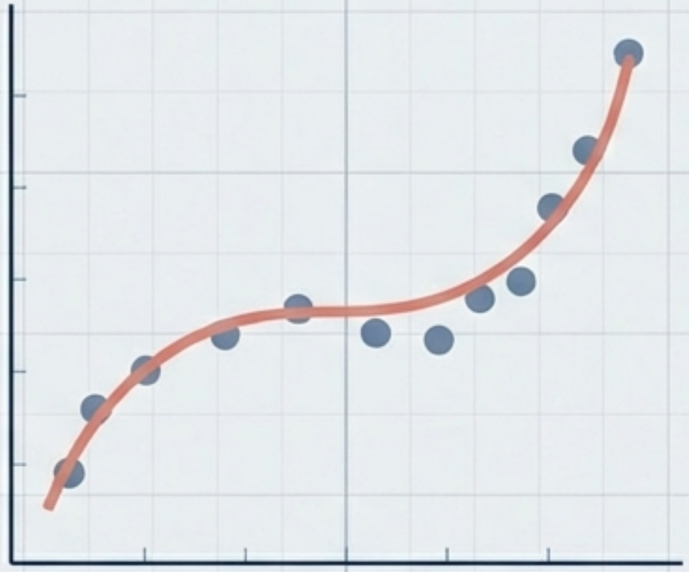
The Bias-Variance Tradeoff

Too Simple



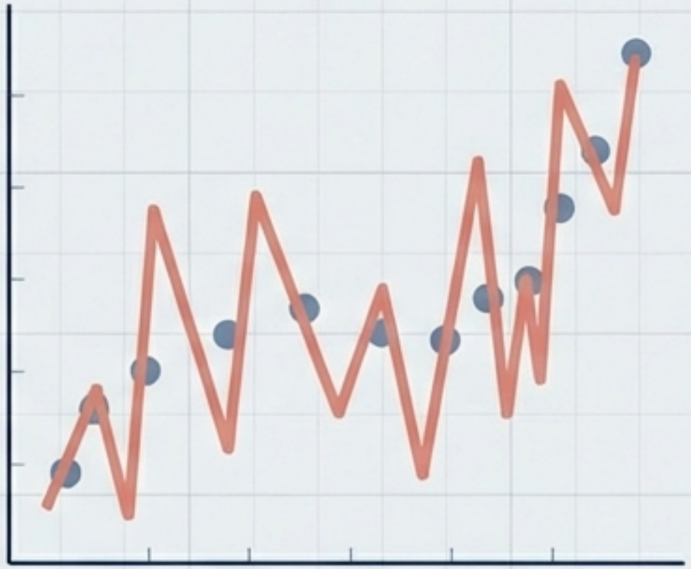
High Bias / Low Variance

The Sweet Spot



Balanced

Too Complex



Low Bias / High Variance

Goal: Generalization—Performance on unseen data.

The Guardrails: Regularization Techniques

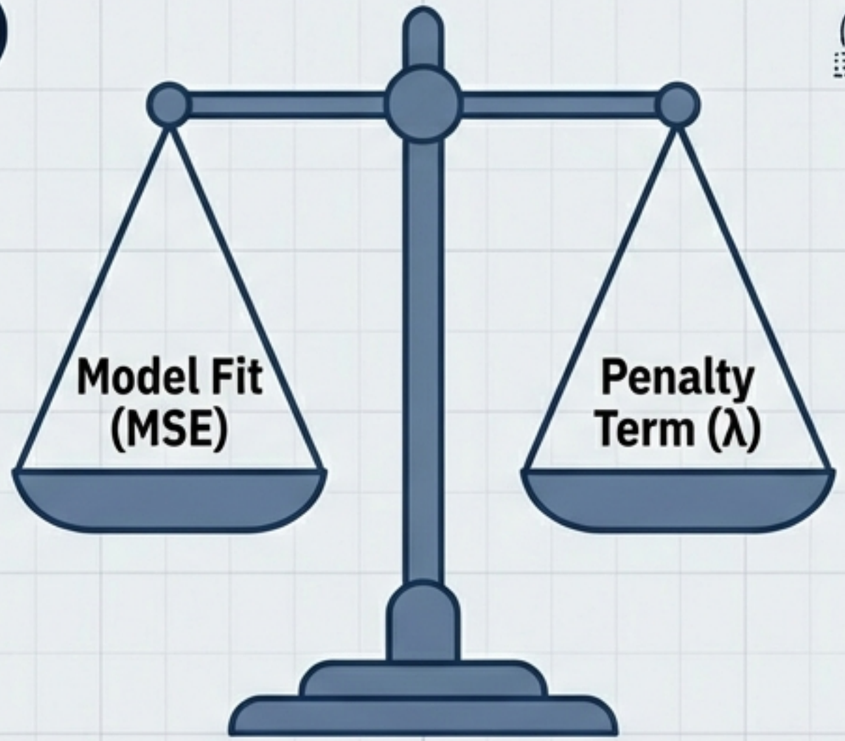
Preventing overfitting by adding a penalty term to the cost function.

Lasso Regression (L1)

Penalty: Absolute Value of Coefficients ($|\beta|$)

Effect: Can shrink coefficients to **Zero**.

Superpower: Automatic Feature Selection (Sparse Models).



Ridge Regression (L2)

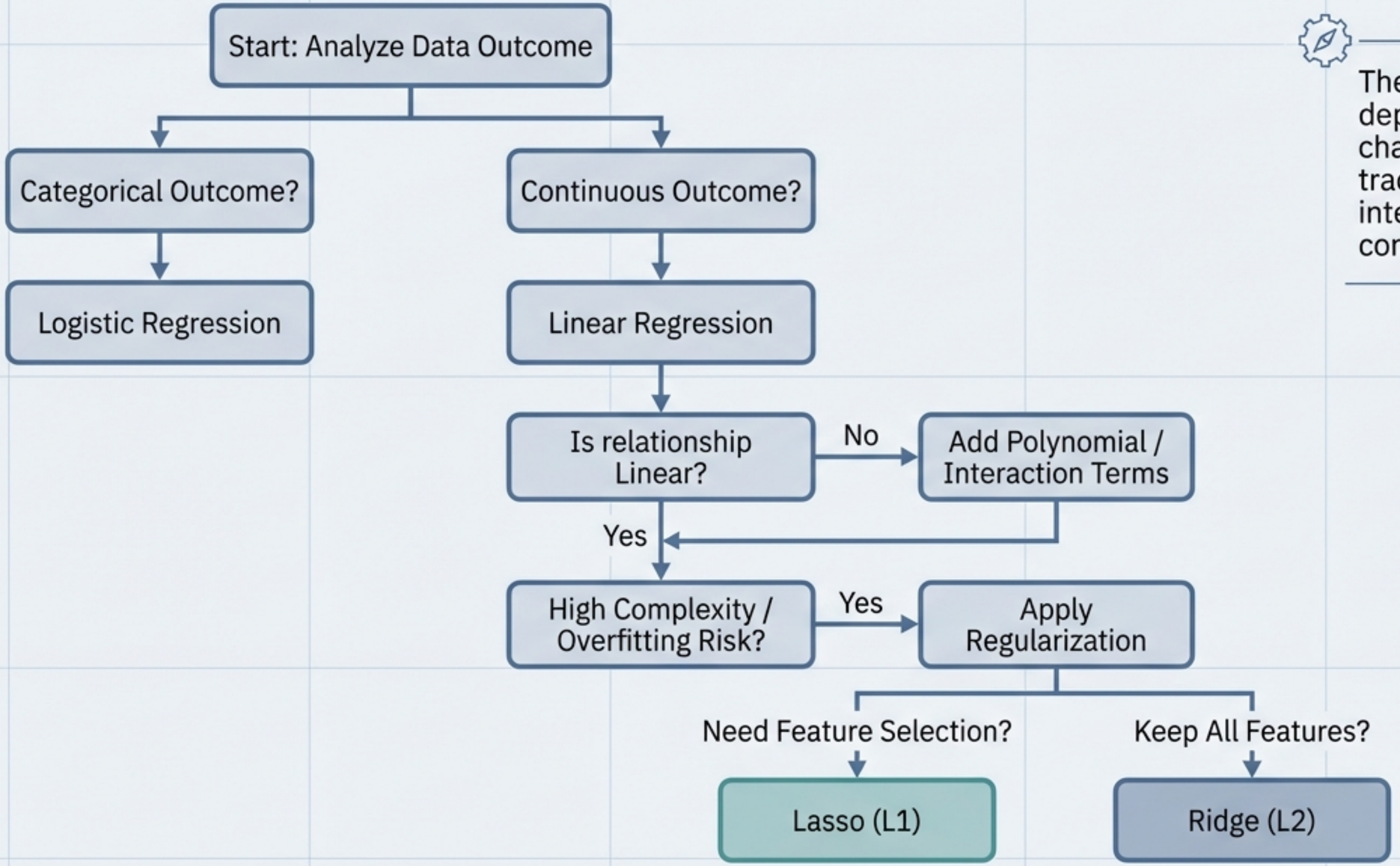
Penalty: Squared Value of Coefficients (β^2)

Effect: Shrinks toward zero, but never reaches it.

Superpower: Handles Multicollinearity; keeps all features.

Elastic-Net: Hybrid of L1 and L2.

Strategic Choice: Selecting the Right Model



The choice of architecture depends on data characteristics and the trade-off between interpretability and complexity.

Conclusion: The Power of Interpretability

- **Bedrock of Inference:** Regression links inputs to outputs through rigorous mathematical optimization.
- **Transparency:** From Simple Linear Regression to complex Regularization, these tools offer "White Box" transparency that deep learning often lacks.
- **Mastery:** True predictive power lies in validating assumptions (L.I.N.E.), engineering features, and balancing bias vs. variance.