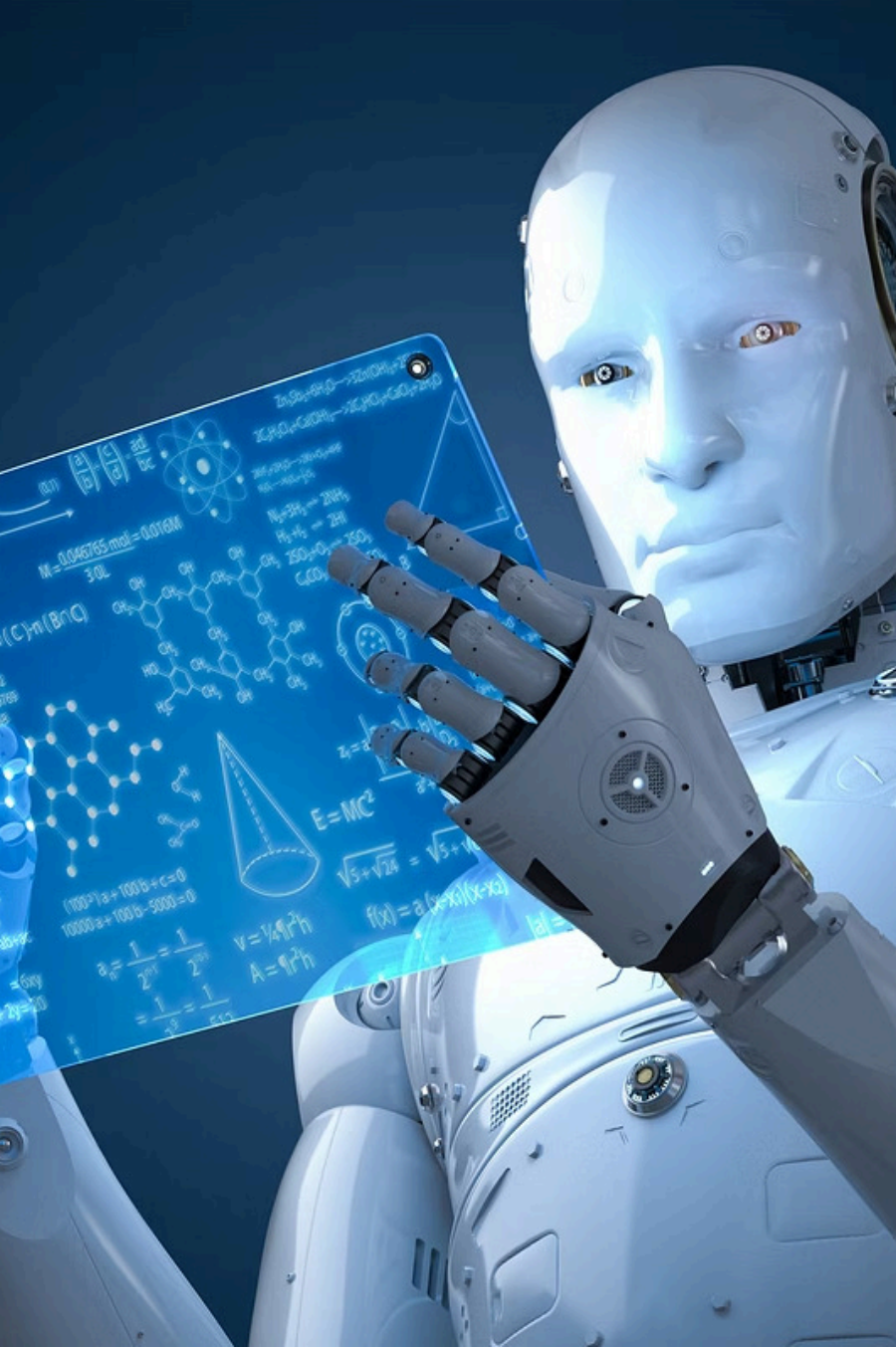




Introduction to Machine Learning: Understanding Supervised Learning

Welcome to the first of two sessions on machine learning fundamentals. Today we'll explore supervised learning, focusing on linear regression as our foundation.

What Is Machine Learning?

Machine Learning Rule Discovery

Computers learn rules by analyzing data to find patterns.

Spotify Music Recommendations

Spotify analyzes listening habits to make personalized music suggestions.

Turning the Real World Into Numbers

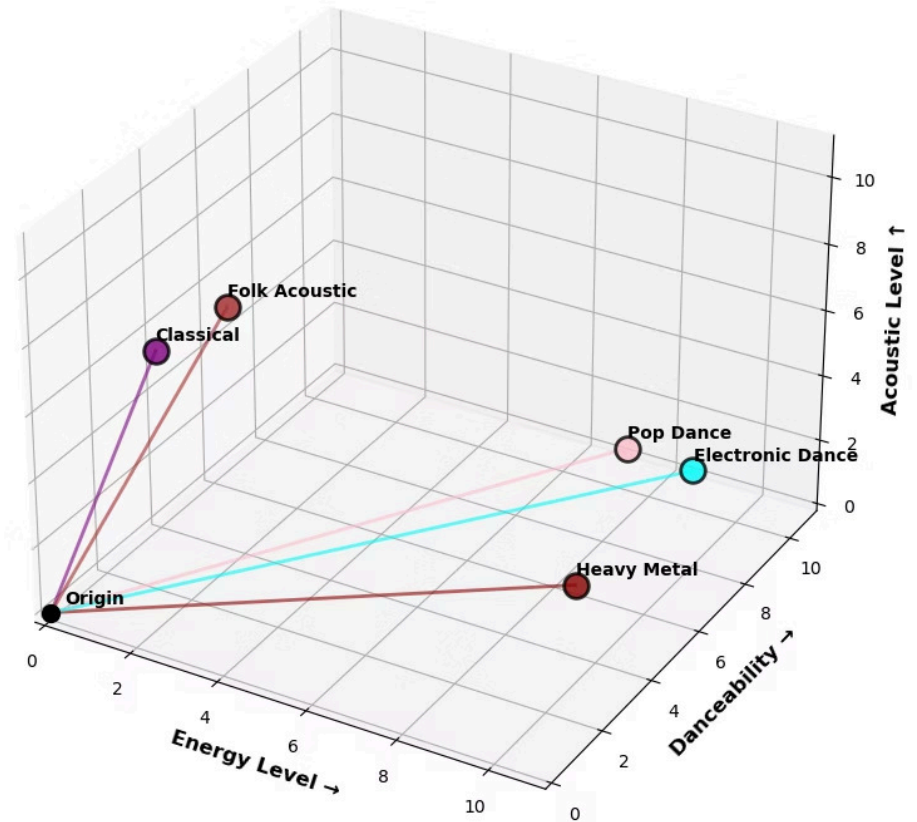
Computers understand numbers, not raw data. We transform real-world things into **features** - numerical values that represent characteristics. Each object becomes a list of numbers called a **vector**.

Songs as Vectors Example

- Song A: [energy: 8, danceability: 6, acoustic: 2]
- Song B: [energy: 3, danceability: 4, acoustic: 9]

Each song becomes a point in 3D space. Songs that sound similar will have similar vectors and sit close together on this invisible map.

Multiple Songs in 3D Vector Space
(Notice how similar genres cluster together)



From Data to Patterns - The Movie Example

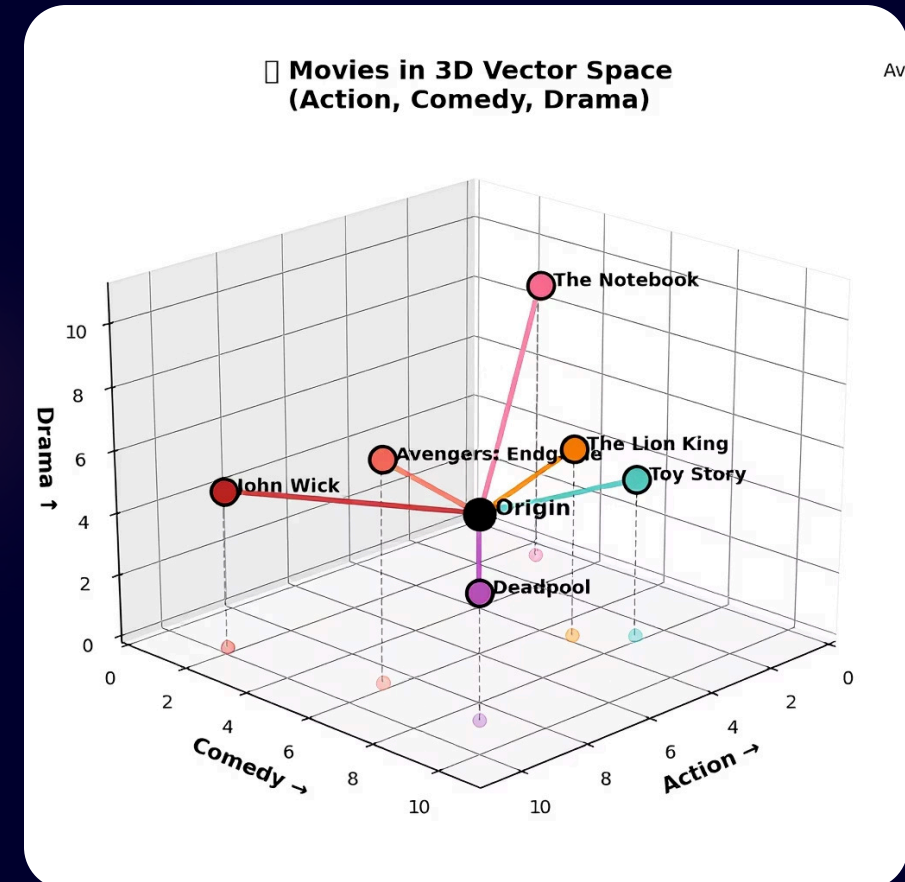
Movie Features

We'll describe movies using five features, each rated from 1 to 10:

- Action – fighting, chases, explosions
- Comedy – funny or lighthearted
- Drama – emotionally intense
- Child-Friendliness – appropriate for kids
- Romance – focus on love stories

Movie Vector

Movie	Action	Comedy	Drama	Child-Friendly	Vector
Avengers: Endgame	9	6	7	7	[9, 6, 7, 7, 3]
John Wick	10	2	5	2	[10, 2, 5, 2, 1]
The Lion King	4	7	6	10	[4, 7, 6, 10, 4]
The Notebook	1	3	9	6	[1, 3, 9, 6, 10]
Deadpool	9	9	4	1	[9, 9, 4, 1, 5]



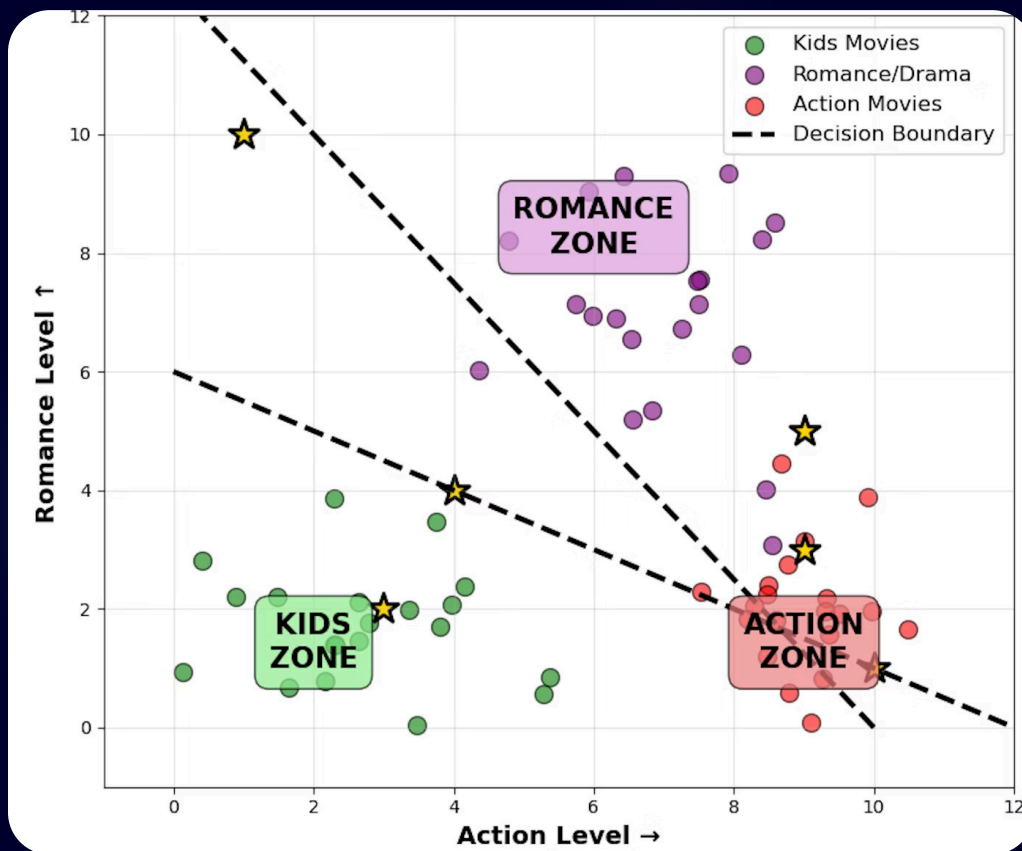
Movies represented as points in a vector space

Two Types of ML Problems

Classification → Predicts discrete categories or labels. *Draw boundaries to separate different groups.* Eg Is this movie family-friendly? (Yes/No)

Regression → Predicts continuous numerical values. Find the best line to predict values. Eg What box office revenue will this movie earn? \$50M?

Classification vs. Regression – Two Different Questions



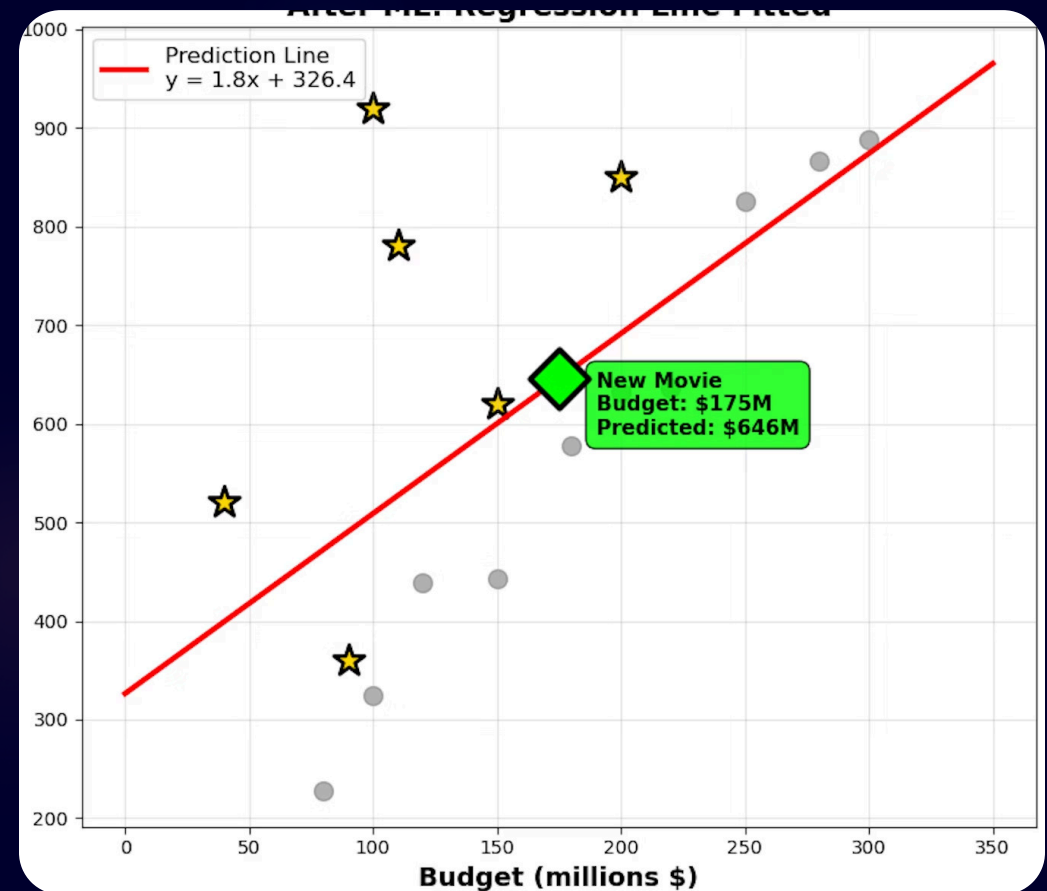
Classification: Which Group Does This Belong To?

Classification puts things into categories. For our movie example, we might sort movies into "Family Friendly" or "Not Family Friendly."

Behind the scenes, the computer analyzes labeled training examples:

- Lion King [4,7,6,10,4] → "Family Friendly"
- John Wick [10,2,5,2,1] → "Not Family Friendly"

The algorithm draws boundaries in the feature space so that similar movies end up on the same side.



Regression: What's the Exact Number?

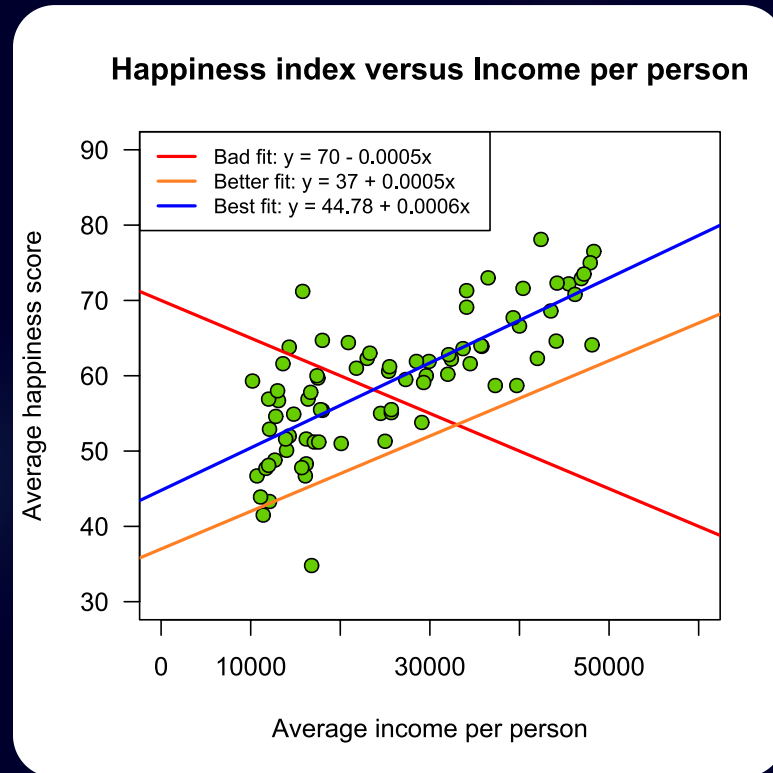
Regression predicts a specific value. For movies, we might predict box office revenue based on features.

We feed the computer data like:

- Avengers [9,6,7,3] → \$2.8 billion
- Lion King [4,7,6,4] → \$1.6 billion

The algorithm finds a mathematical formula that connects the feature values to the revenue prediction. The key difference: Classification draws boundaries between regions, while regression maps from feature space to a predicted value.

Linear Regression - The Foundation of Machine Learning



Why we're starting with linear regression:

- Foundation that underlies virtually every ML algorithm
- Principles you learn here apply to neural networks and recommendation systems
- Gives you interpretability - you can trust and understand the relationships

Linear regression is everywhere:

- Banks: Predict loan defaults
- Netflix: Estimate movie preferences
- Scientists: Model climate change
- Businesses: Forecast sales

How it works: Looks at hundreds of examples with known outcomes. Sees how features relate to actual results. Discovers the mathematical relationship. Uses that relationship to predict new cases. The goal is to find the line that best fits the data points.

Precious benefit is Interpretability : When it tells you increasing action level by 1 point increases revenue by \$300 million, you can trust and understand that relationship

Linear regression can be understood through two complementary lenses:



Geometric Perspective

Seeing how machine learning transforms your spreadsheets into points in space.



Mathematical Machinery

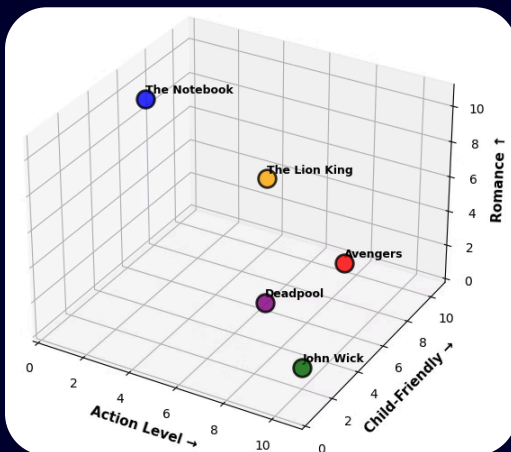
Then we'll dive into the mathematical machinery that finds the optimal solutions.

How Machine Learning Sees Data - The Geometric View

Turning Data into Geometric Spaces

- Remember our movie examples? Each movie's features (action level, comedy level, drama level) are like coordinates
- These coordinates pinpoint the movie's exact location in a geometric space
- Instead of seeing spreadsheet rows, we see points floating in mathematical space

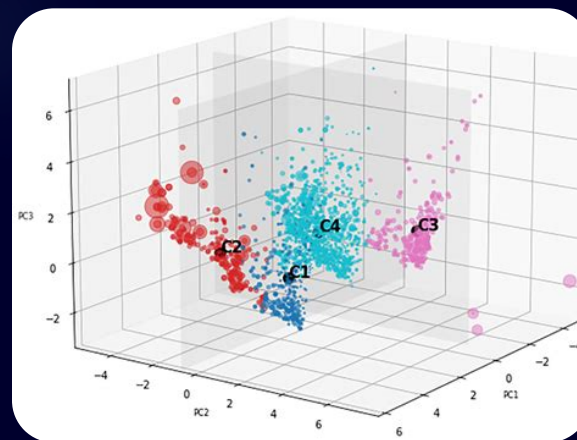
Vectors in 3D Space



Adding More Dimensions

- Adding more features like romance, child-friendliness, and runtime expands this space
- Each additional feature requires another coordinate to describe the movie's location
- This creates a multi-dimensional "feature space" where each movie exists as a point
- We can work in any number of dimensions - 5D, 10D, or even 100D

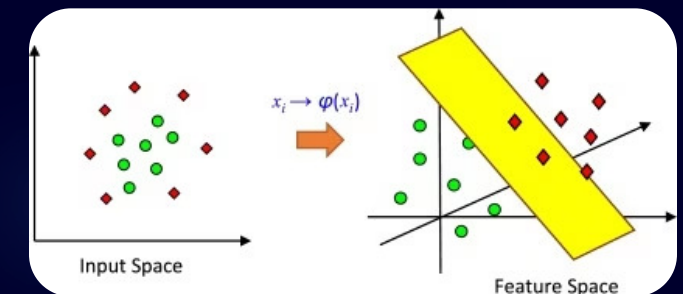
Vectors in 4D Space



The Two-Space Reality

- Machine learning works by learning the connection between two separate realms
- **Feature space:** Where your movie data resides as coordinate points
- **Output space:** Where predictions like revenue are made
- Linear regression learns to map from feature coordinates to prediction values
- It's like having a mathematical GPS between two different worlds

Input (Feature) Space vs Output (Label) Space



What Lines Actually Do - The Mapping Function

The Universal Truth: ML is fundamentally about dimensional mapping

- 10D → 500D: Features to predictions
- 784D → 10D: Image pixels to digit classification
- 5D → 1D: Movie features to revenue (our example)

Understanding the Mapping Process

Let's make this concrete with our movie example:

1. Avengers exists as point [9,6,7,7,3] in 5D feature space
2. We learn a mapping function from training data: [8,3,2,5,1]
3. We apply the mapping using dot product: $9 \times 8 + 6 \times 3 + 7 \times 2 + 7 \times 5 + 3 \times 1 = 142$
4. Add base amount: Revenue = $142 + 10 = \$152\text{M}$

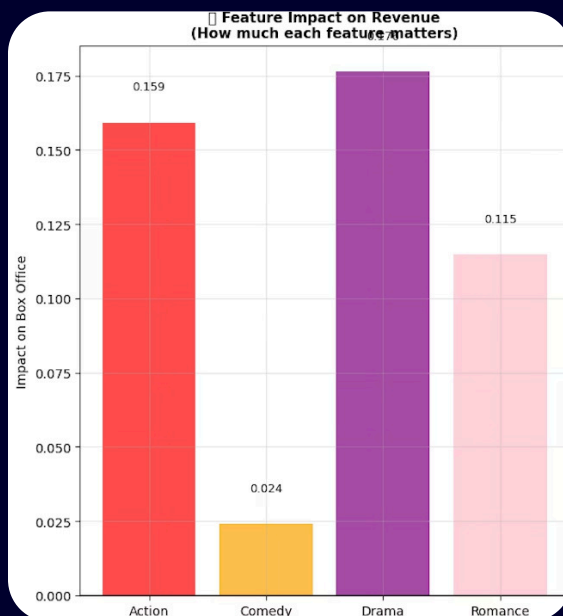
What This Mapping Means

The weights [8,3,2,5,1] ARE our learned mapping function - they define how to transform any 5D movie vector into a 1D revenue prediction.

Each coefficient tells you how much that feature contributes:

- If action coefficient = \$8M, every 1-point increase in Action adds \$8M to revenue prediction
- If romance coefficient = \$1M, romance has less impact on revenue

The mapping function emerges from patterns in the data - if action movies consistently make more money, the algorithm learns to weight action more heavily.



Universal Framework:

Linear regression: Linear mappings

Neural networks: Complex non-linear mappings

All ML algorithms learn dimensional mappings

The Bridge Between Geometry and Mathematics

The geometric and mathematical views of linear regression are two sides of the same coin, both aiming to find the optimal line of best fit through an identical underlying process.

Geometric Perspective

This view seeks the optimal mapping direction in multi-dimensional feature space that best projects data points onto the one-dimensional output (e.g., revenue) line.

Error is the direct distance between the predicted point and the actual point in the revenue space.

Mathematical Perspective

This view identifies the line coefficients that minimize the sum of squared prediction errors (the difference between actual and predicted values).

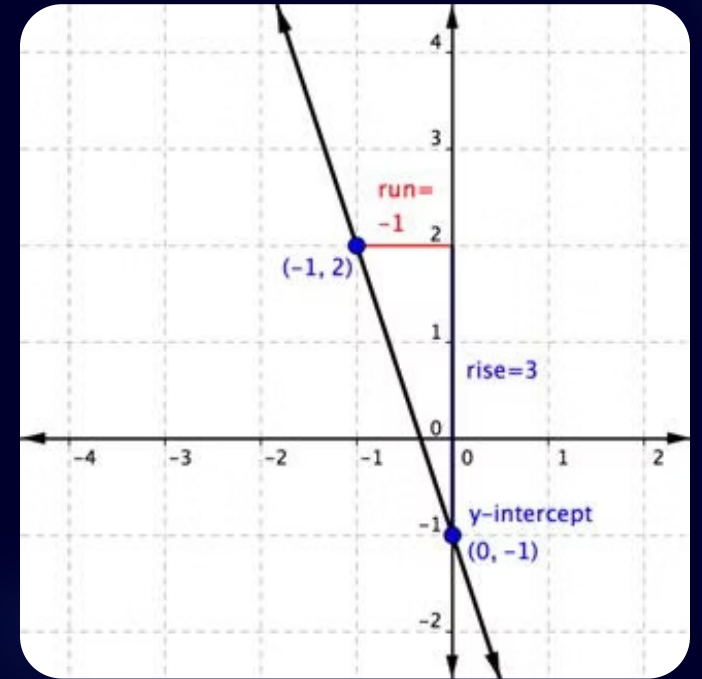
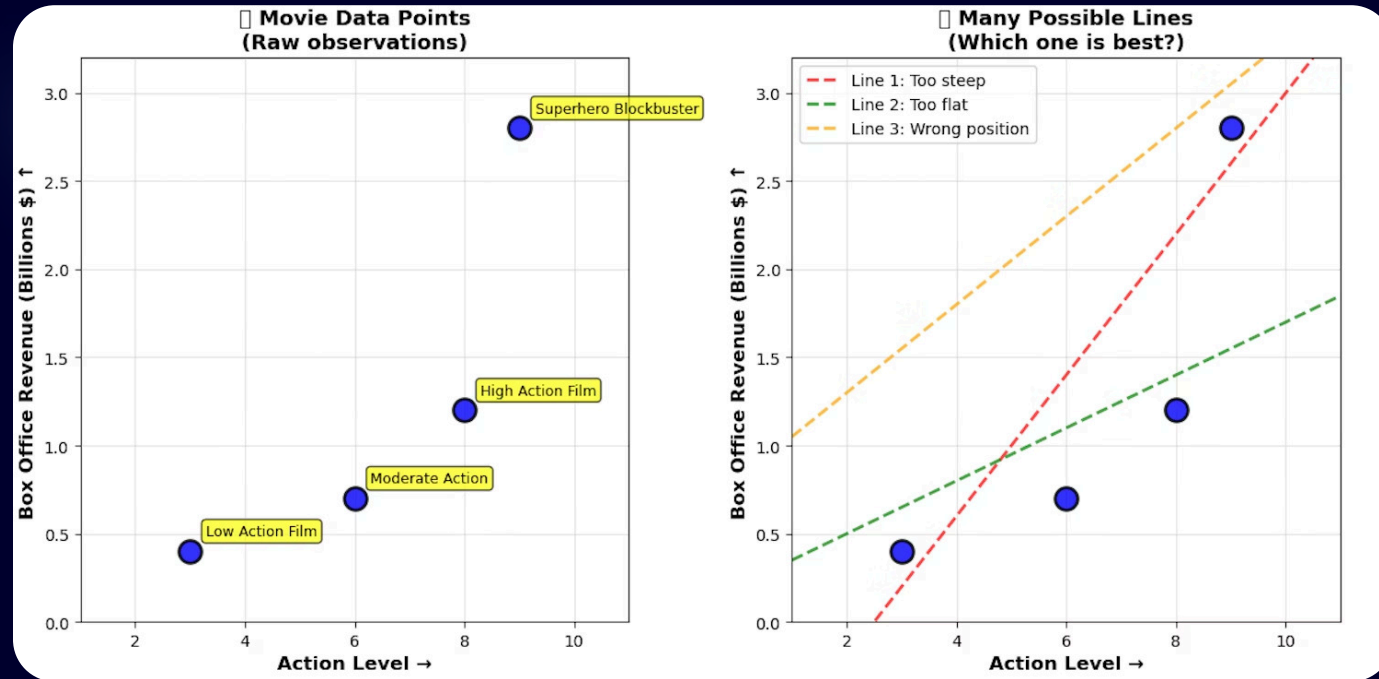
Error is the numerical difference between the actual and predicted values, squared to quantify divergence.

The Connection

Both perspectives are fundamentally seeking the same optimal solution: the specific vector or set of coefficients that best describes the data's underlying relationship. Mathematical error precisely quantifies this geometric distance.

The Challenge - Which Line is Best?

In linear regression, finding the best-fitting line is crucial. With infinite possibilities, determining the optimal slope and intercept involves balancing impact and accuracy. Discover the search for the perfect line.



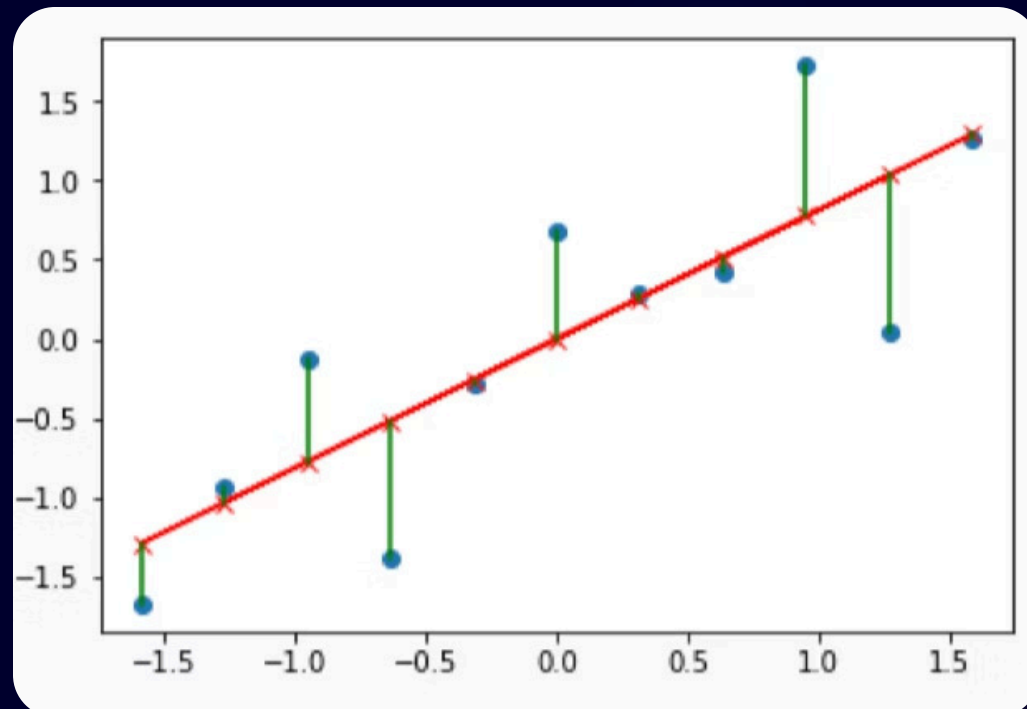
- **Adjusting intercept (b)** = translating the line up or down
- **Adjusting slope (m)** = rotating the line steeper or flatter

Finding the Best Line Through Optimization

Linear regression aims to find the "best" mapping or line of best fit that minimizes prediction errors from an infinite number of possibilities.

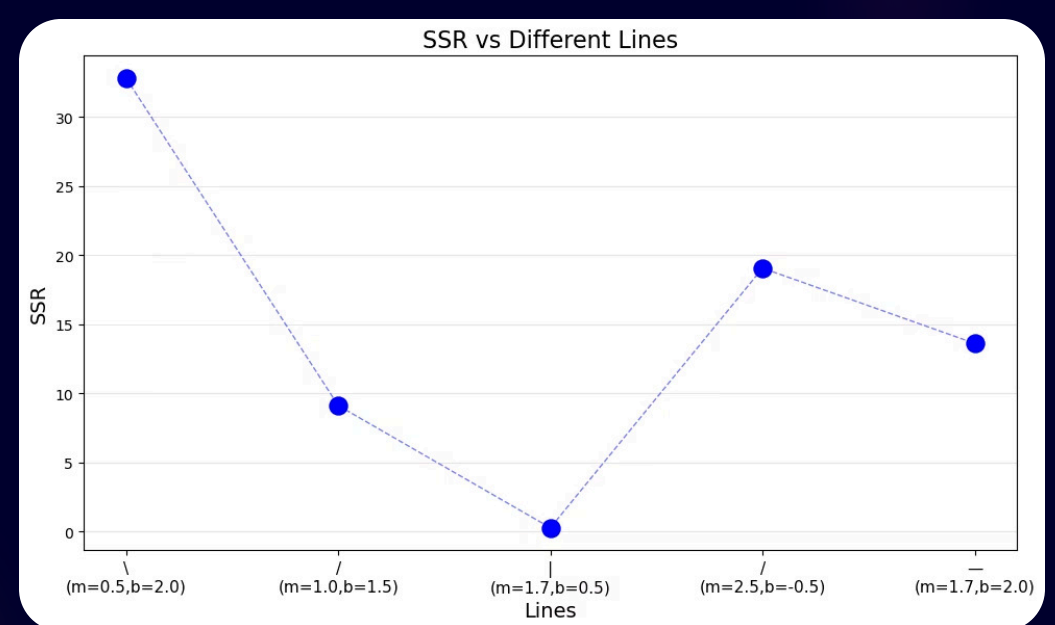
Defining "Error": The Gap Between Prediction and Reality

- **What is error?** Error (or residual) quantifies how "wrong" a prediction is - the numerical difference between what our model predicted and what actually happened
- **The calculation:** Error = Actual Revenue - Predicted Revenue (can be positive if we predicted too low, negative if too high)
- **Geometric visualization:** Each error is the vertical distance between a data point and our mapping line - you can see these gaps in the plot
- **Why errors matter:** They tell us how well our learned mapping is working - small errors mean good predictions, large errors mean our mapping needs improvement

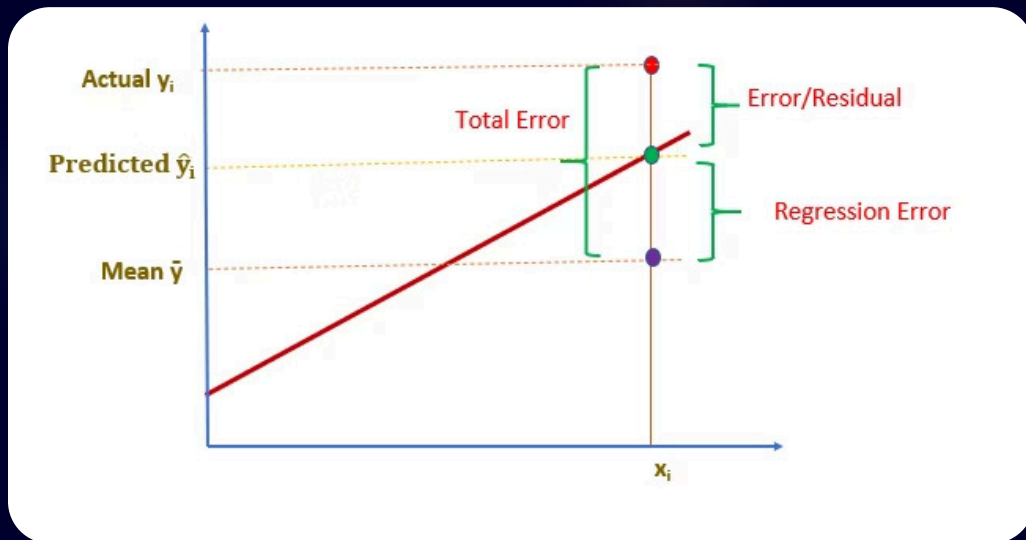


To find the "best" line, we use Mean Squared Error (MSE) to score each possible mapping.

- **What we're measuring:** The vertical distance between each data point and our line - these are our prediction errors
- **Why square errors?** Makes them positive and punishes big mistakes more than small ones - creating our Sum of Squared Residuals (SSR)
- **The testing process:** Algorithm tries different slopes (m) and intercepts (b) - each combination creates a different line through our data points
- **Each line gets an SSR score:** Different m and b values produce different total squared errors
- **The optimization goal:** Find the m and b that give us the lowest SSR - that's our best-fitting line



Going Deeper: What Can Our Line Explain?



When our model makes a prediction, we can break down the story into two parts: what our model explains and what it can't.

Three Key Reference Points

- **Actual value (Y):** Where the data point truly lands
- **Prediction ($\hat{Y}_i$):** Where our model says it should land
- **Average ($\bar{Y}$):** The mean of all values - our baseline

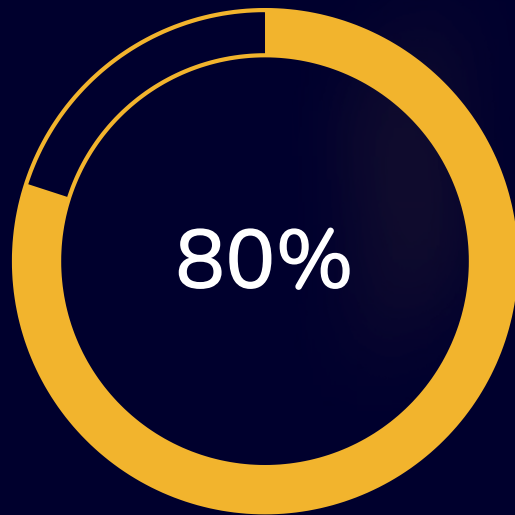
Breaking Down the Prediction

1. **Explained deviation:** Prediction - Average ($\hat{Y} - \bar{Y}$) What our model explains using features
2. **Unexplained deviation:** Actual - Prediction ($Y - \hat{Y}$) What our model can't explain (error)

Example: For a day with temperature 75°, actual sales 60 cones, predicted sales 50 cones, and average sales 40 cones:

- Explained: $50 - 40 = 10$ cones (explained by temperature)
- Unexplained: $60 - 50 = 10$ cones (due to other factors)

Measuring Model Performance with R-Squared

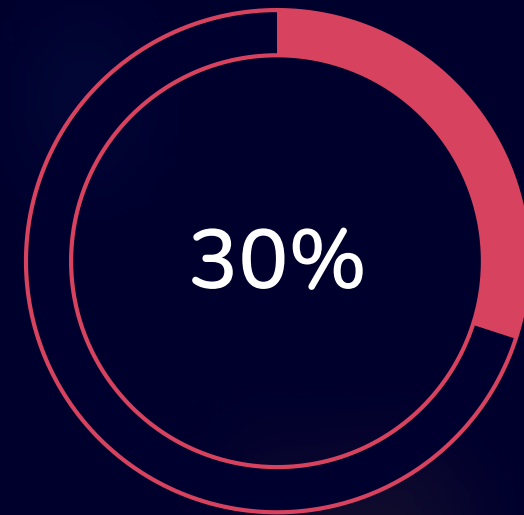


High R-Squared

Model explains 80% of variation in data

Only 20% remains unexplained

The temperature-to-sales mapping is working well



Low R-Squared

Model explains only 30% of variation

70% is still unexplained

May need more features like humidity, day of week, or local events

Sum of Squares Regression (SSR)

Total variation our model explains

Sum of all $(\text{Prediction} - \text{Average})^2$ across all data points

1

2

Sum of Squares Error (SSE)

Total variation our model cannot explain

Sum of all $(\text{Actual} - \text{Prediction})^2$ across all data points

Sum of Squares Total (SST)

Total variation in our data

$SST = SSR + SSE$

3

4

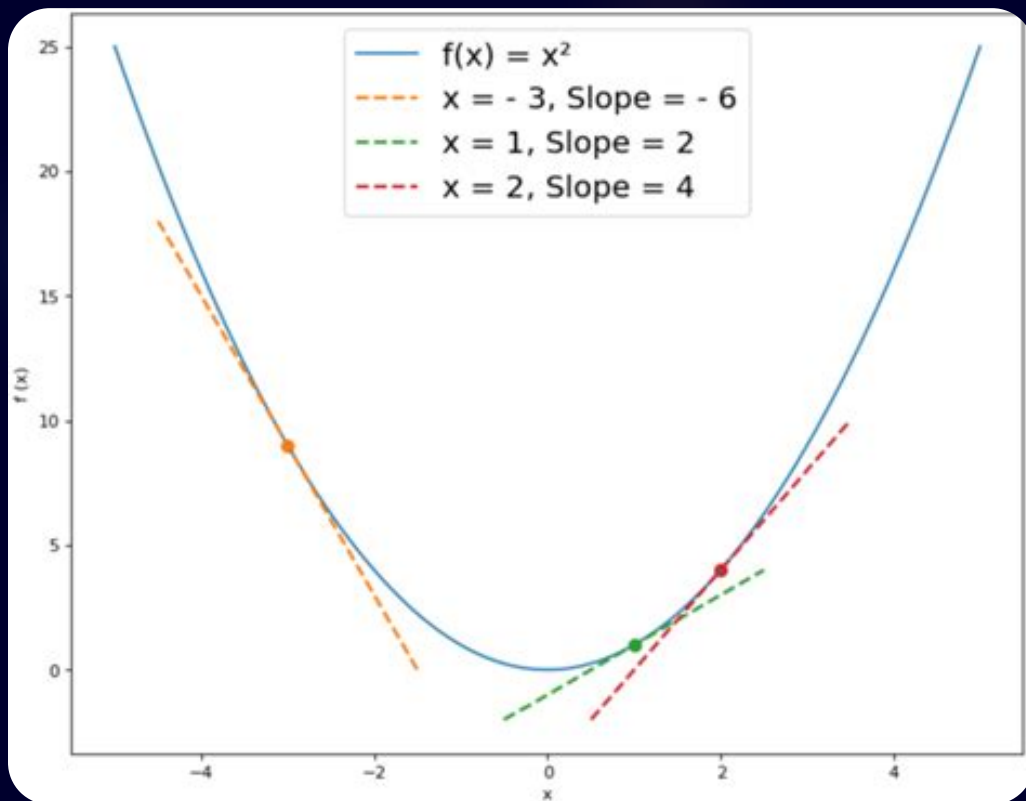
R-Squared

Proportion of variation explained by our model

$R^2 = SSR \div SST$

Always between 0 and 1

How Linear Regression Actually Learns - Gradient Descent



We know we need to find the best line by minimizing squared errors, but how does the computer actually find it?

What Makes a Line?

Every line is defined by just two numbers:

- **Slope:** How steep the line is
- **Intercept:** Where the line crosses the y-axis

The Gradient Descent Process

Gradient descent is like being dropped blindfolded on a mountainside and trying to find the bottom of the valley:

1. Start with an initial guess (e.g., slope=0.5, intercept=0)
2. Calculate the sum of squared errors (SSE) for this line
3. Calculate the derivative - the steepness of the error landscape
4. Take a small step downhill in the direction that reduces error
5. Repeat until you reach the bottom (minimal error)

Gradient Descent in Action - Step by Step Learning

1 Initial Parameters

Start with slope=0.5, intercept=0

Height = $0 + 0.5 \times \text{weight}$

Calculate SSE = 4.2

2 Adjust Intercept

Derivative shows: "Increase intercept to reduce error"

New intercept = 0.3

Height = $0.3 + 0.5 \times \text{weight}$

Recalculate: SSE = 3.7 (improved!)

3 Adjust Slope

Derivative shows: "Increase slope to reduce error more"

New slope = 0.6

Height = $0.3 + 0.6 \times \text{weight}$

Recalculate: SSE = 3.1 (even better!)

4 Iterate Until Convergence

Continue taking small steps in the direction that reduces error

Eventually reach a point where derivatives are nearly zero

Found the optimal line for our data!

This gradient descent process is how every machine learning algorithm learns from data. Whether it's linear regression or neural networks, they all follow the same pattern: measure errors, adjust parameters to reduce errors, and repeat until converged.

Evaluation Metrics - R-Squared and Error Measures

Error-Based Metrics

- **Mean Squared Error (MSE):** Average of squared errors $MSE = 75$ squared cones
- **Root Mean Squared Error (RMSE):** Square root of MSE $RMSE = 8.7$ cones
- **Mean Absolute Error (MAE):** Average of absolute errors $MAE = 7$ cones

R-Squared Calculation

Average sales = 45 cones

SSE (sum of squared errors) = 375

SST (total variation) = 675

$R\text{-squared} = 1 - (SSE/SST) = 1 - (375/675) = 0.44$

Our model explains 44% of the variation in ice cream sales.

Conclusion: The Foundation for Machine Learning

Data as Geometry

Computers see data as vectors - points floating in high-dimensional space. Movies become coordinates in action-comedy-drama space.

Next Steps

With this foundation, you can understand any machine learning algorithm as a different approach to finding patterns in high-dimensional space.

Performance Metrics

R-squared, RMSE, and MAE help us quantify explained vs. unexplained variation and evaluate model quality.

Two Core Problems

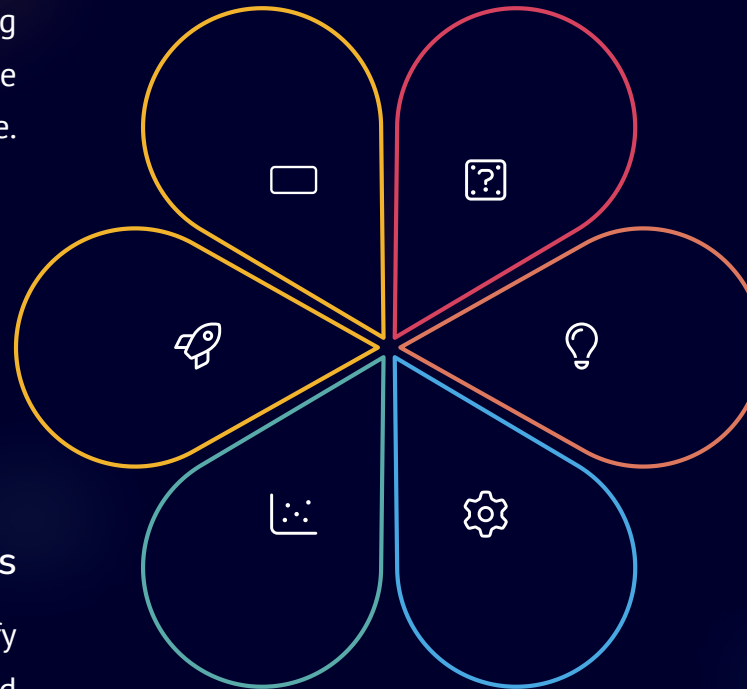
Classification draws boundaries between groups. Regression maps from input features to target values. Both find patterns in vector space.

Universal Principles

The geometric thinking and optimization principles you learned today apply to all machine learning algorithms.

Learning Process

All algorithms measure errors, use optimization to minimize those errors, and learn mappings between spaces.



You've demystified machine learning! It's not magic - it's systematic pattern recognition through mathematical optimization. The patterns are everywhere. Now you know how to find them.