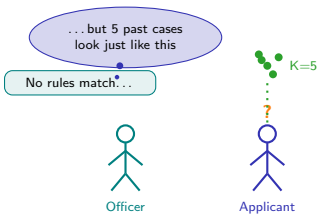
Why Would a Loan Officer Want to Ask the Neighbors?

The Dilemma

- A new applicant walks in with no credit history
- The rulebook says nothing about this profile
- But the database holds thousands of past applicants with known outcomes

What if the answer is already in the data – hiding among the neighbors?



Insight

KNN formalizes human reasoning: when rules fail, look at the most similar past cases and follow their lead.

Instance-based learning stores all training data – no model parameters, no training phase

Classifying a Stranger – Did Similarity Cross Your Mind?

Think Before You Compute

Imagine you meet someone at a conference. You know nothing about them. Within seconds, your brain classifies: academic or industry? Junior or senior? You did not run a logistic regression. You compared them to people you already know.

- How many "neighbors" did your brain consult?
- Did you weight closer acquaintances more heavily?
- What "features" did you use – appearance, speech, context?

Pause and reflect:

When you last recommended a restaurant to a friend, did you match their taste to similar friends who liked similar places?

That is KNN.

Reflection Prompt

Write down one real decision you made this week by mentally consulting similar past cases. What K did you use?

KNN mirrors how humans naturally classify: by analogy to known examples, not by learned rules

What Makes KNN Different from Logistic Regression and Trees?

Taxonomy of Classifiers

Property	KNN	Log. Reg.	Dec. Tree
Training	None	Optimize	Build tree
Boundary	Local	Linear	Axis-aligned
Interpret.	Low	High	High
New classes	Natural	Retrain	Retrain
Memory	All data	Params	Tree only
Speed	O(nd)	O(d)	O(depth)

Pattern: KNN trades training speed for prediction cost – the opposite of parametric models.

Insight

Lazy learners memorize; eager learners generalize. KNN defers all computation to prediction time.

KNN: Lazy, local, non-parametric

LogReg: Eager, global, parametric

Tree: Eager, local, non-parametric

Non-parametric means the model complexity grows with the data – no fixed number of parameters

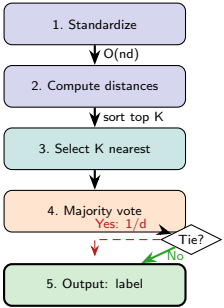
What Happens When One Applicant Enters the Feature Space?

One Prediction, Step by Step

- Applicant arrives with feature vector (income, debt ratio, employment length)
- All features standardized to zero mean, unit variance
- Algorithm computes distance to every stored example
- K closest examples "vote" on the label
- Majority wins; ties broken by distance weighting

Insight

Feature standardization is not optional – without it, high-magnitude features dominate the distance.



Complexity per query: O(nd) for brute force, O(n log n) with KD-trees when d is small

Who Should Measure Distance – Euclid, Manhattan, or Cosine?

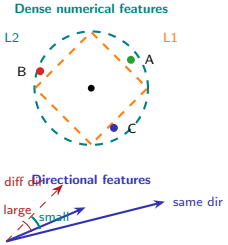
Three Distance Philosophies

- **Euclidean** (L2): straight-line, default for dense numerical data
- **Manhattan** (L1): city-block, robust to outliers
- **Cosine**: measures angle not magnitude, ideal for sparse high-D

The choice of metric reshapes which points count as neighbors.

Insight

Euclidean assumes all features equally scaled and important. When that fails, Manhattan or Cosine often outperform.



Also consider: Mahalanobis (correlated features), Hamming (binary features)

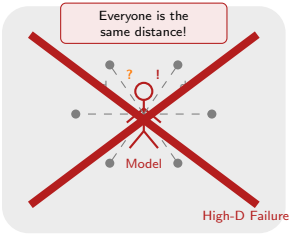
What Could Go Wrong If Every Dimension Votes Equally?

Three Ways KNN Fails Silently

- **Curse of dimensionality**: in high-D, all points become equidistant
- **Wrong K**: too small = noise, too large = local structure lost
- **Imbalanced classes**: majority dominates the vote

Insight

The curse of dimensionality is not theoretical trivia – adding irrelevant features silently degrades KNN.



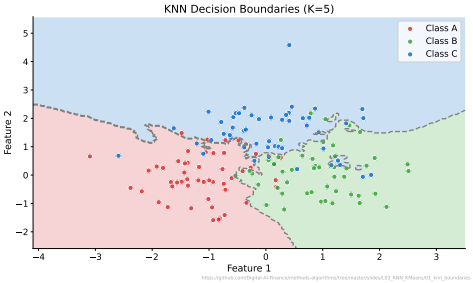
Mitigation: feature selection, PCA for dimensionality reduction, distance-weighted voting

Why Do So Many Practitioners Start with KNN?

KNN as a Baseline Classifier

- No assumptions about data distribution
- Naturally handles multi-class problems
- Decision boundaries adapt to local density
- Fast to prototype: only K to tune

The chart shows how boundaries smooth as K increases.



Insight

KNN decision boundaries emerge from the data, not from optimization. This makes KNN the natural first-look tool.

KNN with K=1 achieves asymptotic error at most twice Bayes optimal (Cover–Hart theorem)

Who Wins and Who Loses When KNN Replaces Rules?

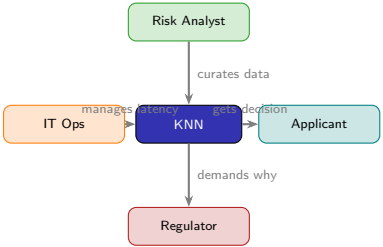
Stakeholder Analysis

- **Winners**: Risk teams, medical teams, marketing (segmentation)
- **Losers**: Compliance (hard to explain), IT ops (latency scales with data)

KNN shifts power from those who write rules to those who curate data.

Insight

KNN's Achilles heel in regulated industries is not accuracy – it is explainability.



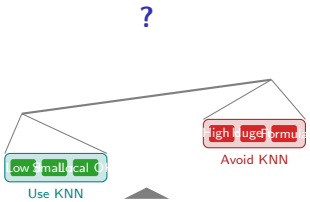
Explainability: show the K neighbors and their features as the reason for classification

3 Questions That Reveal Whether KNN Is the Right Algorithm

The Decision Framework

1. **Is the data low-dimensional?** – If $d > 20$, distances lose meaning
2. **Is the training set storable?** – If $n > 1M$, consider approximate methods
3. **Can you live without a formula?** – If regulators demand one, look elsewhere

If all three answers are yes, KNN is a strong candidate.



Insight

No algorithm is universally best. KNN excels in low dimensions, moderate data, local patterns.

The No Free Lunch theorem guarantees no single algorithm dominates across all datasets

Can You Evaluate This Real Classification Problem?

The Scenario
A retail bank wants to predict loan defaults. Features: income, age, employment length, debt-to-income, number of open accounts. The dataset has moderate size and all features are numerical.

- Apply the 3-question framework from the previous slide
- Decide: Is KNN appropriate here?
- If yes: recommend K and distance metric
- If no: name a better algorithm

Deliverable

Fill in the table. Be prepared to defend your verdict to a skeptical risk manager.

Question	Your Answer
Low-dimensional?	
Storable data?	
Example-based OK?	
Verdict	
Recommended K	
Distance metric	

Hint: consider dimensionality, data size, and regulatory environment