Exploring the Mathematics of Machine Learning Algorithms

What is Machine Learning?

Mathematically Speaking... ML is Just Curve Fitting!

- At its core, Machine Learning is about:
  1. Adjusting slopes → weights
  2. Adjusting constants → biases
- The goal?

Find a function that best fits the data.

- Futhermore, in the realm of deep learning-
  1. Slopes generalize to weights
  2. Constants generalize to biases

But then,Why Do We Need Different Algorithms?

- Some algorithms like linear regression try to learn one function that explains everything.
- But in many real-world problems:
- Data doesn’t follow a single clear pattern. Simple functions fail to generalize well.
- Some problems fall under unsupervised learning, where there’s no “output” to predict.
- Some algorithms try to split the problem and solve those chunks.
- Some are just lazy and their output is based on the surrounding datapoints.

K-Nearest Neighbors (KNN)

- Core Idea: "Birds of a Feather Flock Together"
- To classify a new data point, KNN looks at its 'k' closest neighbors in the feature space.
- The new point's class is determined by the majority class among its 'k' neighbors (for classification) or the average of their values (for regression).
- A simple, non-parametric, instance-based learning algorithm.
- "Lazy Learner": It doesn't learn a model explicitly during training; it just stores the training data. Predictions are made at query time.


neighboring point
neighboring point
neighboring point

The Mathematics of Distance

1. Euclidean Distance
   - The straight-line distance between two points in N-dimensional space

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

2. Manhattan Distance

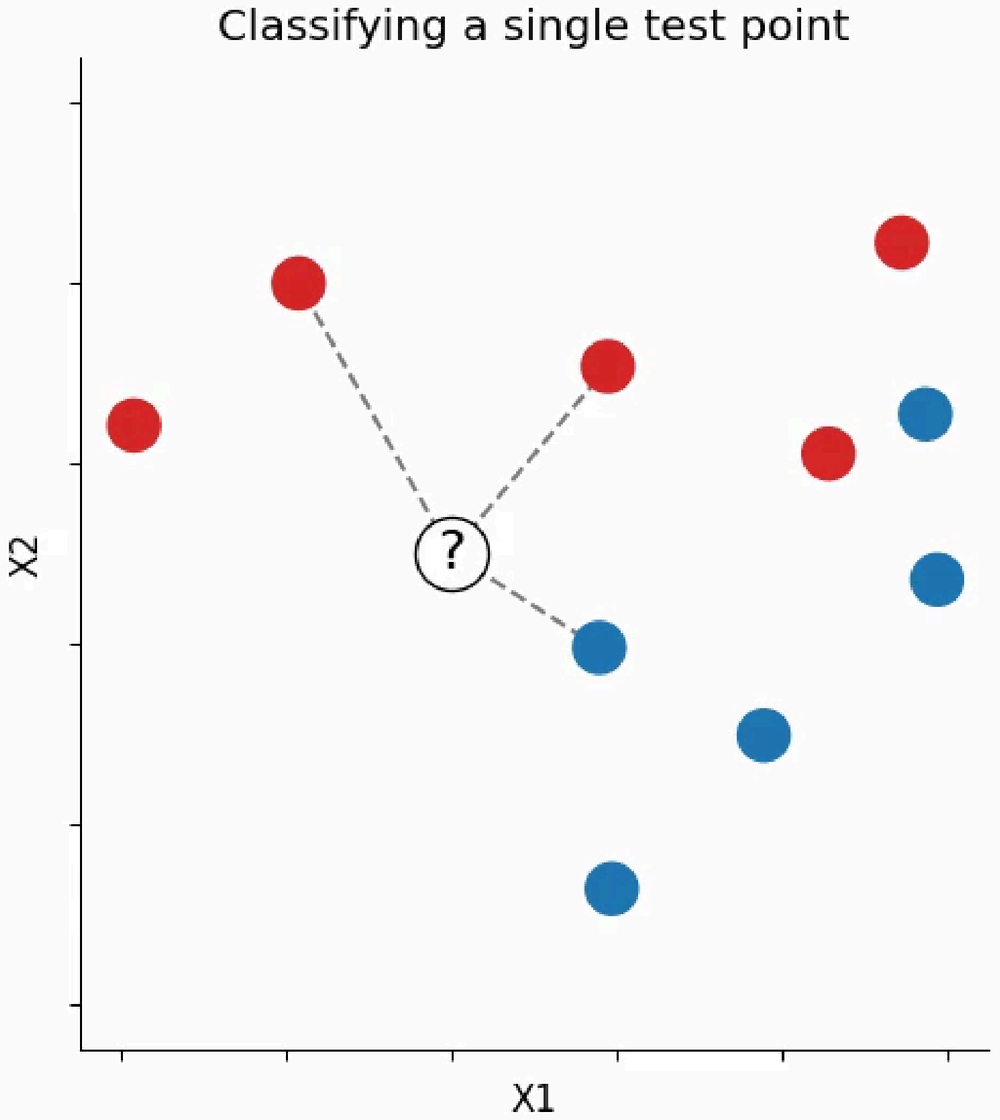
- The sum of the absolute differences of their Cartesian coordinates. Like navigating city blocks.

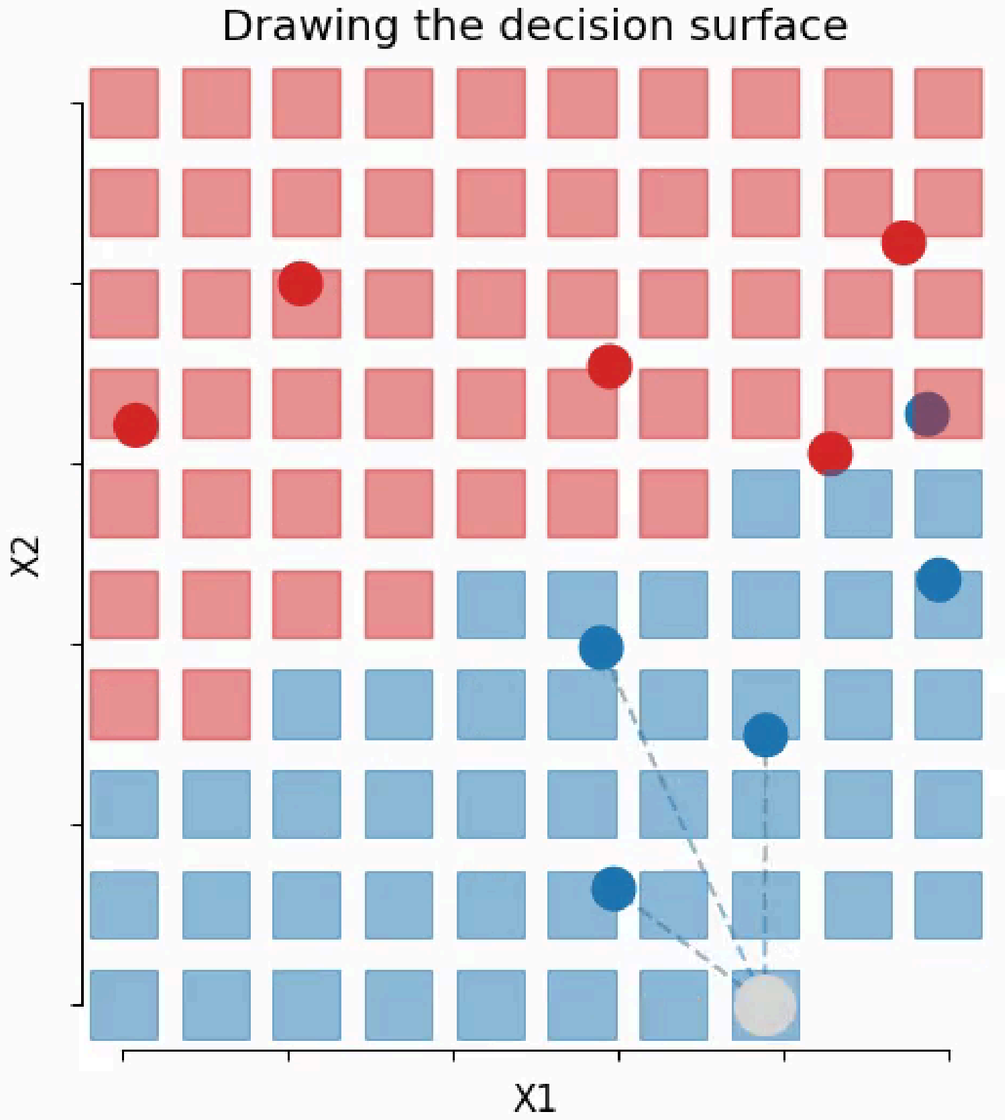
$$d(p, q) = \sum_{i=1}^{n} |q_i - p_i|$$

Example:

- Point A: (1,2)
- Point B: (4,6)
- d(A,B)-Euclidean=5
- d(A,B)-Manhattan= | 4−1 | + | 6−2 | =3+4=7

kNN classifier | $k = 3$





Clustering - Grouping Unlabeled Data

- Core Idea: "Birds of a Feather Flock Together"
- To classify a new data point, KNN looks at its 'k' closest neighbors in the feature space.
- The new point's class is determined by the majority class among its 'k' neighbors (for classification) or the average of their values (for regression).
- A simple, non-parametric, instance-based learning algorithm.
- "Lazy Learner": It doesn't learn a model explicitly during training; it just stores the training data. Predictions are made at query time.

- When You Don’t Know What’s What… So You Just Group Similar Things Together.
- The goal is to group a set of data points such that points in the same group (cluster) are more similar to each other than to those in other groups.
- Unlike classification, there are no predefined categories or target variables. The algorithm discovers inherent structures in the data, and therefore this is a technique of unsupervised learning.
- Types:
- Centroid-based Clustering (Partitioning methods)
- datasets are separated into a predetermined number of clusters.
- Density-based Clustering (Model-based methods)
- identifies clusters as areas of high density separated by regions of low density in the data space.
- Connectivity-based Clustering (Hierarchical clustering)
- builds a hierarchy of clusters using a measure of connectivity based on distance.


Corporate needs you to find the differences
between this picture and this picture.


They’re the same picture.

Step-by-Step Mathematical Explanation (K-Means Clustering)

- Choose the number of clusters (K).
- Initialize K points randomly
- Repeat until convergence (algorithm stops changing much):
- a) Assign Points to Nearest Centroid:
- For each data point Xi, find the closest centroid using

$$d(x_i, \mu_j) = \sqrt{(x_{i1} - \mu_{j1})^2 + (x_{i2} - \mu_{j2})^2 + \dots}$$

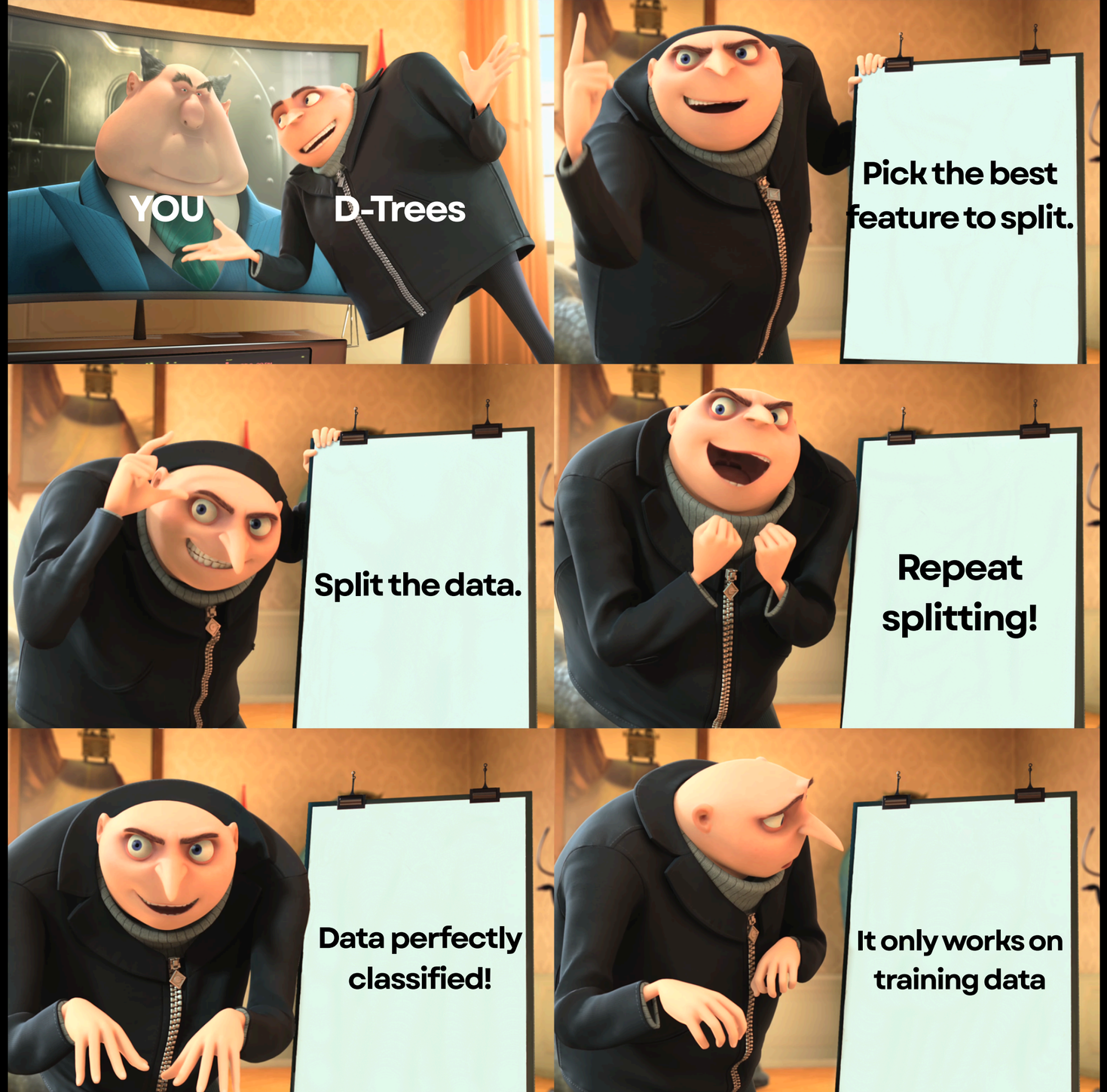
Recalculate Centroids:

After all points are assigned to a cluster, recalculate each centroid μ_j as the mean of all points assigned to that cluster:

$$\mu_j = \frac{1}{N_j} \sum_{x \in C_j} x$$

Repeat these steps until centroids stop moving much (converge).

Decision Trees


YOU
D-Trees
Pick the best feature to split.
Split the data.
Repeat splitting!
Data perfectly classified!
It only works on training data

- The goal is to group a set of data points such that points in the same group (cluster) are more similar to each other than to those in other groups.
- Unlike classification, there are no predefined categories or target variables. The algorithm discovers inherent structures in the data, and therefore this is a technique of unsupervised learning.

- Types:

1. Centroid-based Clustering (Partitioning methods)

- datasets are separated into a predetermined number of clusters.

2. Density-based Clustering (Model-based methods)

- identifies clusters as areas of high density separated by regions of low density in the data space.

3. Connectivity-based Clustering (Hierarchical clustering)

- builds a hierarchy of clusters using a measure of connectivity based on distance.

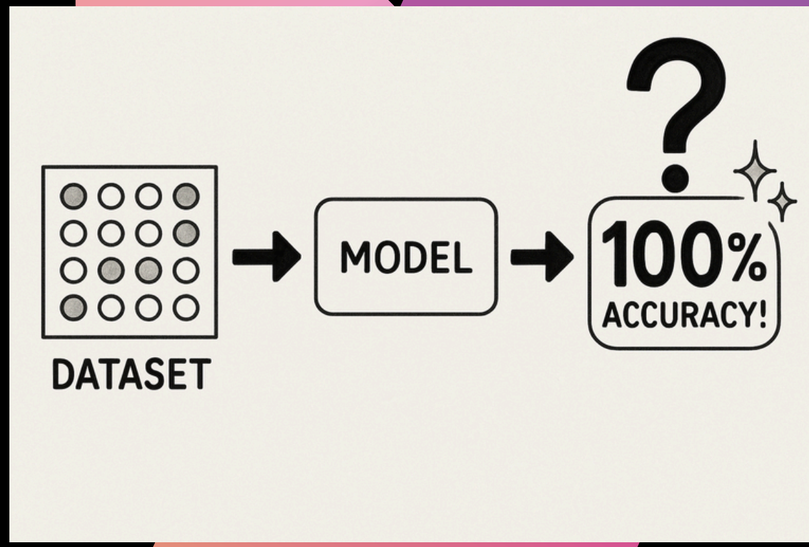
PART-2

The Trust Problem

- How do we know our model is actually good?
- Training and testing on the same data is like being your own referee—you'll always win.
- A simple Train/Test split is better, but what if you got a "lucky" or "unlucky" split of data? Your score could be misleading.
- We need a more robust, reliable testing method.

Before we learn the solution, let's solidify the problem.

- If we train our model on data, how can we get an honest grade of its performance?
- If we test it on the same data it learned from, it's like an open-book exam where the student already has the exact questions and answers.
- The score will be an inflated, overly optimistic 100%. A simple 80/20 train-test split is better, but the score can dramatically change based on pure luck—what if all the 'easy' examples ended up in the test set, or all the 'hard' ones?
- We need a system that is fair, robust, and gives us a score we can actually trust.



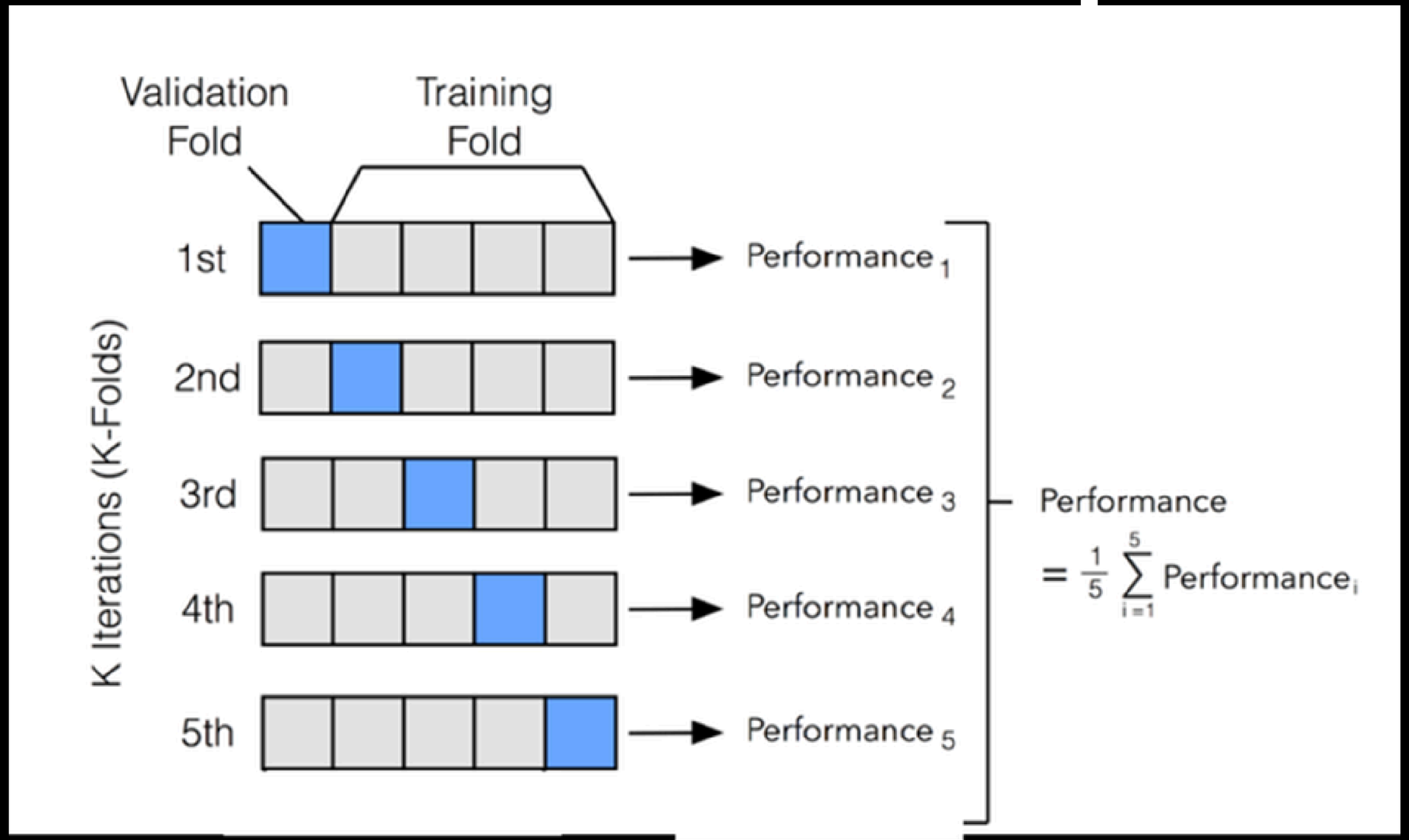
K-Fold Cross-Validation (The Setup)

- Divide and Conquer
- The Idea: Instead of one single split, we'll do multiple, systematic splits and average the results.
- This ensures every single piece of data gets to be in a test set exactly once, making our evaluation much more robust.

Step 1 : The SPLIT

- We take our entire dataset and randomly shuffle it to remove any existing order.
- We then divide this shuffled dataset into a specific number, 'k', of equal-sized groups. These groups are called "folds".
- A common and practical choice for 'k' is 5 or 10.
- When k = 5 the method is also known as 5 fold method.
- Analogy: "Think of it like slicing a cake into 'k' equal slices before anyone gets to eat. We want to make sure every slice is tested."

Divide and Conquer



Iteration 1:

- The Process:
- Isolate the Test Set: We temporarily set aside the first fold (Fold 1) to be our test set. It will not be used for training.

- Form the Training Set: We take all the other folds (Folds 2, 3, 4, and 5) and stitch them together. This combined data becomes our training set.

- Train and Test: We train our model only on the training set. Then, we evaluate its performance on the unseen test set (Fold 1) and record the score (e.g., Accuracy = 92%).

"In this first round, the model has never seen the data in Fold 1. This is a fair, unbiased test. We hold onto that score of 92% and get ready for the next round."

k-Fold Cross-Validation (The ROTATE)

- The Process: Now, we simply rotate which fold is the test set.
    - For Iteration 2:
        - Test Set: Fold 2.
        - Training Set: Folds 1, 3, 4, and 5.
        - We train a brand new model on this new training set and evaluate it on Fold 2. We record the score (e.g., Accuracy = 88%).
    - For Iteration 3:
        - Test Set: Fold 3.
        - Training Set: Folds 1, 2, 4, and 5.
        - Train a new model, test, and record the score (e.g., Accuracy = 95%).
- We repeat this process until every fold has served as the test set exactly once.

THE AVERAGE

The Process: After completing all 'k' iterations, we will have 'k' different performance scores.

- Score from Round 1 (Test on Fold 1): 92%
- Score from Round 2 (Test on Fold 2): 88%
- Score from Round 3 (Test on Fold 3): 95%
- Score from Round 4 (Test on Fold 4): 91%
- Score from Round 5 (Test on Fold 5): 94%

The Final Score: We calculate the average of these individual scores.

Final CV Accuracy = (92 + 88 + 95 + 91 + 94) / 5 = 92.0%

THE AVERAGE

The Process: After completing all 'k' iterations, we will have 'k' different performance scores.

- Score from Round 1 (Test on Fold 1): 92%
- Score from Round 2 (Test on Fold 2): 88%
- Score from Round 3 (Test on Fold 3): 95%
- Score from Round 4 (Test on Fold 4): 91%
- Score from Round 5 (Test on Fold 5): 94%

The Final Score: We calculate the average of these individual scores.

Final CV Accuracy = (92 + 88 + 95 + 91 + 94) / 5 = 92.0%

This final score is much more reliable. It tells us that, on average, our model is expected to perform at around 92% accuracy on new, unseen data. It's less sensitive to the "luck of the draw" of a single train-test split. We also get a sense of the variance (the scores ranged from 88% to 95%), which tells us how stable the model's performance is.

Leave-One-Out Cross-Validation (LOOCV)

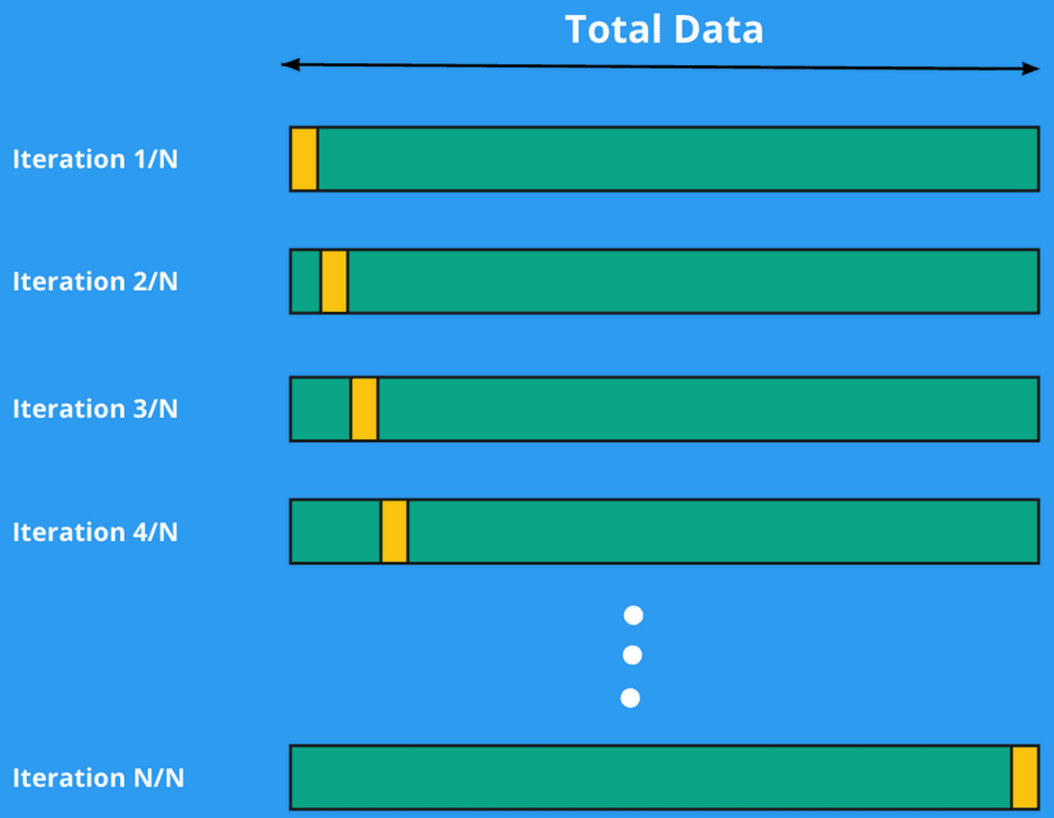
- This is the most extreme and thorough version of k-Fold Cross-Validation. It happens when you set 'k' equal to 'N', where N is the total number of data points in your dataset.

The Process in a Nutshell:

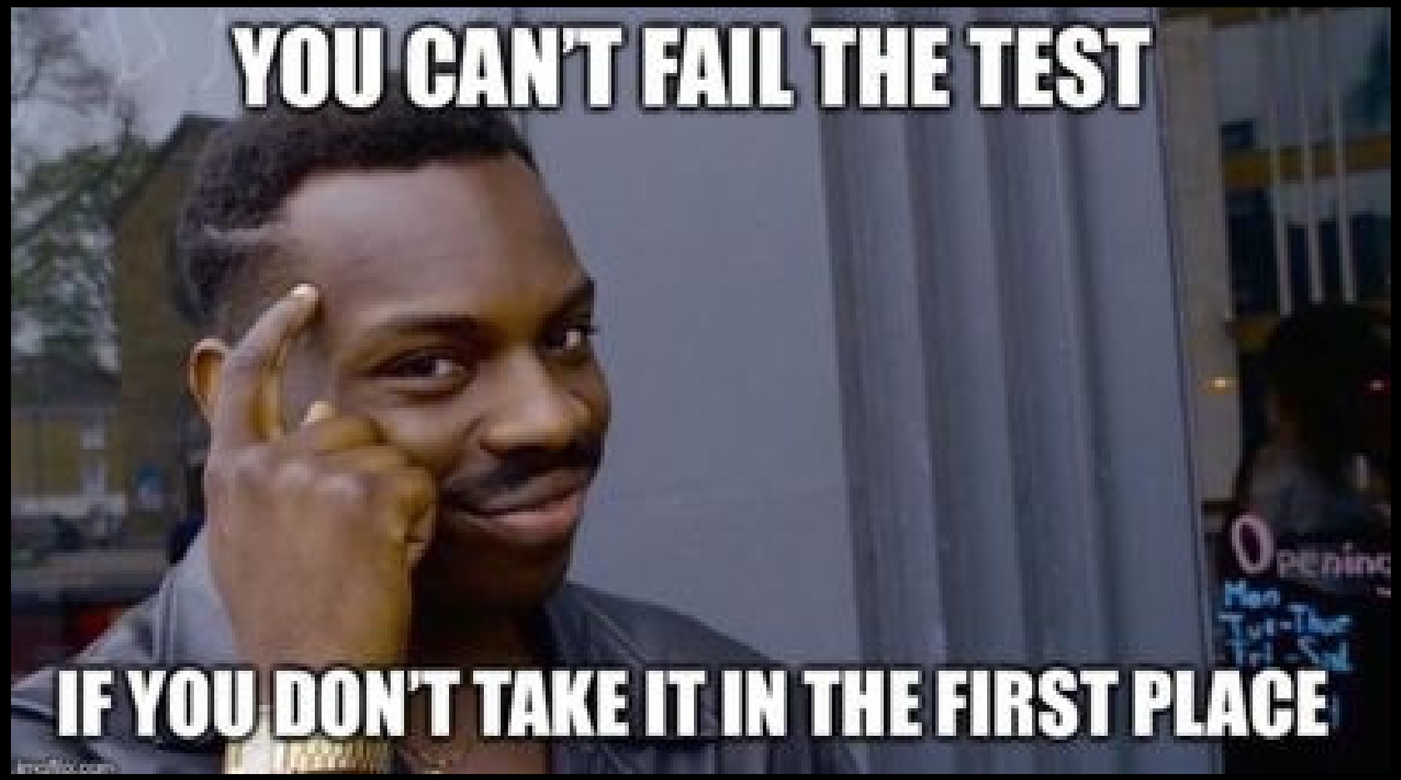
- You take one single data point out of your dataset to be the test set.
- You train the model on all other N-1 data points.
- You test the model on that single point you left out and see if the prediction is correct.
- You repeat this process N times, with each data point getting a turn to be the lone test set.

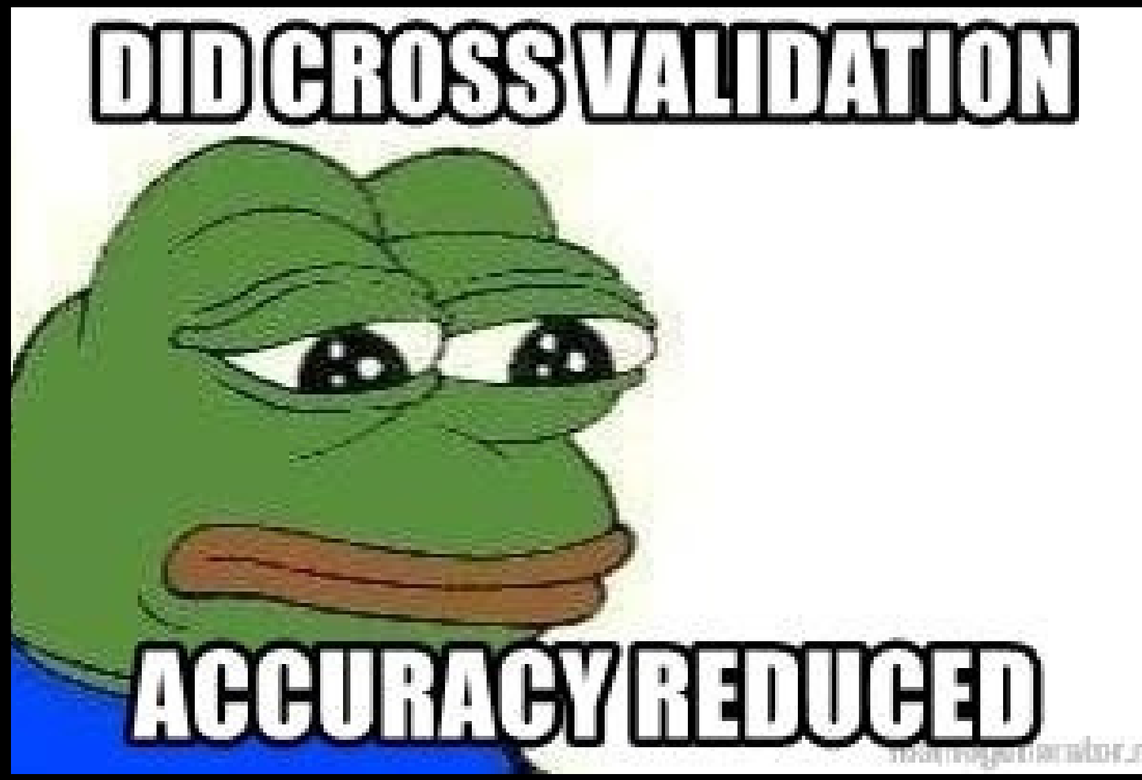
Analogy:
Imagine you have a class of 30 students. To perform LOOCV, you would have one student leave the room. You would then 'train' the remaining 29 students. Finally, you would test the student who left. You repeat this process until every single student has had a turn being tested.

LOOCV: Leave One Out Cross Validation



dataaspirant.com


YOU CAN'T FAIL THE TEST
IF YOU DON'T TAKE IT IN THE FIRST PLACE


DID CROSS VALIDATION
ACCURACY REDUCED

QUIZ TIME!

Thank You