# Prediction

Community health workers in a school, Rabai county, Kenya, September 2021

**WASHA, Takwimu, UKZN, 30 August 2023**

**Till Bärnighausen**, Heidelberg Institute of Global Health, University Hospital and Medical Faculty, Heidelberg University

# Quantitative analyses serve many important functions in health systems research

## 4 FUNCTIONS

1. Description
2. Discovery – unsupervised machine learning
   - Dimension reduction: PCA
   - Cluster analysis: k means
3. **Prediction – supervised machine learning**
   - **Penalized regression**
   - **kNN**
4. Causation
   - IV
   - RDD
   - FE

HEIDELBERG INSTITUTE OF GLOBAL HEALTH

# Causal and predictive analysis require very different approaches

## COMPARISON

| | Causation | Prediction |
|---|---|---|
| **Disciplines** | • Epidemiology<br>• Economics | • Machine learning<br>• Computer science |
| **Foundation** | • Theory-based<br>• Hypothesis testing | • Atheoretical<br>• Data-driven insight |
| **Purposes** | • Understanding<br>• Policy guidance<br>• Regulatory approval | • Intervention targeting<br>• Intervention tailoring<br>• Now- and forecasting |
| **Goal of approach** | • Minimize bias | • Optimize bias-variance trade-off |
| **Approach** | • Estimation | • Training-(validation)-testing<br>• Complexity reduction |
| **Example** | • Ordinary multiple regression<br>• PSM<br>• IV and RDD | • Regularized multiple regression<br>• kNN<br>• Neural networks |

PSM = propensity score matching, IV = instrumental variable analysis, RDD = regression discontinuity design, kNN = k nearest neighbors

# What is the generic objective in prediction?

## MATHEMATICAL INTUITION

- **Data:** $(Y_i, x_{i1}, \ldots, x_{ip})$ for $i = 1, \ldots, n$
- **Objective:** predict $Y$ for a given **new input** $x_{new} = (x_1, \ldots, x_p)$
- Two major **categories** are (machine learning language)
  - **Regression:** continuous data
  - **Classification:** discrete values representing clases

| | **Cluster membership** (reference: cluster 1) | | | |
|---|---|---|---|---|
| | *Cluster 2* | *Cluster 3* | *Cluster 4* | *Cluster 5* |
| **Age** (years) | 1.055*** | 1.038*** | 1.165*** | 1.084*** |
| | (0.0004) | (0.005) | (0.001) | (0.0004) |
| **Education** (years) | 0.986*** | 1.063*** | 1.007*** | 1.031*** |
| | (0.001) | (0.010) | (0.001) | (0.001) |
| **Wealth index** (standard deviation) | 0.910*** | 1.412*** | 1.443*** | 1.628*** |
| | (0.004) | (0.051) | (0.006) | (0.005) |
| **Rural** *Residence (reference: urban)* | 1.000*** | 0.951*** | 0.990*** | 1.116*** |
| | (0.009) | (0.091) | (0.012) | (0.009) |
| **Religion** *(reference: Hindu)* | | | | |
| Muslim | 1.124*** | 1.373*** | 1.558*** | 1.522*** |
| | (0.011) | (0.116) | (0.015) | (0.011) |
| Christian | 1.289*** | 0.986*** | 1.570*** | 1.290*** |
| | (0.013) | (0.165) | (0.018) | (0.014) |
| Sikh | 0.919*** | 1.170*** | 2.189*** | 1.768*** |
| | (0.030) | (0.252) | (0.030) | (0.025) |
| Buddhist | 1.260*** | 0.915** | 1.810*** | 1.228*** |
| | (0.033) | (0.410) | (0.040) | (0.035) |
| Jain | 1.290*** | 0.018*** | 1.199*** | 1.013*** |
| | (0.112) | (0.00005) | (0.128) | (0.105) |
| Jewish | 0.003*** | 0.526*** | 0.029*** | 1.334*** |
| | (0.00000) | (0.00001) | (0.00000) | (0.00001) |
| Zoroastrian | 4.175*** | 0.580*** | 3.354*** | 3.313*** |
| | (0.257) | (0.0002) | (0.278) | (0.282) |
| No religion | 2.492*** | 0.112*** | 5.235*** | 2.173*** |
| | (0.168) | (0.0002) | (0.193) | (0.186) |
| Other religions | 1.860*** | 0.190*** | 2.425*** | 1.505*** |
| | (0.031) | (0.001) | (0.041) | (0.036) |
| Constant | 0.244*** | 0.001 | 0.003 | 0.056*** |
| | (0.018) | (0.196) | (0.027) | (0.019) |

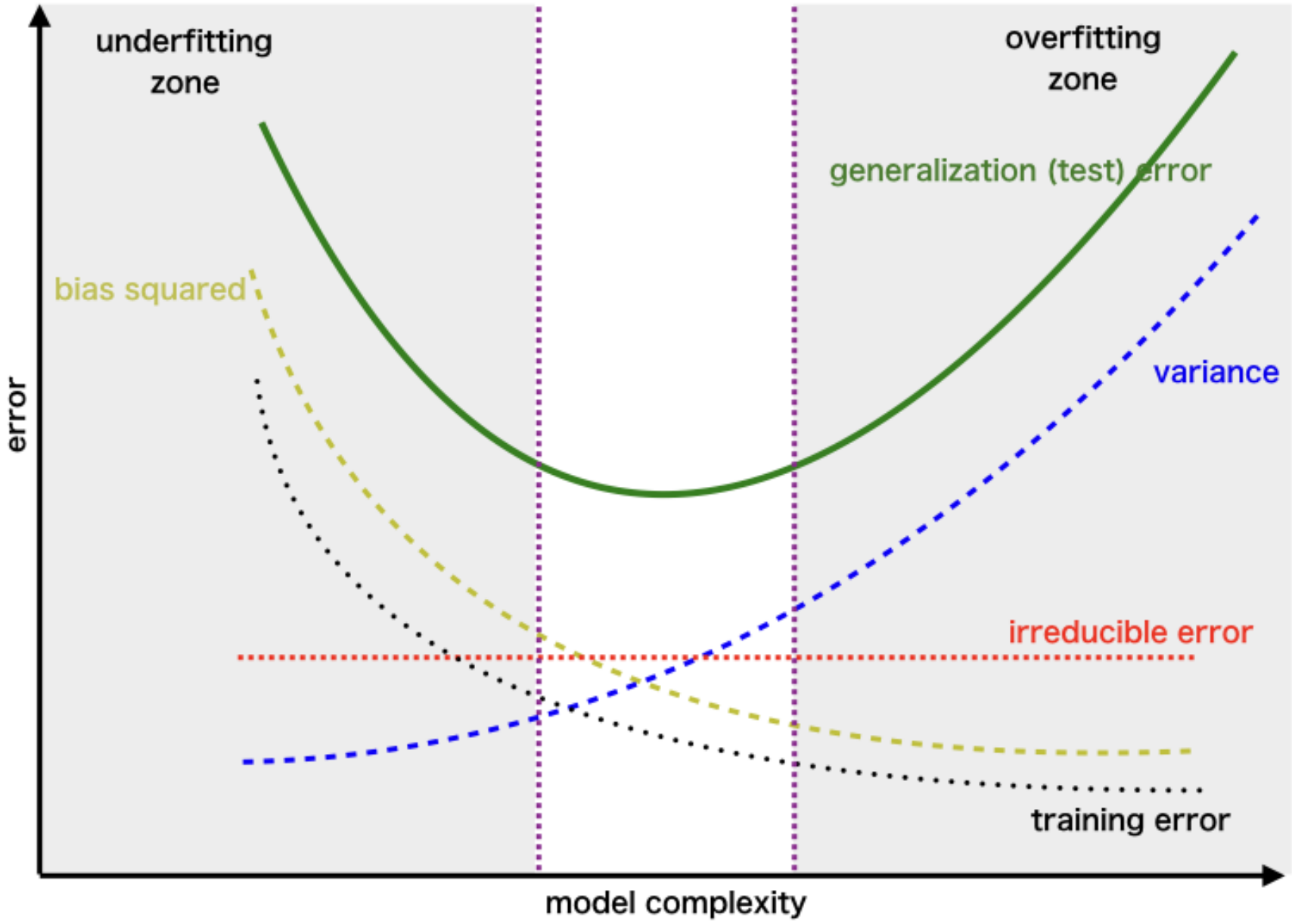
# Penalized regression takes us away from the unbiased coefficient estimators – to achieve higher predictive accuracy

## MATHEMATICAL INTUITION

- **OLS** $\min RSS$
- **LASSO** $\min(RSS + \delta_1 \sum_{j=1}^{p} |\beta_j|)$
- **Ridge** $\min(RSS + \delta_2 \sum_{j=1}^{p} \beta_j^2)$
- **Elastic net** $\min(RSS + +\delta_1 \sum_{j=1}^{p} |\beta_j| + \delta_2 \sum_{j=1}^{p} \beta_j^2)$

# For best prediction, we trade-off bias and variance

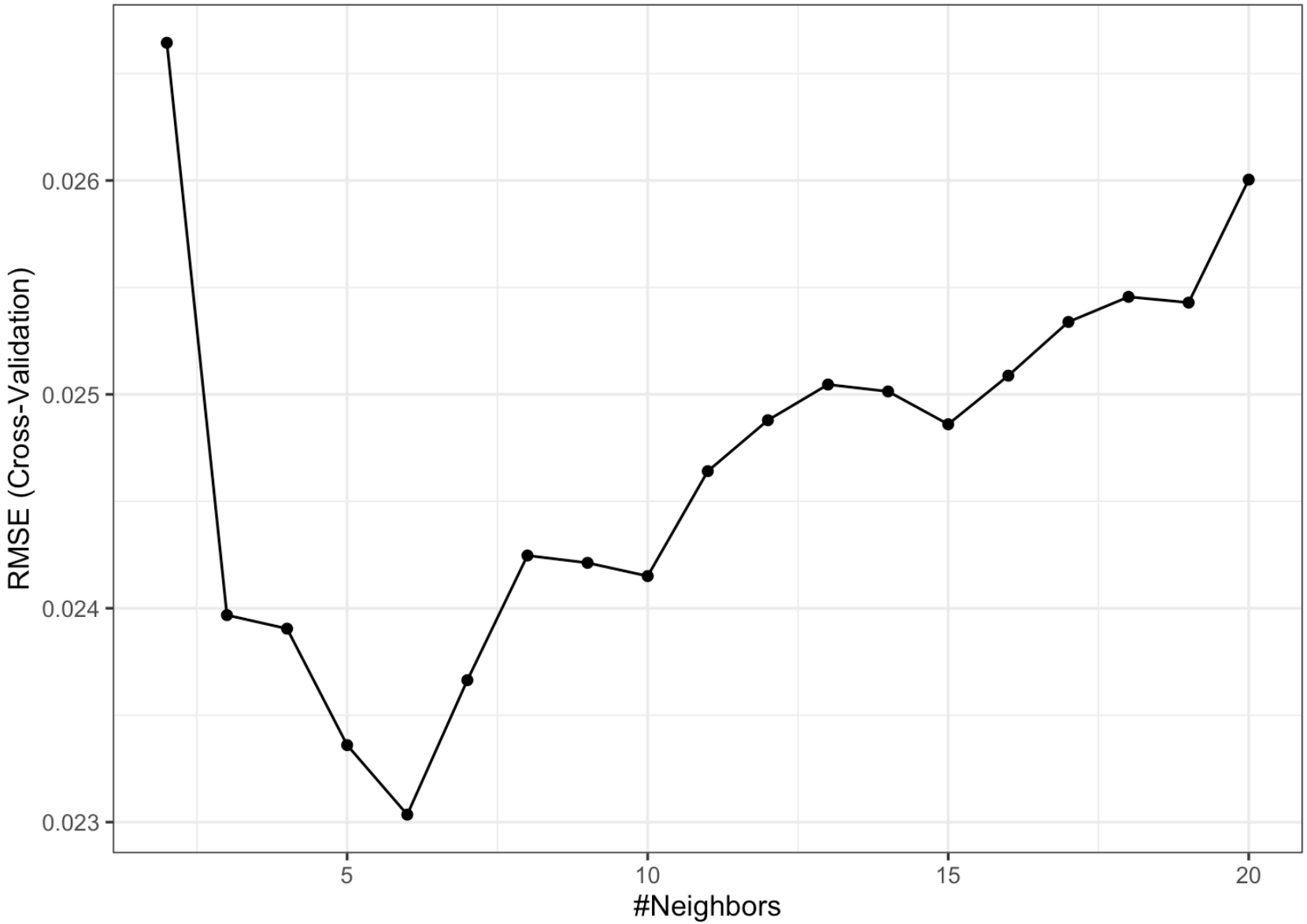
## CONCEPTS




Source: https://www.geeksforgeeks.org/ml-bias-variance-trade-off/

# We expect/hope to find a minimum in a loss function value

## EXAMPLE




RMSE = root mean squared error

# k-nearest neighbors (kNN) is a simple non-parametric supervised learning method

## OVERVIEW

- Fix & Hodges 1951
- Non-parametric
- Memoryless
- Choice of k defines locality
- As a local method: strong for low-dimensional large data
- Weak for understanding

# How do we define near?

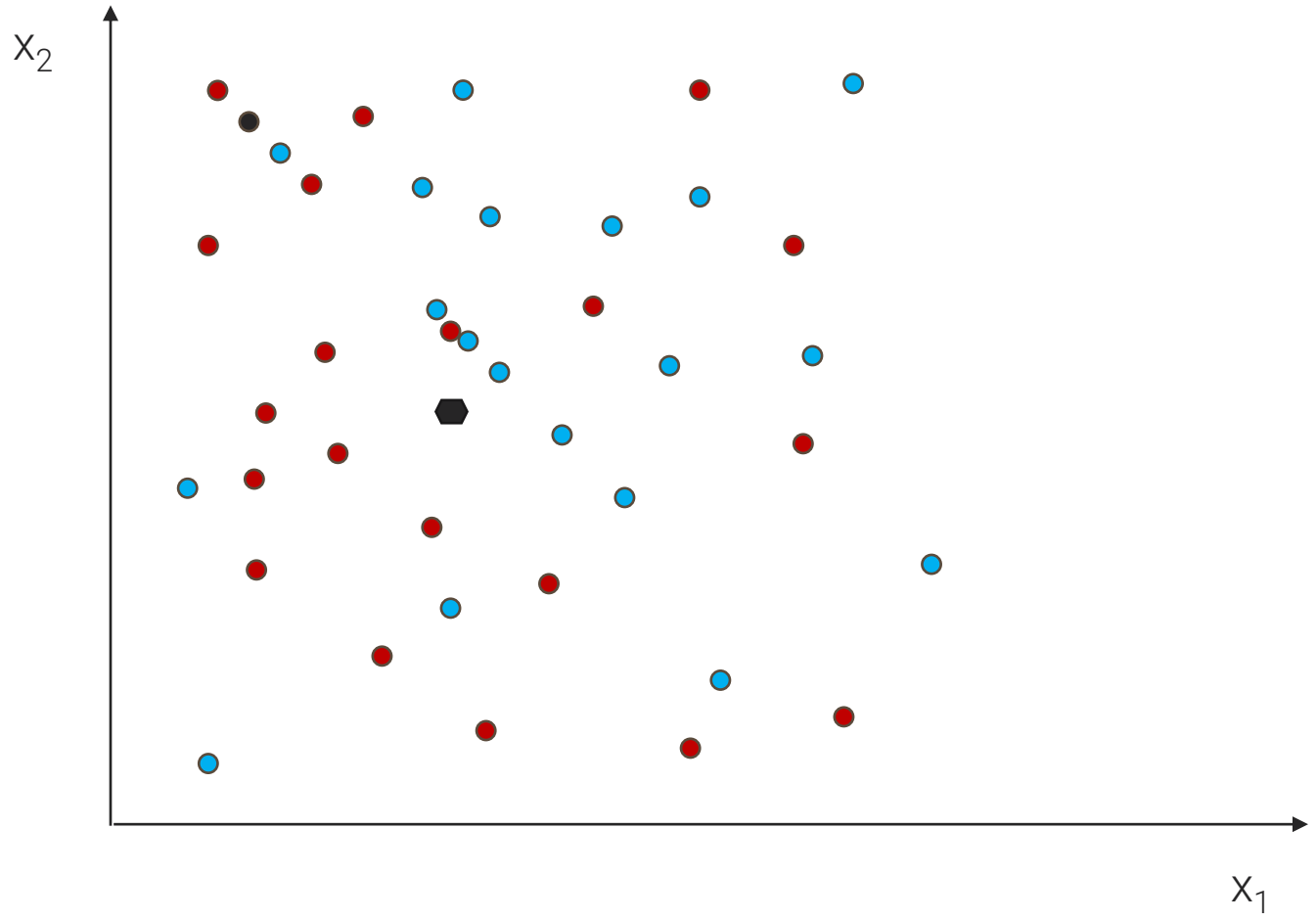
## MINKOWSKI – DISTANCE METRICS

In $\mathbb{R}^q$

$$\|\mathbf{x}' - \mathbf{x}_j\|^p = \left( \sum_{i=1}^{q} |(x_i)' - (x_i)_j|^p \right)^{1/p}$$

# For the binary case, the intuition is the majority vote

## BASIC IDEA



# kNN works for binary classification …

## FORMULA

$$\mathcal{Y} = \{1, -1\}$$

$$f_{\mathrm{KNN}}(\mathbf{x}') = \begin{cases} 1 & \text{if } \sum_{i \in \mathcal{N}_K(\mathbf{x}')} y_i \geq 0 \\ -1 & \text{if } \sum_{i \in \mathcal{N}_K(\mathbf{x}')} y_i < 0 \end{cases}$$

# … and for multiple-class classification …

## FORMULA

$$f_{\mathrm{KNN}}(\mathbf{x}') = \arg\max_{y \in \mathcal{Y}} \sum_{i \in \mathcal{N}_K(\mathbf{x}')} \mathcal{I}(y_i = y)$$

# … as well as for regression

## FORMULA

$$\mathbf{f}_{KNN}(\mathbf{x}') = \frac{1}{K} \sum_{i \in \mathcal{N}_K(\mathbf{x}')} \mathbf{y}_i$$

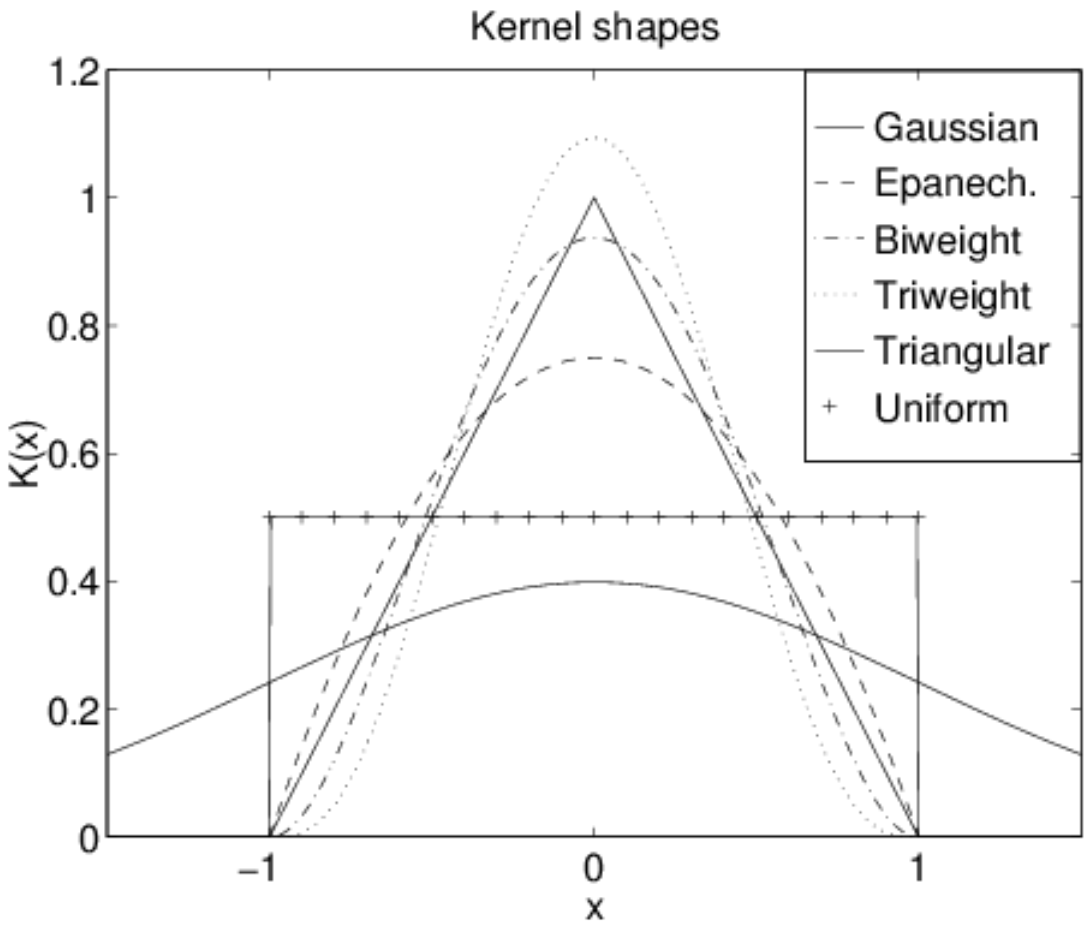
# Weighted knn may boost performance

## WEIGHTED KNN AND KERNEL TRICK

$$K(\boldsymbol{x}, \boldsymbol{y}) = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{y}) \rangle$$

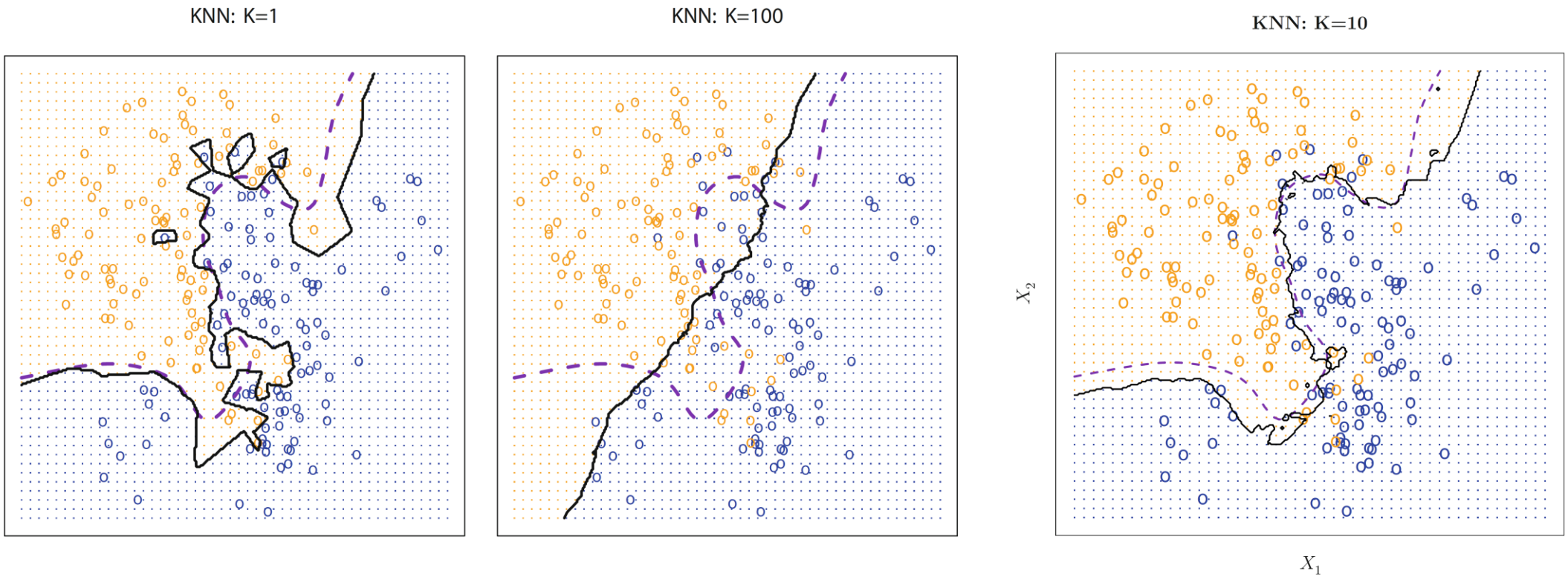
# We can use kernel functions for weighting distance

## KERNEL TYPES



# kNN needs to be ‘tuned’

## EXAMPLE




James, Witten, Hastie, Tibshirani *An Introduction to Statistical Learning* 2017

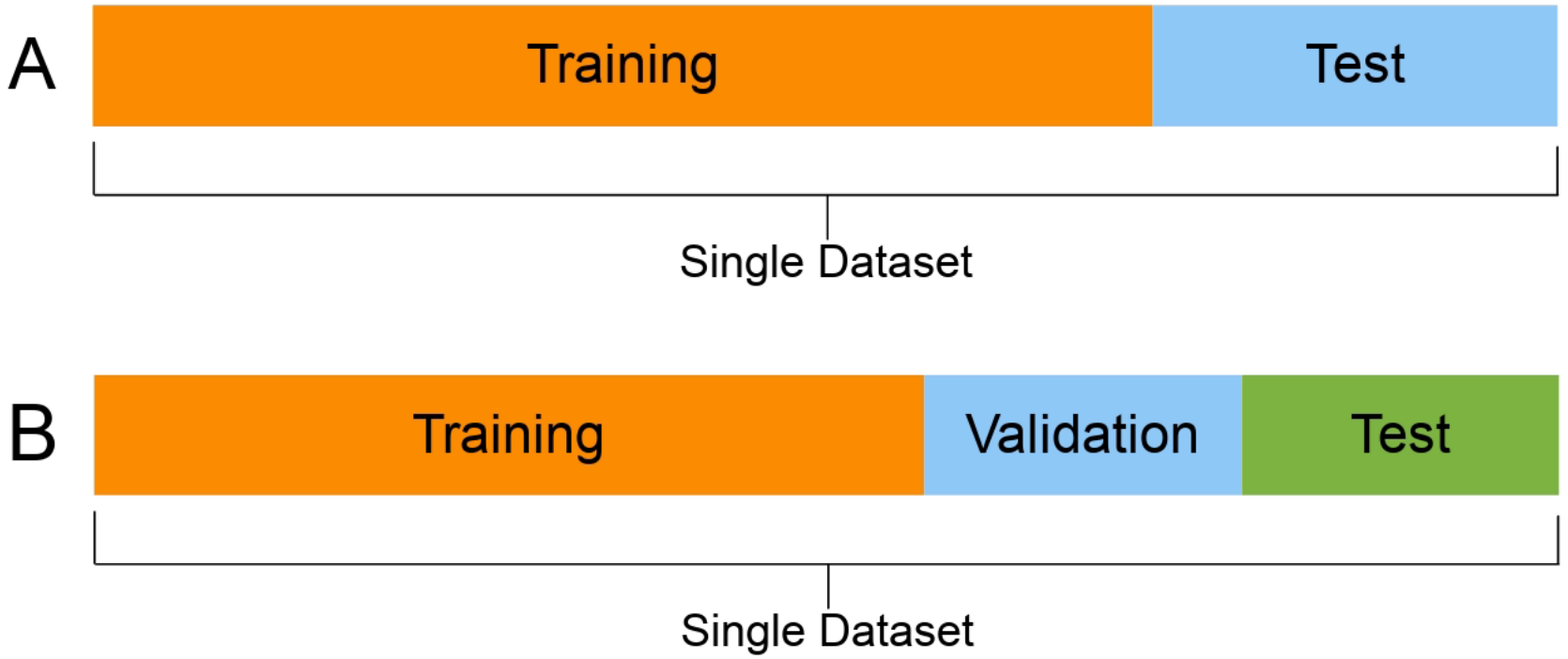
# Hyperparameters are used during the learning process – parameters are the result

## EXAMPLES

- **Hyperparmeters**
  - Train-test split ratio
  - Choice of optimization algorithm
  - Number of principal components in PCA
  - Kernel size
  - *K* in k means cluster analysis
  - *K* in knn analysis
  - Penalty term weights in penalized regression
- **Parameters**
  - Weights that generate principal components as linear combinations of the original data
  - Cluster centroids in cluster analysis
  - The actual nearest neighbors in knn
  - Coefficients in penalized regression

# Predictive practice is fundamentally different from causal practice

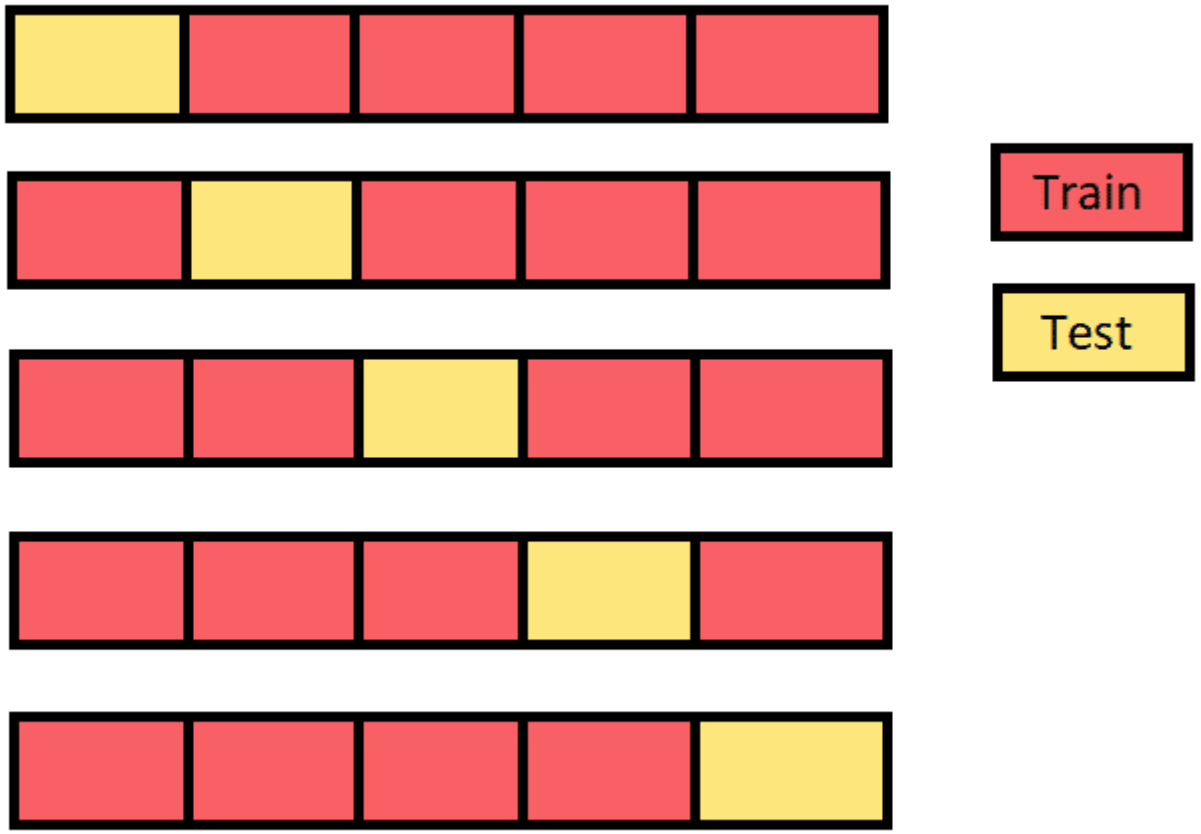
## VALIDATION DATASET FOR HYPERPARAMETER TUNING




https://commons.wikimedia.org/wiki/File:ML_dataset_training_validation_test_sets.png

# Cross-validation increases the efficiency of hyperparameter tuning

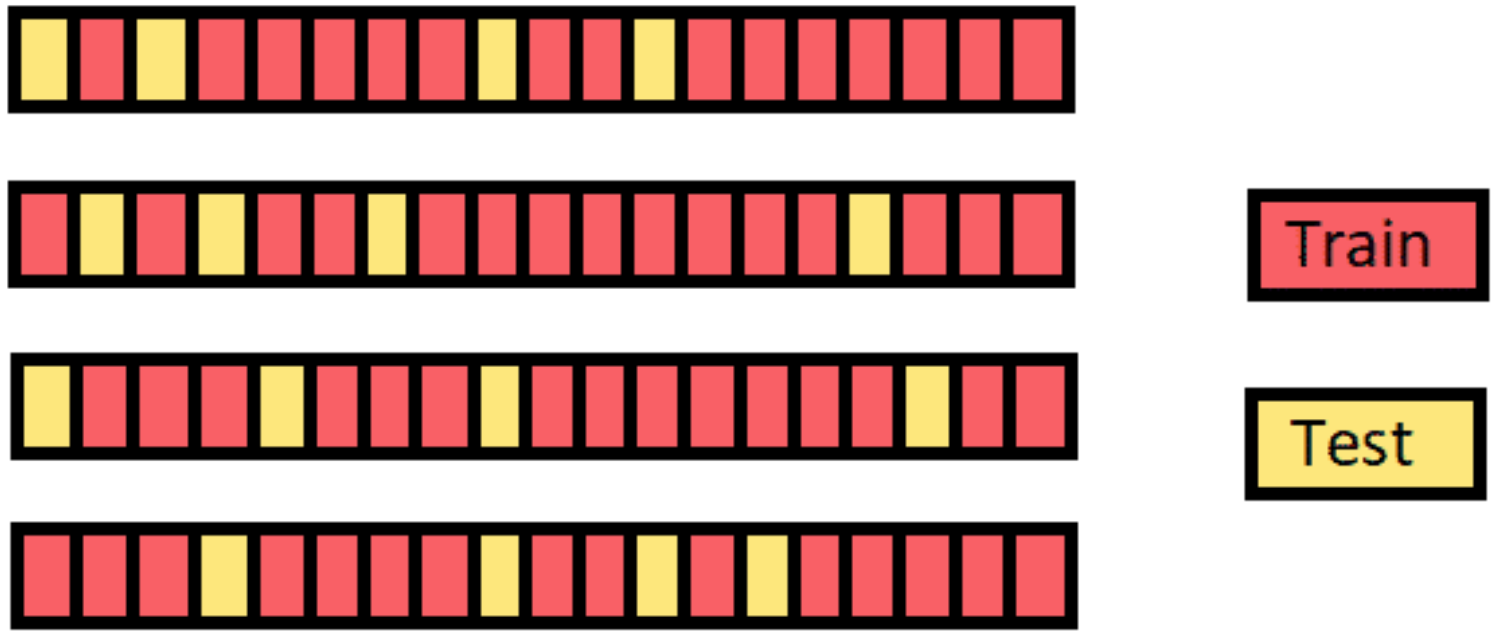
## K-FOLD




https://aiaspirant.com/cross-validation/

# Cross-validation increases the efficiency of hyperparameter tuning

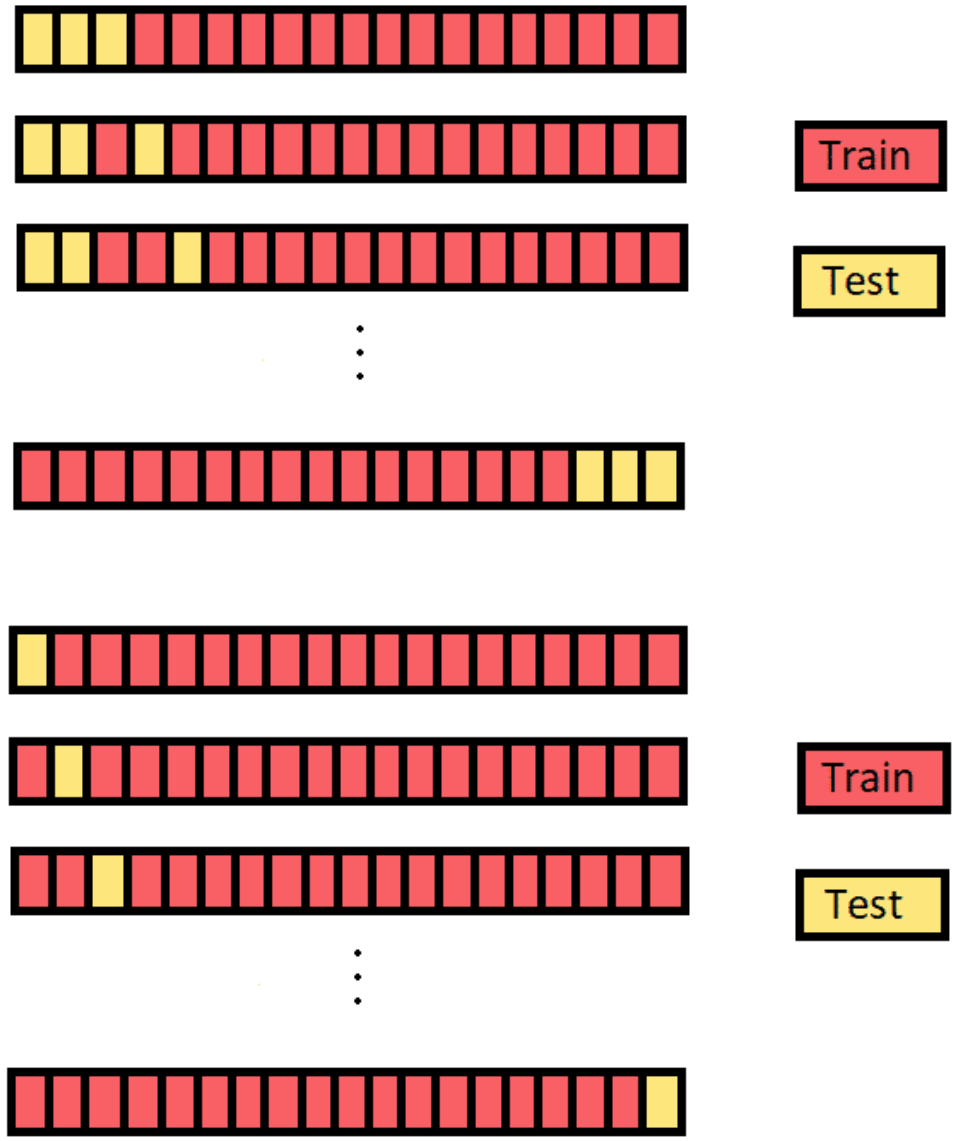
**MONTE CARLO**




https://aiaspirant.com/cross-validation/

# Cross-validation increases the efficiency of hyperparameter tuning

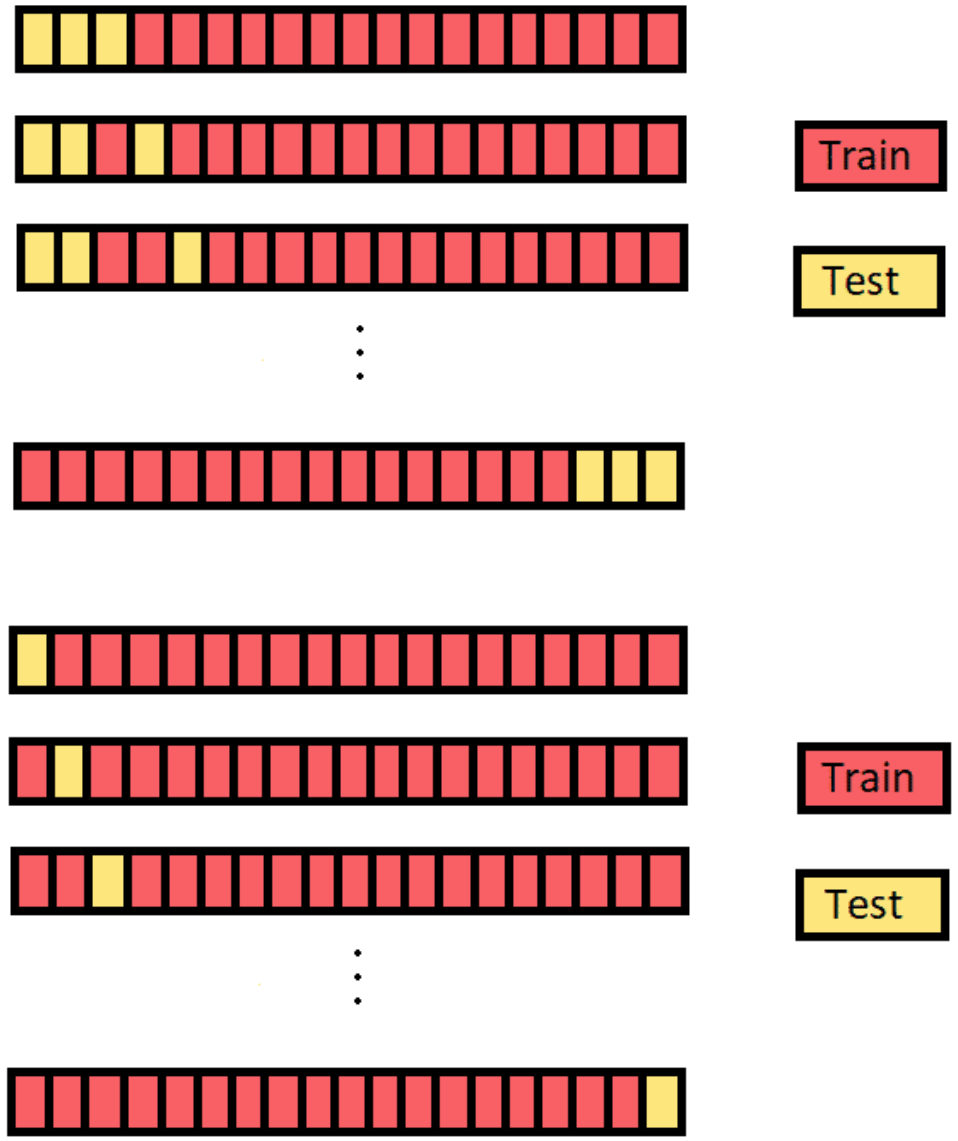
## LEAVE P OUT, LEAVE ONE OUT




https://aiaspirant.com/cross-validation/

# Cross-validation increases the efficiency of hyperparameter tuning

## LEAVE P OUT, LEAVE ONE OUT




https://aiaspirant.com/cross-validation/

# The curse of dimensionality is particularly problematic for similarity-based algorithms

**BELLMAN 1961**

- General: Sample size needed to estimate a function with a given level of accuracy grows exponentially with the dimensionality of the data

- Specifically for our topic: For similarity-based algorithms (k means, knn), the number of instances that need to be accessed for precise estimation grows exponentially with data dimensionality

# Dimension reduction is typically an important data preprocessing step for kNN

## EXAMPLES

- ***Linear*** **–**
  - Unsupervised: PCA, FA, SVM, …
  - Supervised: LDA, PLS …
- ***Non-linear*** **–** kernel PCA, FAMD, t-SNE, …

PCA = principal component analysis, SVM = support vector machine, LDA = linear discrimnant analysis, PLS = partial least squares, FAMD = factor analysis for mixed data, t-SNE = t-distributed stochastic neighbor embedding