→ K-Nearest Neighbours

- What is ~~RNN~~ KNN

KNN is a supervised, instance-based, non-parametric learning algorithm used for classification and regression.
It makes predictions by looking at the K closest data points in the training set.

- Purpose (When)

To predict the class or value of a data point based on similarity to nearby data points.

- Why it is used

- Simple and intuitive
- No training phase
- Works well with small datasets
- Effective when decision boundary is irregular

- How it works

- Choose the value of K
- Compute distance between test point and all training points
- Select the K nearest neighbors
- Aggregate their outputs ∴ Classification → majority vote
  ∴ Regression → mean/median
- Output final prediction

- Distance Metrics
  - Euclidean : $\sqrt{\Sigma (x_i - y_i)^2}$
  - Manhattan : C\sum
  - Minkowski : CC\sum
  - Cosine : $1 - \frac{x \cdot y}{\|x\| \| y\|}$

- Formula
  - Classification : $\hat{y} = mode\{y_1, y_2, \ldots, y_K\}$
  - Regression : $\hat{y} = \frac{1}{K} \sum_{i=1}^{K} y_i$

- Technical Details
  - Learning Type : • Lazy Learning (no explicit training)
    • Instance-based learning
  - Complexity : • Training time → O(1)
    • Prediction time → O(n.d)
    (n = no. of samples, d = no. of features)

- Parameters
  - Model Parameters : • training dataset itself (stored in memory)
  - Hyperparameters : • n_neighbors (K) : no. of neighbors
    • distance metric : Euclidean, Manhattan, etc.
    • weights : uniform / distance-based.
    • algorithm : brute, kd-tree, ball-tree

- Assumptions
  - Similar points have similar outputs
  - Distance metric captures similarity correctly
  - Features are properly scaled

- Pros
  - Simple to understand
  - No training required
  - Flexible decision boundaries
  - Adapts easily to multi-class problems

- Cons
  - Computationally expensive at ~~inter~~ inference
  - Sensitive to feature scaling
  - Poor performance with high-dimensional data
  - Requires large memory
  - Sensitive to noise and outliers

- Real-World Applications (where)
  - Recommendation Systems : Movie and product recommendations
  - Healthcare : Disease classification
    Medical image analysis
  - Anomaly Detection : Outlier detection
  - Finance : Credit risk assessment
  - Computer vision : Image similarity
    Face recognition

- Best Practices
  - Always scale features
  - Choose odd K for binary classification
  - Use cross-validation to find K
  - Use distance-weighted voting