K – Nearest Neighbors (KNN) algorithm

Classifying multiple labels simultaneously

No training needed. This is also called non-parametric algorithm

It is not very efficient when the data points are high-dimensional

How or where does it work?

1st step: Perception

2nd step: Extract features

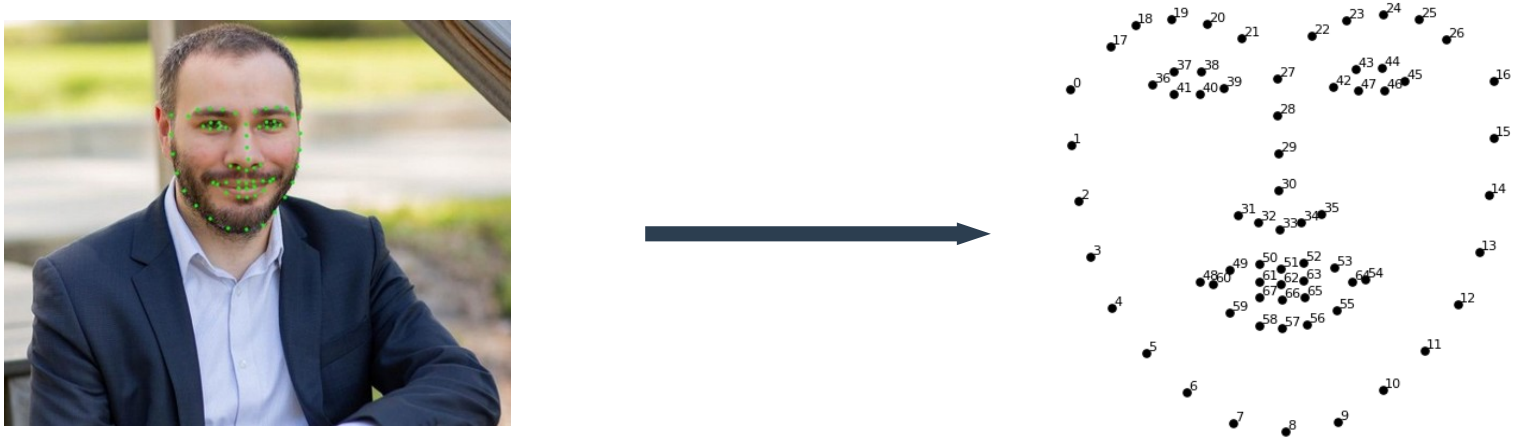
3rd step: Stack features in a vector

4th step: Compare such a vector with “data base”

The KNN algorithm focuses on the comparison

Example

Features in face recognition

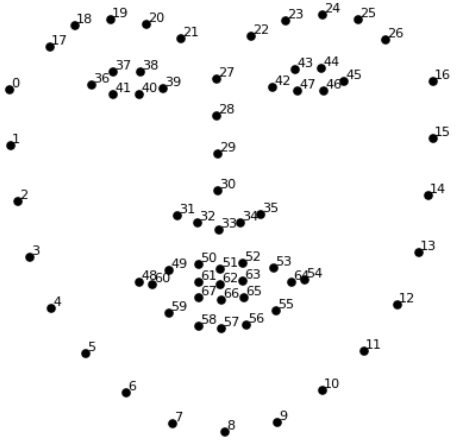


Example

Construct the vector

(typically relative points)

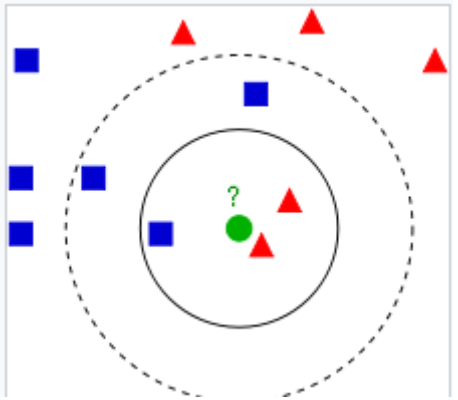
- Jaw Points = 0–16
- Right Brow Points = 17–21
- Left Brow Points = 22–26
- Nose Points = 27–35
- Right Eye Points = 36–41
- Left Eye Points = 42–47
- Mouth Points = 48–60
- Lips Points = 61–67



Example

What person is it?

Depends on the metric and the number of neighbors to consider



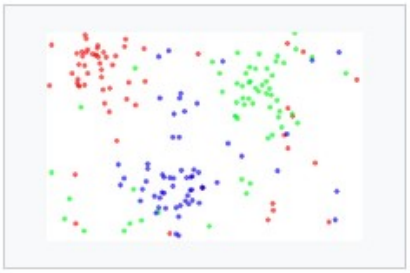


Fig. 1. The dataset.

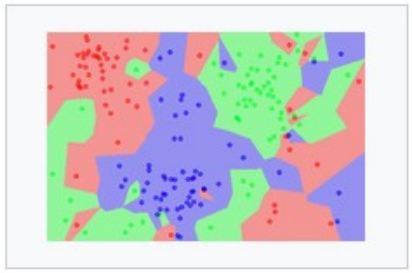


Fig. 2. The 1NN classification map.

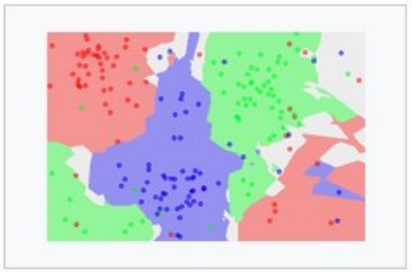


Fig. 3. The 5NN classification map.

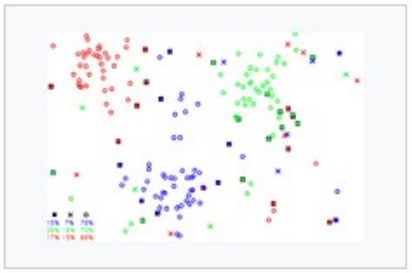


Fig. 4. The CNN reduced dataset.

Fig. 5. The 1NN classification map based on the CNN extracted prototypes.

Curse of dimensionality

Euclidean distance (d) $d^2(p,q) = (p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_i - q_i)^2 + \cdots + (p_n - q_n)^2.$

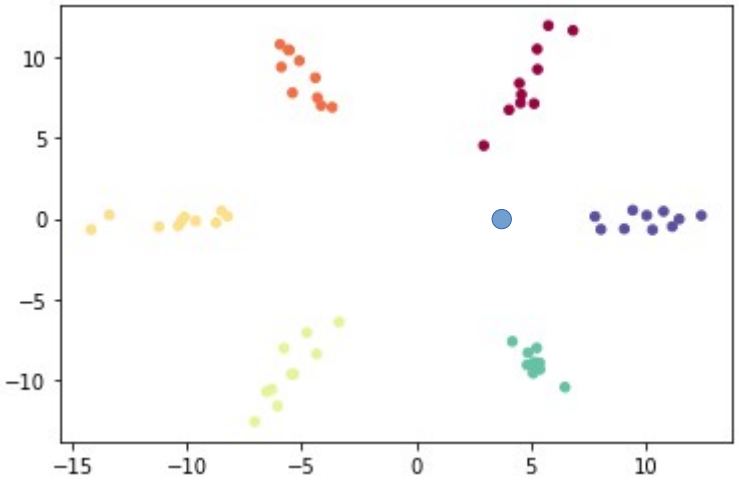
Hypersphere vs hypercube (d is dimension here)

$$\frac{V_{\text{hypersphere}}}{V_{\text{hypercube}}} = \frac{\pi^{d/2}}{d2^{d-1}\Gamma(d/2)} \to 0 \text{ as } d \to \infty.$$

Preprocessing of the data

Dimension reduction: Some features may not be the significant.

Chose your metric well!



Preprocessing of the data

Dimension reduction: Some features may not be the significant.

Chose your metric well! (Dot product as a metric to measure angles) . It works well to identify texts