

The background of the slide is a repeating pattern of various light gray icons related to technology and artificial intelligence. These icons include human heads with circuitry inside, gears, microchips, robots, computer monitors, and abstract circuit diagrams. The icons are arranged in a grid-like fashion, creating a dense, textured background.

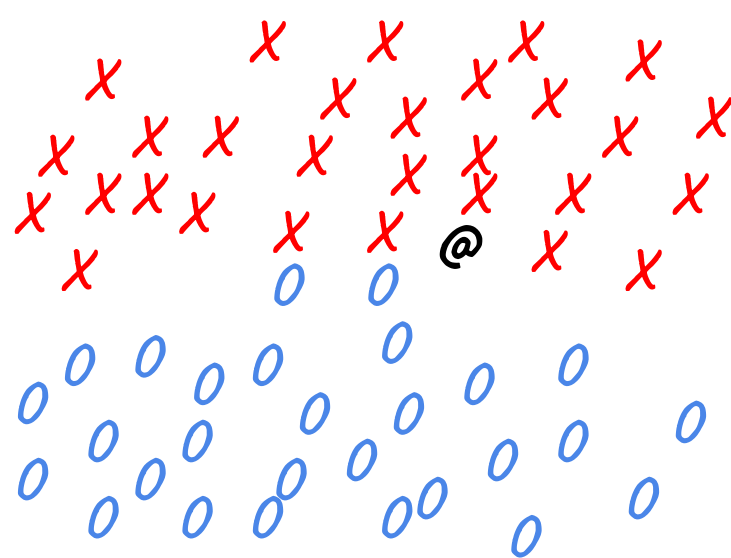
Machine Learning I

Mohamed Hussien

The background of the slide is a repeating pattern of various light gray icons related to technology and artificial intelligence. These icons include computer monitors, circuit boards, gears, human heads with circuitry inside, robots, and abstract network diagrams. A dashed rectangular box is centered on the slide, containing the title text.

K - Nearest Neighbors (KNN)

K - Nearest Neighbor [Classifier]



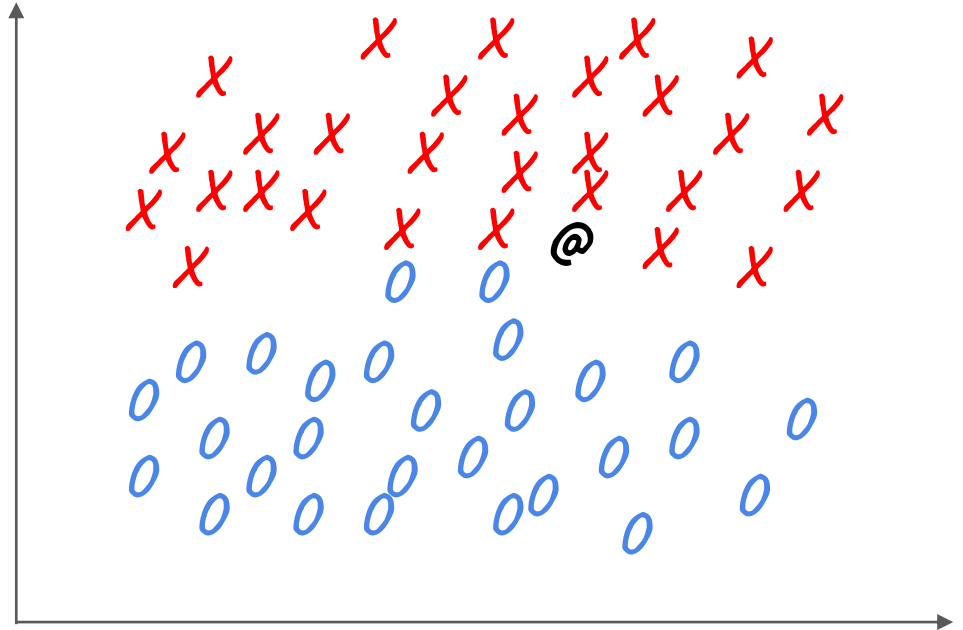
K - Nearest Neighbor [Classifier]

Problem:

We want to classify @

Steps:

- 1- Define K parameter (ex: 5)
- 2- Get the **nearest K** data-points to @
- 3- Take **votes** from these points



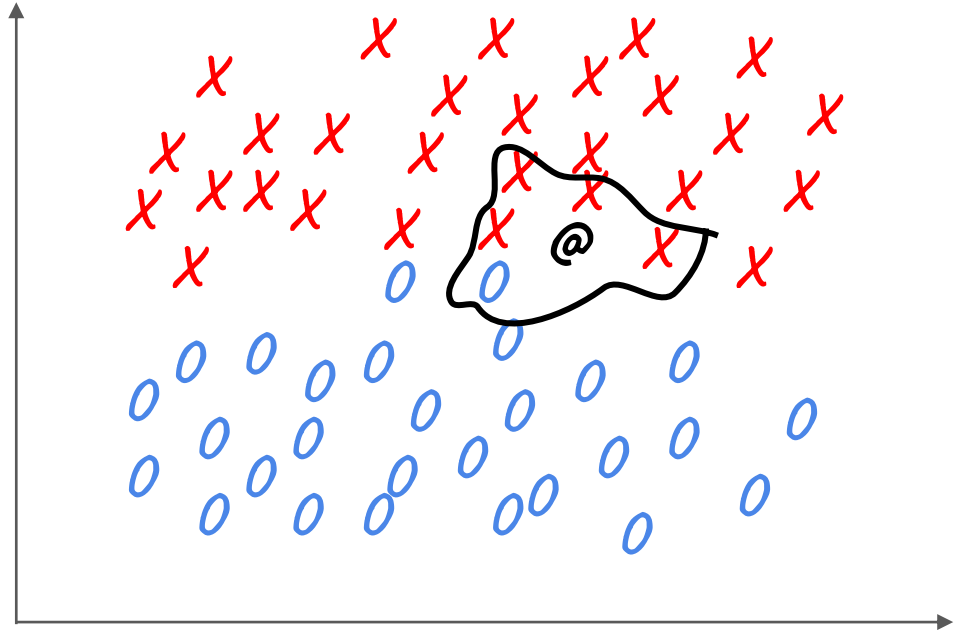
K - Nearest Neighbor [Classifier]

Problem:

We want to classify @

Steps:

- 1- Define K parameter (ex: 5)
- 2- Get the **nearest K** data-points to @
- 3- Take **votes** from these points



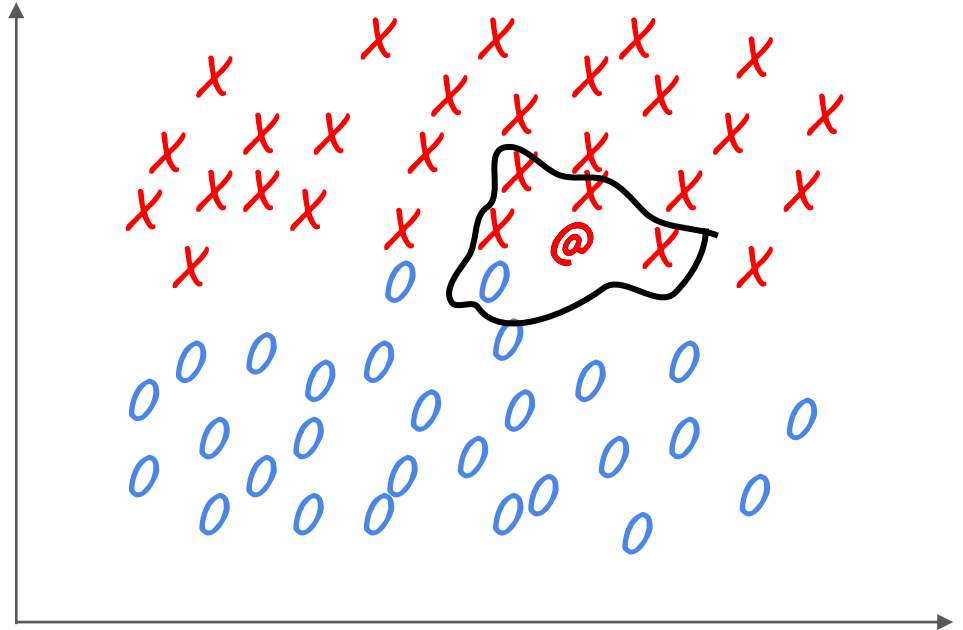
K - Nearest Neighbor [Classifier]

Problem:

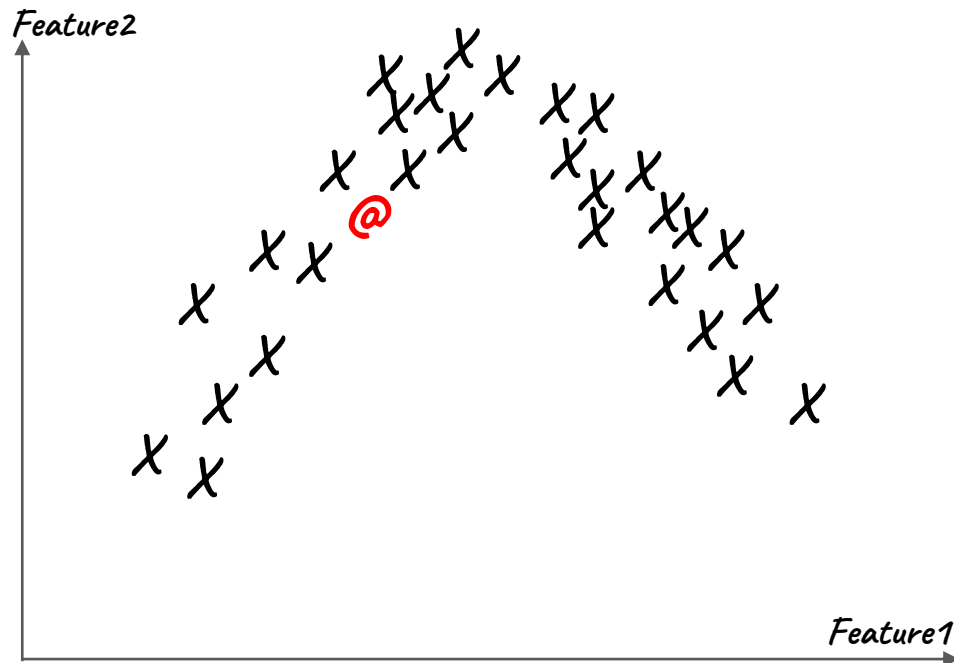
We want to classify @

Steps:

- 1- Define K parameter (ex: 5)
- 2- Get the **nearest K** data-points to @
- 3- Take **votes** from these points



K - Nearest Neighbor [Regression]



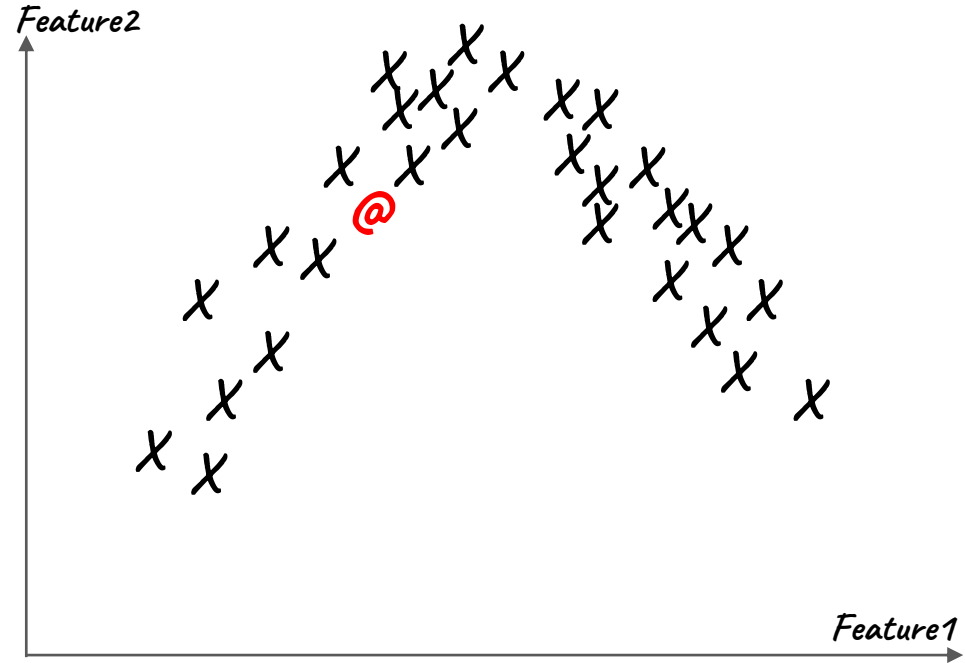
K - Nearest Neighbor [Regression]

Problem:

We want to find the prediction of @

Steps:

- 1- Define K parameter (ex: 5)
- 2- Get the **nearest K** data-points to @
- 3- Take **average** from these points labels



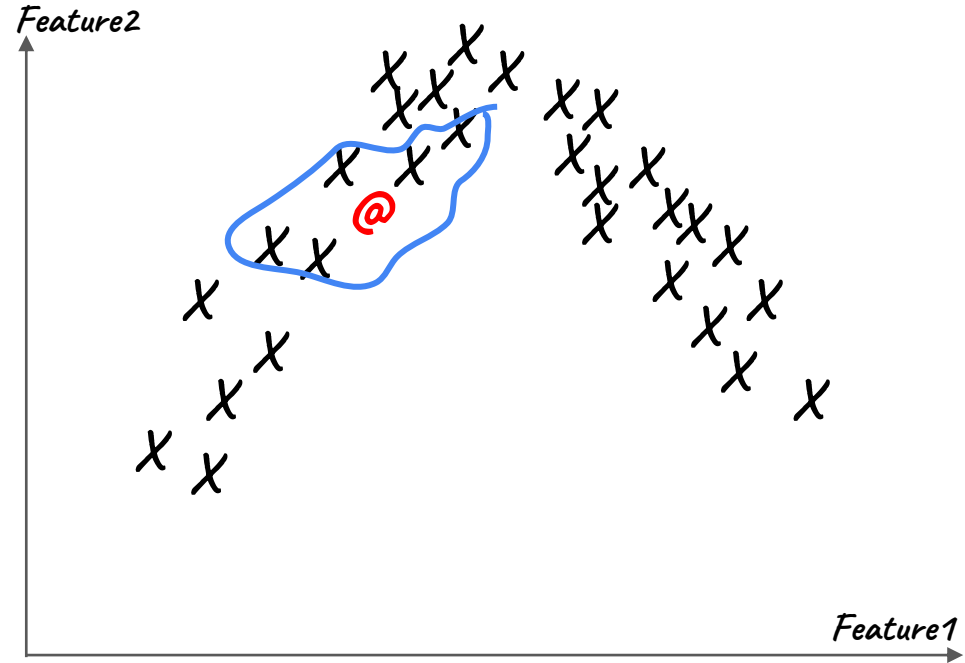
K - Nearest Neighbor [Regression]

Problem:

We want to find the prediction of @

Steps:

- 1- Define K parameter (ex: 5)
- 2- Get the **nearest K** data-points to @
- 3- Take **average** from these points labels



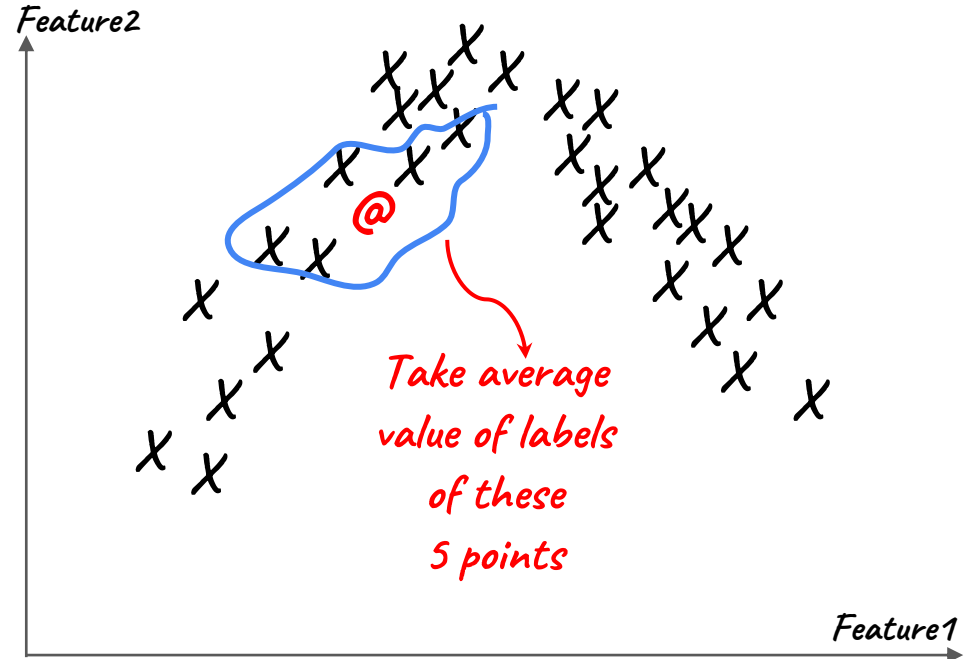
K - Nearest Neighbor [Regression]

Problem:

We want to find the prediction of @

Steps:

- 1- Define K parameter (ex: 5)
- 2- Get the **nearest K** data-points to @
- 3- Take **average** from these points labels



K - Nearest Neighbor [Regression]

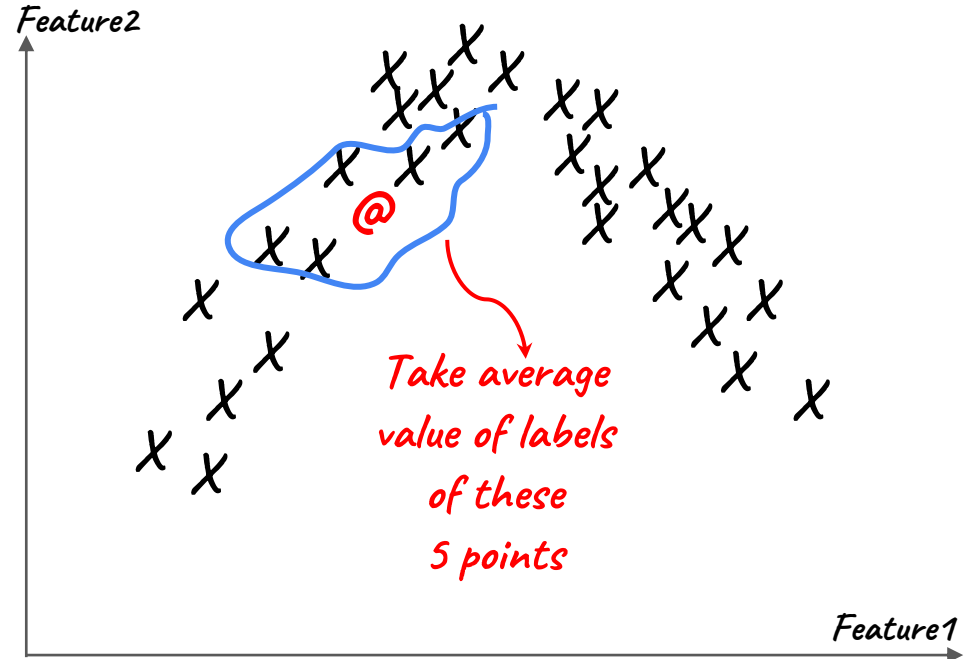
Problem:

We want to find the prediction of @

Steps:

- 1- Define K parameter (ex: 5)
- 2- Get the **nearest K** data-points to @
- 3- Take **average** from these points labels

Note: we drew **features only** without the label



K - Nearest Neighbor [Pros & Cons]



K - Nearest Neighbor [Pros & Cons]

Pros:

- Non training time.
- Easy to implement
- Non-linear performance

Cons:

- High complexity with large data or number of features
- Sensitive to noise

Sklearn



Use colab to open this github notebook:

`"s7s/machine_learning_1/KNN/KNN.ipynb"`