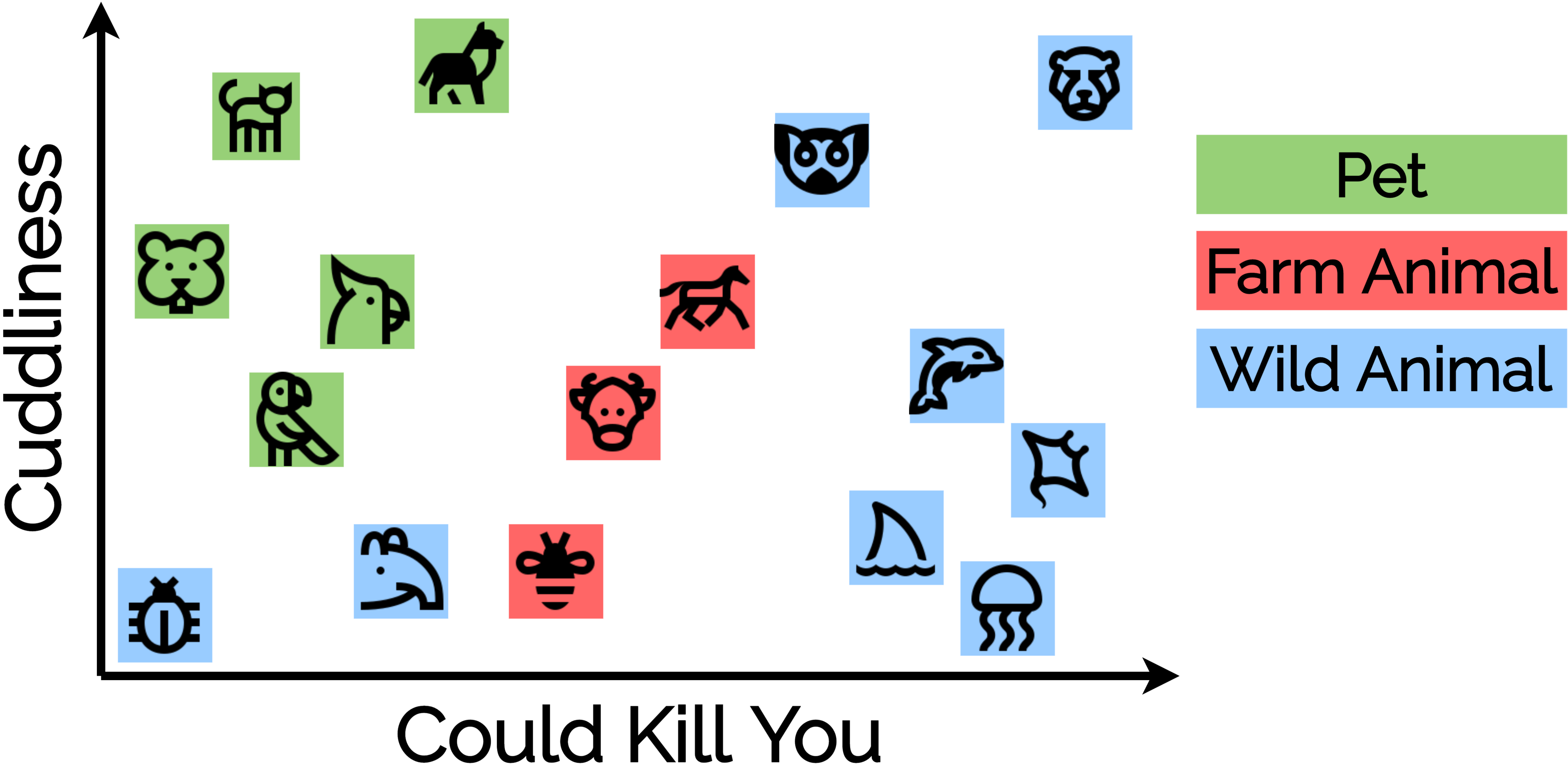
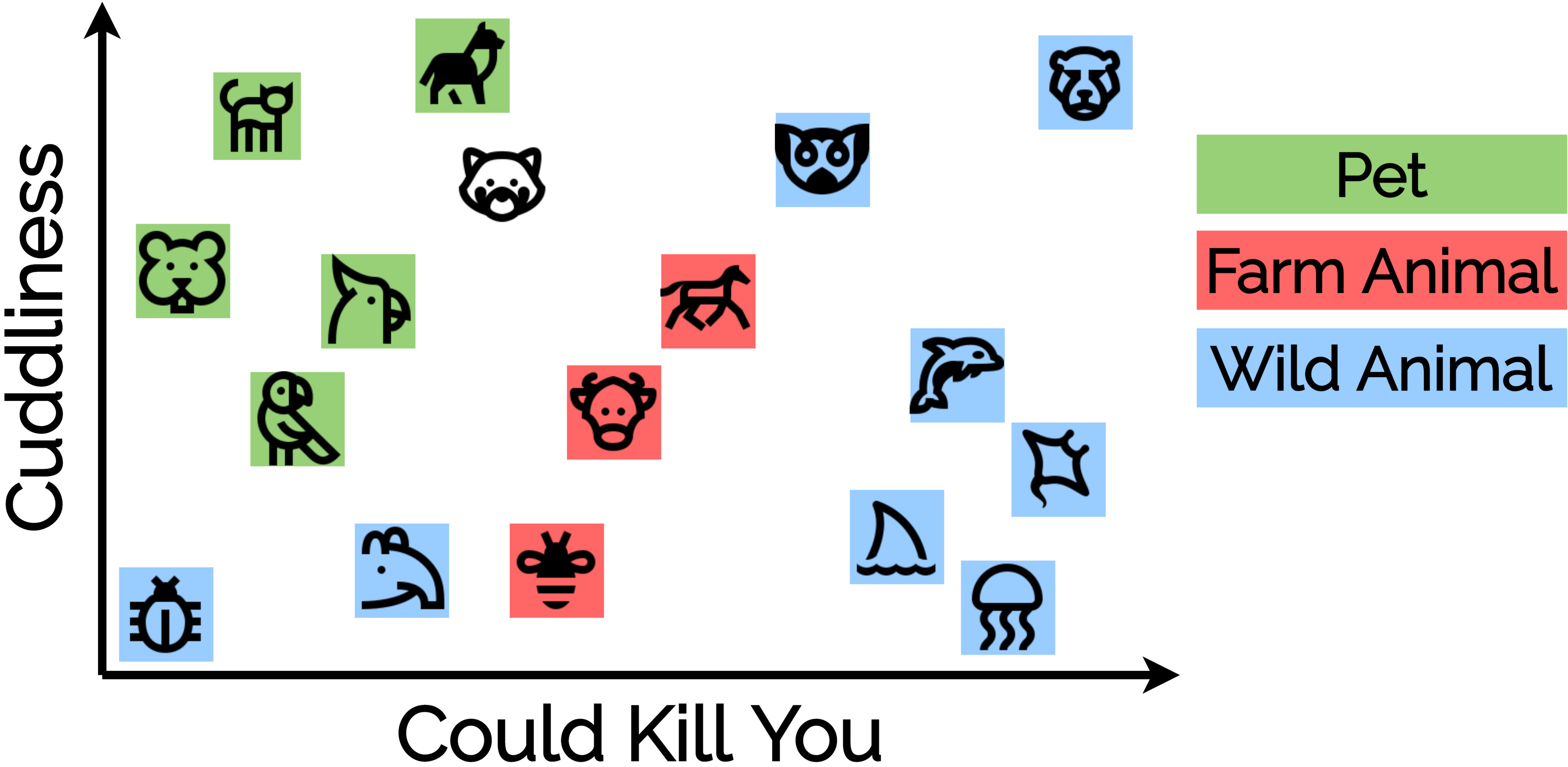
K-NEAREST NEIGHBORS

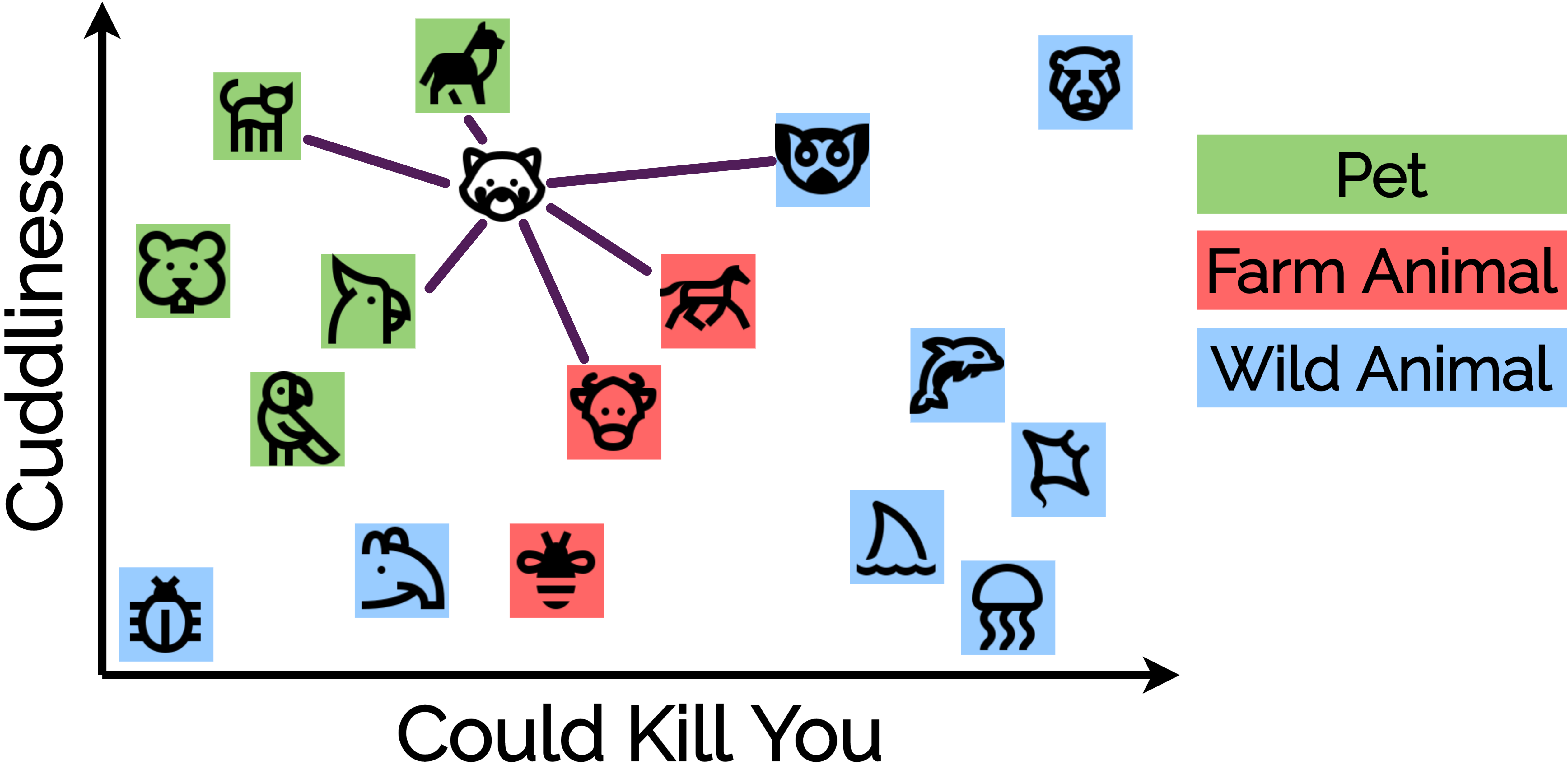
GROUP 5: KRIPANJALI DHUNGANA,
JACK LYNN, NORMAN MORRIS, &
SAM WAINRIGHT

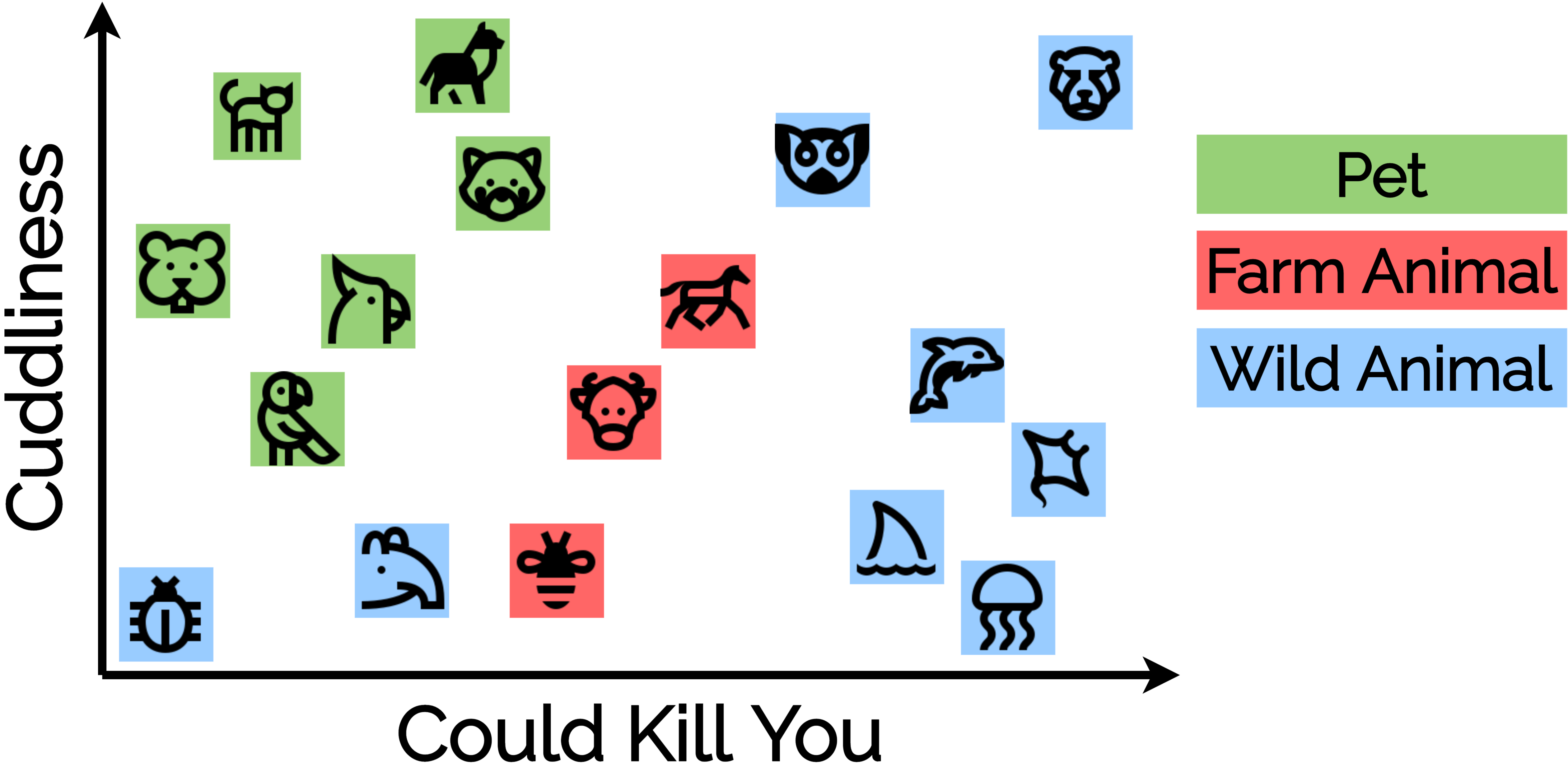
K-NEAREST NEIGHBORS (K-NN)

- **Supervised learning**
- Can be used for **regression** and **classification** algorithms
- **Instance-based learning** (lazy learning)
- Based on the assumption that **like instances** are the **closest distance** to one another
- Uses **voting** mechanism: **k-nearest instances** vote to classify an unknown instance


Cuddliness
Could Kill You
Pet
Farm Animal
Wild Animal


Cuddliness
Could Kill You
Pet
Farm Animal
Wild Animal


Cuddliness
Could Kill You
Pet
Farm Animal
Wild Animal


Cuddliness
Could Kill You
Pet
Farm Animal
Wild Animal

PREPROCESSING: INTERPOLATION AND NAN HANDLING

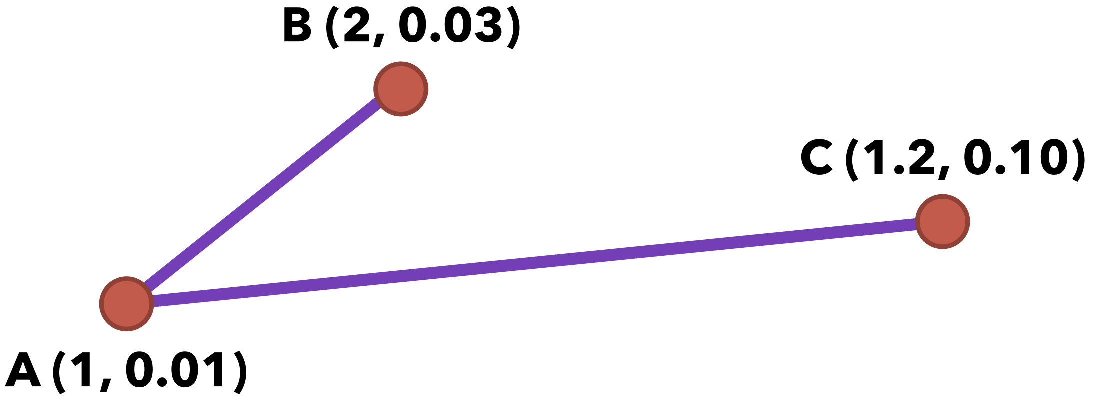
- **Mean interpolate** any unknown numeric values *(optional)*
- **Mode interpolate** any unknown categorical values *(optional)*
- **Remove** any completely unique columns (e.g., names, IDs…)
- **Convert** all categories to numeric values
  - Make need to consider the **values** of being part of **category**
  - E.g.: **extremely rare category** could be given a **weight stronger than 1**

PREPROCESSING: NORMALIZATION

Normalization of data is REQUIRED

Why?

- Distance is evaluated by the **relative difference between points**
- Points with **greater unit values** will be **more heavily weighted**

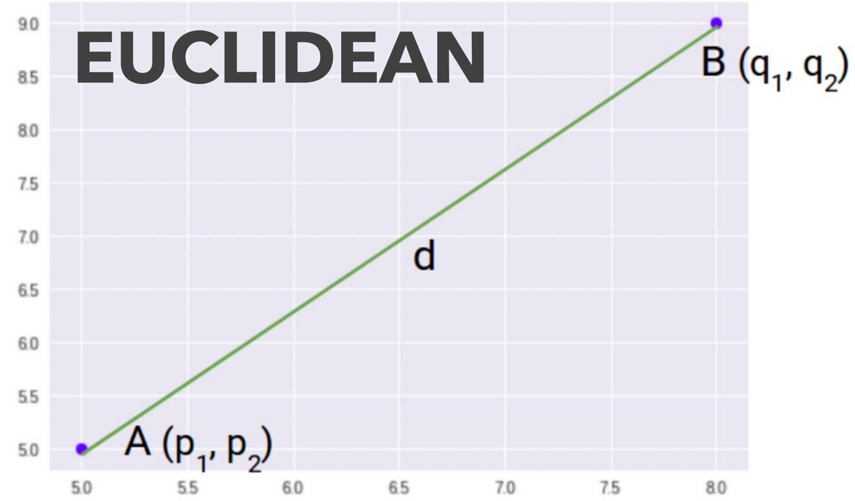


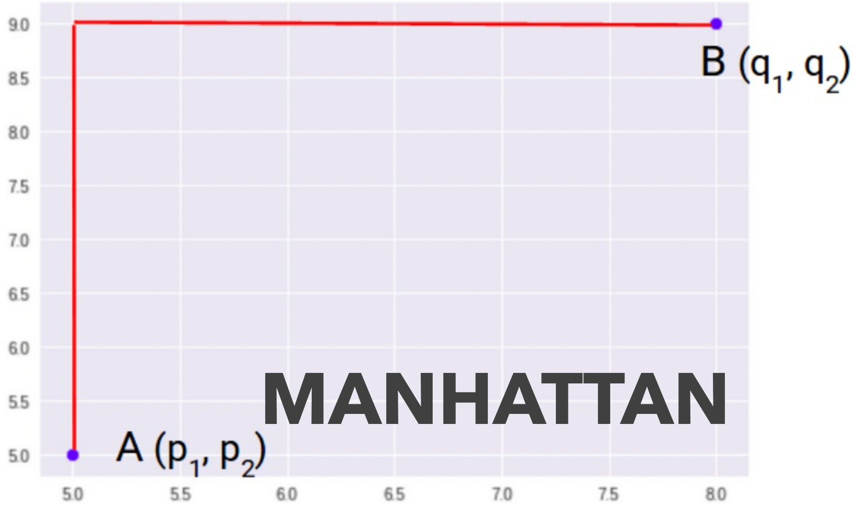
$$\overline{AB} = \sqrt{(2-1)^2 + (0.03-0.01)^2} = \mathbf{1.0002}$$

$$\overline{AC} = \sqrt{(1.2-1)^2 + (0.10-0.01)^2} = \mathbf{0.2193}$$

HYPERPARAMETERS: DISTANCE

Type	Description	Dim.	Equation
Manhattan	City block distance	1	$d(A,B) = \sum_{i=1}^{n} \lvert p_i - q_i \rvert$
Euclidean	Pythagorean shortest distance	2	$d(A,B) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$
Minkowski	General form of Manhattan and Euclidean	L	$d(A,B) = \sqrt[L]{\sum_{i=1}^{n} (\lvert p_i - q_i \rvert)^L}$

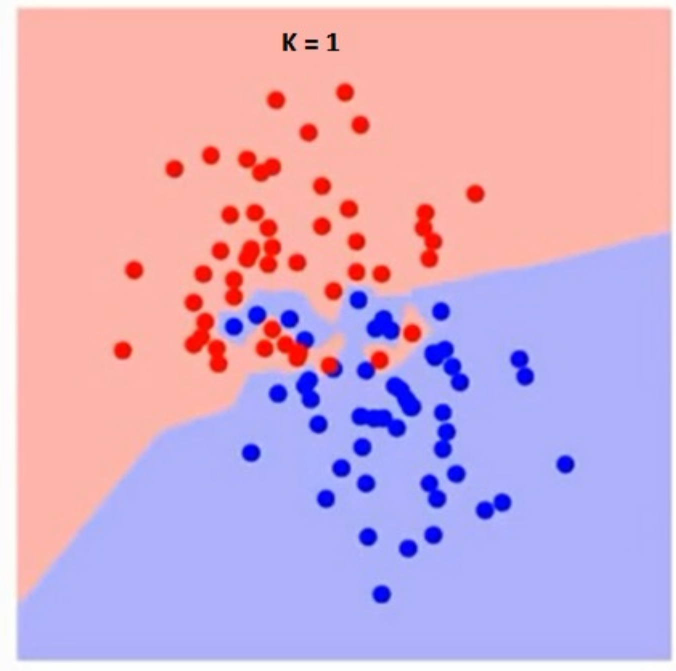




HYPERPARAMETERS: K

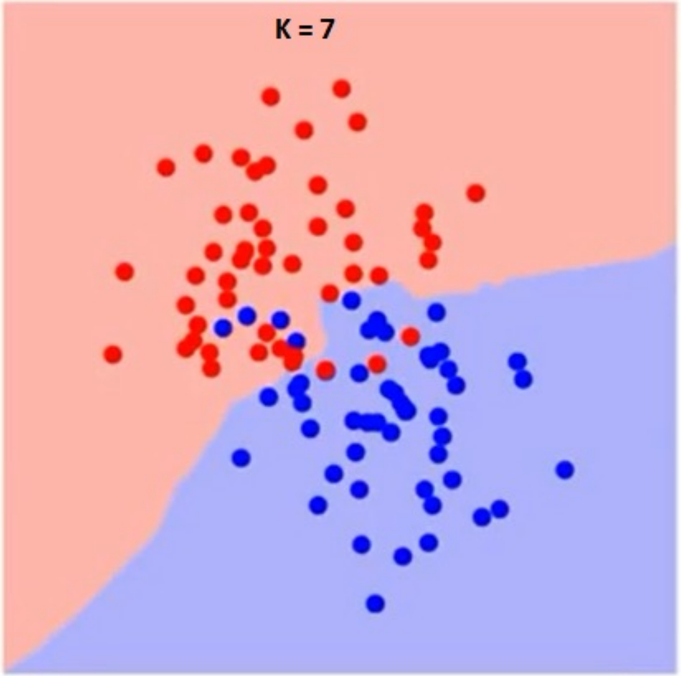
K: number of nearest neighbors involved in classification

- **Overfitted**: too **small** K
- **Underfitted**: too **large** K



OVERFIT

UNDERFIT



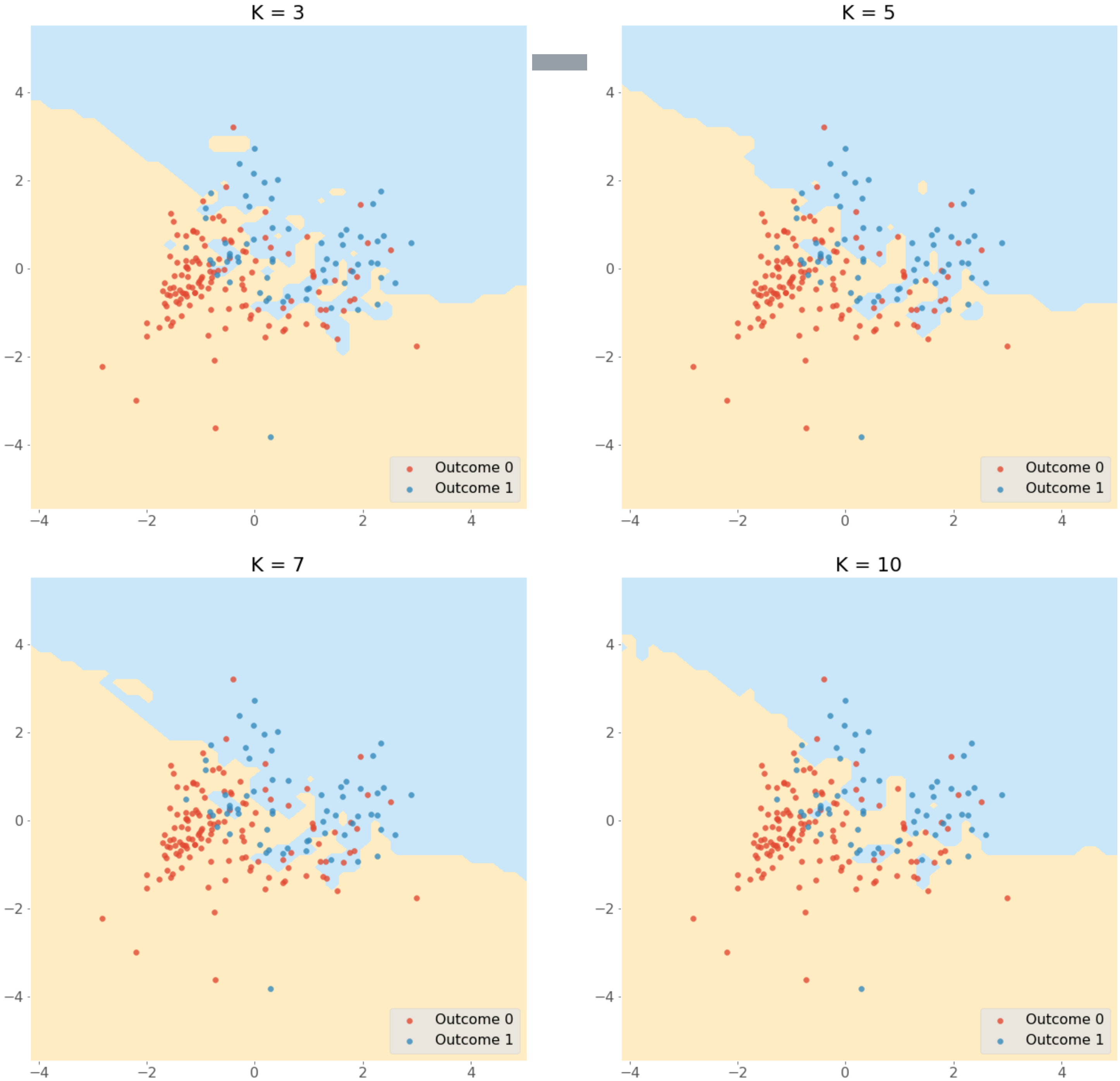
HYPERPARAMETERS: ALGORITHMS

- Change the **how distance is calculated** (to save time)
- Also tuned with **'leaf_size'**

Name	Description	Time Complexity	Dataset Size	Dataset Sparsity	Accuracy
'brute'	Compute all distances	$O(DN^2)$ - *cubic*	Small	Sparse	Highest Accuracy
'kd_tree'	Approximate distances with KD tree	$O(DN)$ – *quadratic*	Large	Dense	High Accuracy
'ball_tree'	Approximate distances with ball tree	$O(D\log N)$ - *nlog(n)*	Very Large	Dense	Decent Accuracy

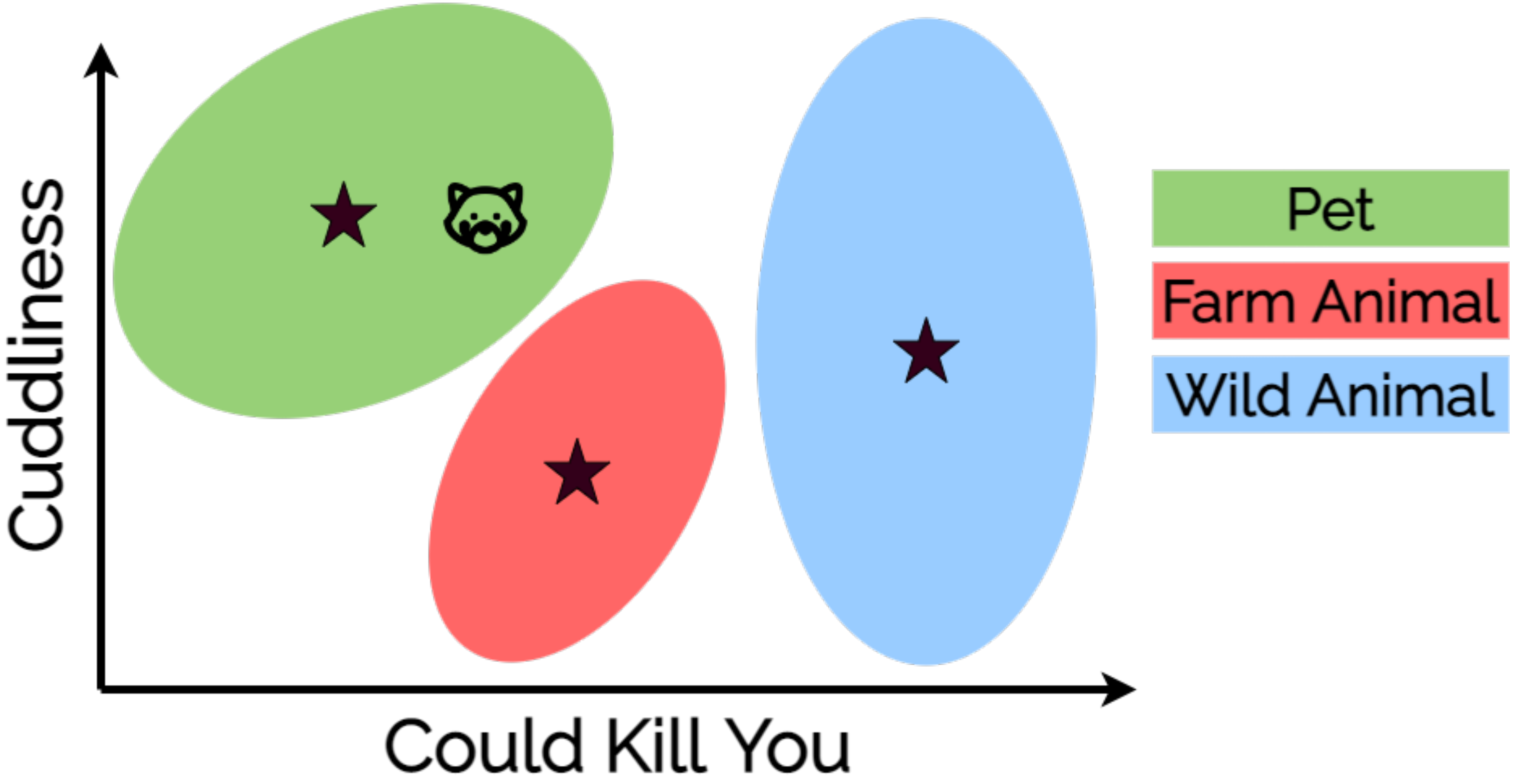
BENEFITS

- **Intuitive** and human friendly
- Allows user to **chose** the **hyperparameters**
- Memory-based
- Both **classification** and **regression**
- Ease after establishing hyperparameters
- **Non-parametric**



TRADE-OFFS AND LIMITATIONS

- **Non-parametric**
- **Slow** to implement
- **Dimensionality**
- Requires **homogenous** features
- **Sensitive to outliers**



QUESTIONS?