



# K Nearest Neighbors



- KNN (K nearest neighbors) is one of the simplest algorithms we will learn about!
- Section Overview
  - KNN Theory and Intuition
  - KNN Classification Coding Example
  - KNN Exercise Overview
  - KNN Exercise Solution



- While KNN can be used for regression tasks, its performance can be quite poor **and** less efficient than other algorithms, so we've decided not to exhibit its use for regression.
- However if you do want to use it for regression it is very easy to swap in the KNNRegressor model with scikit-learn.



- You may have also heard of K means algorithm.
- K means is unrelated to KNN, be careful not to confuse the two due to their similar sounding names!

- ISLR Relevant Reading
  - Chapter 2
  - Formula 2.12 starts discussion on KNN for classification.

$$\Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j).$$



# KNN Classification

Theory and Intuition

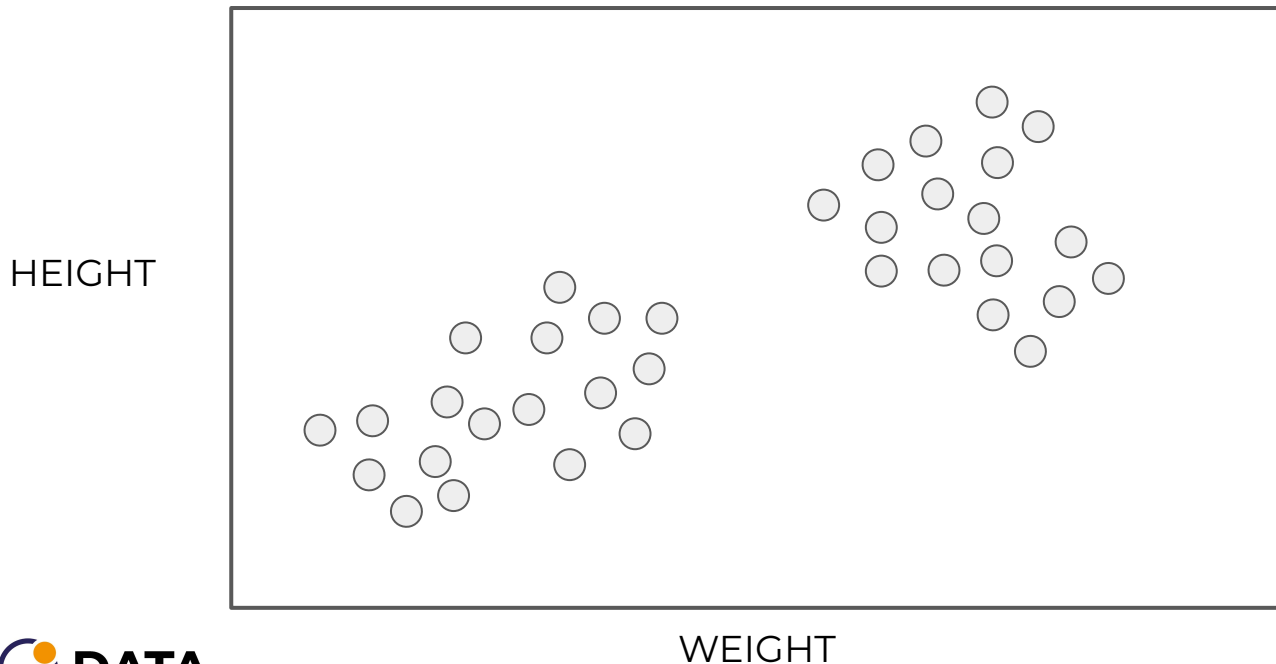


- K nearest neighbors is one of the simplest machine learning algorithms.
- It simply assigns a label to new data based on the **distance** between the old data and new data.
- Let's go through the intuition with an example use case...

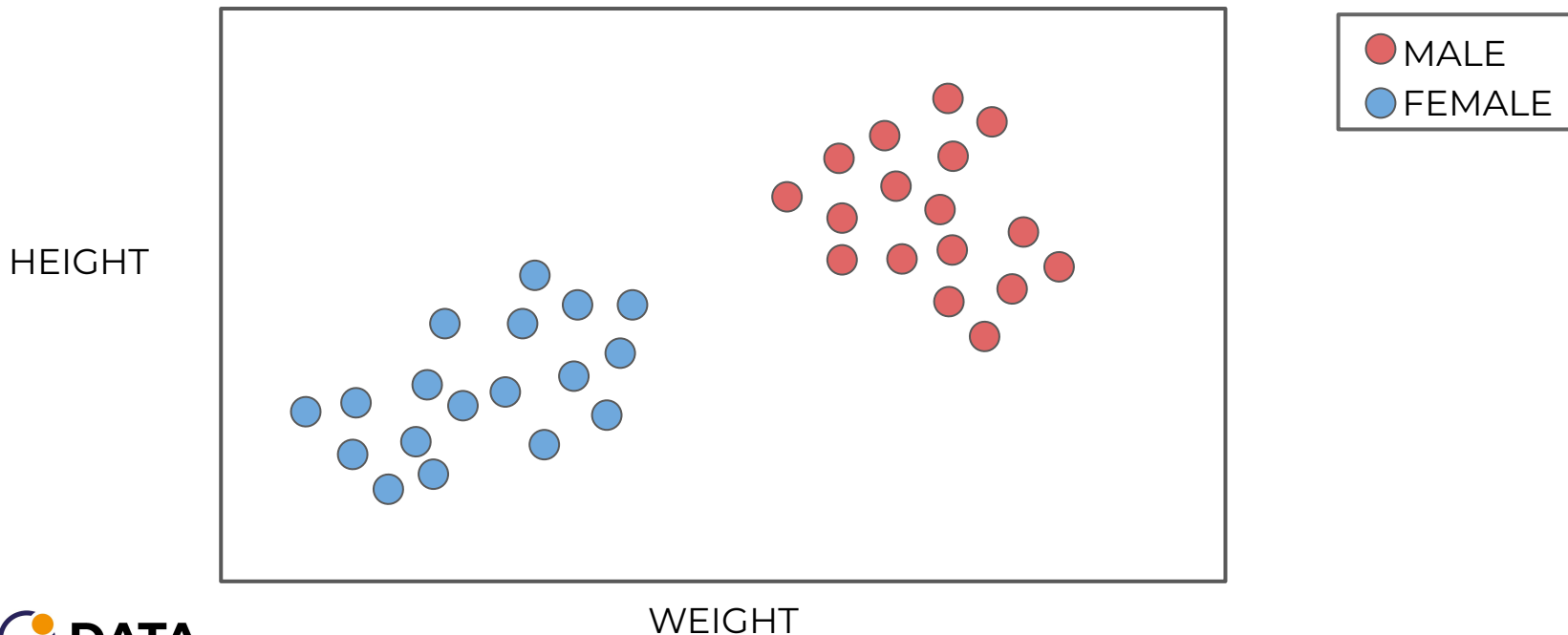
- Sexing chicks is still a very manual process:
  - [en.wikipedia.org/wiki/Chick\\_sexing](https://en.wikipedia.org/wiki/Chick_sexing)
- Let's imagine we gathered a dataset of baby chick heights and weights.
- How could we train an algorithm to identify the sex of a new baby chick based on historical features?



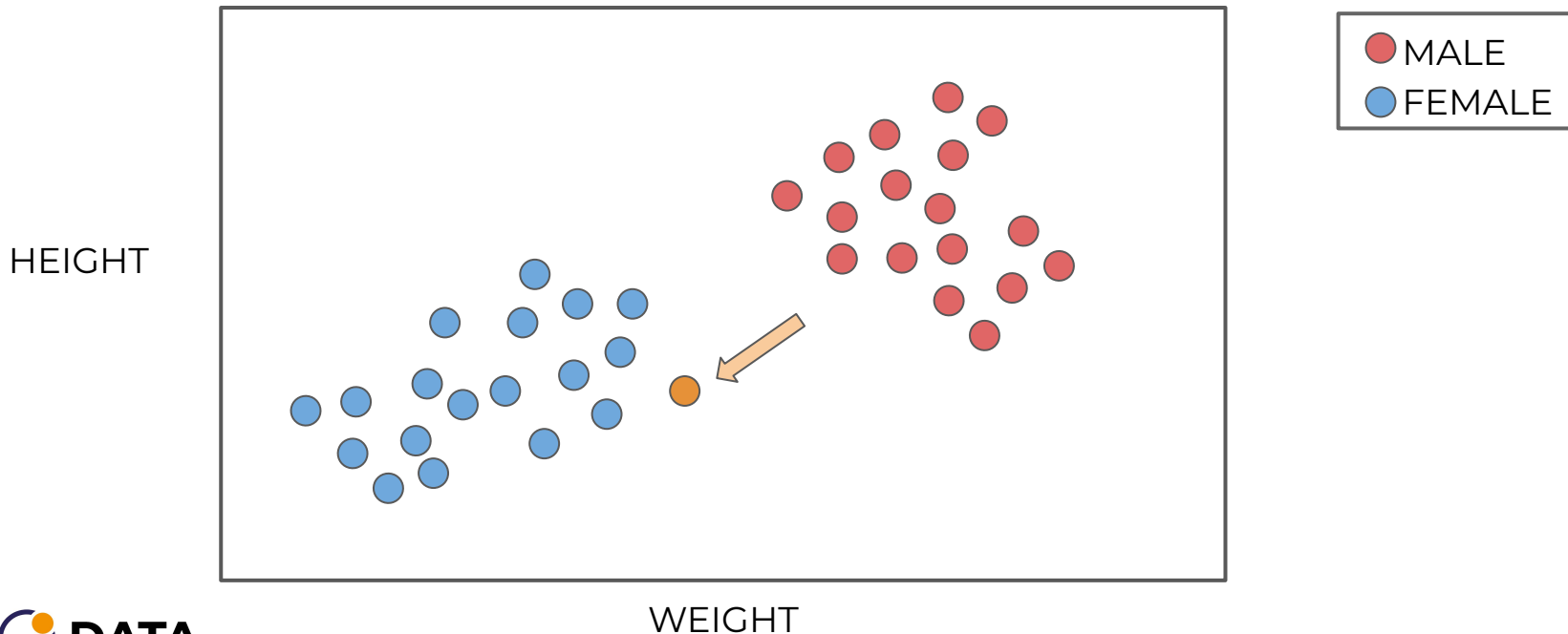
- Imagine a height and weight data set



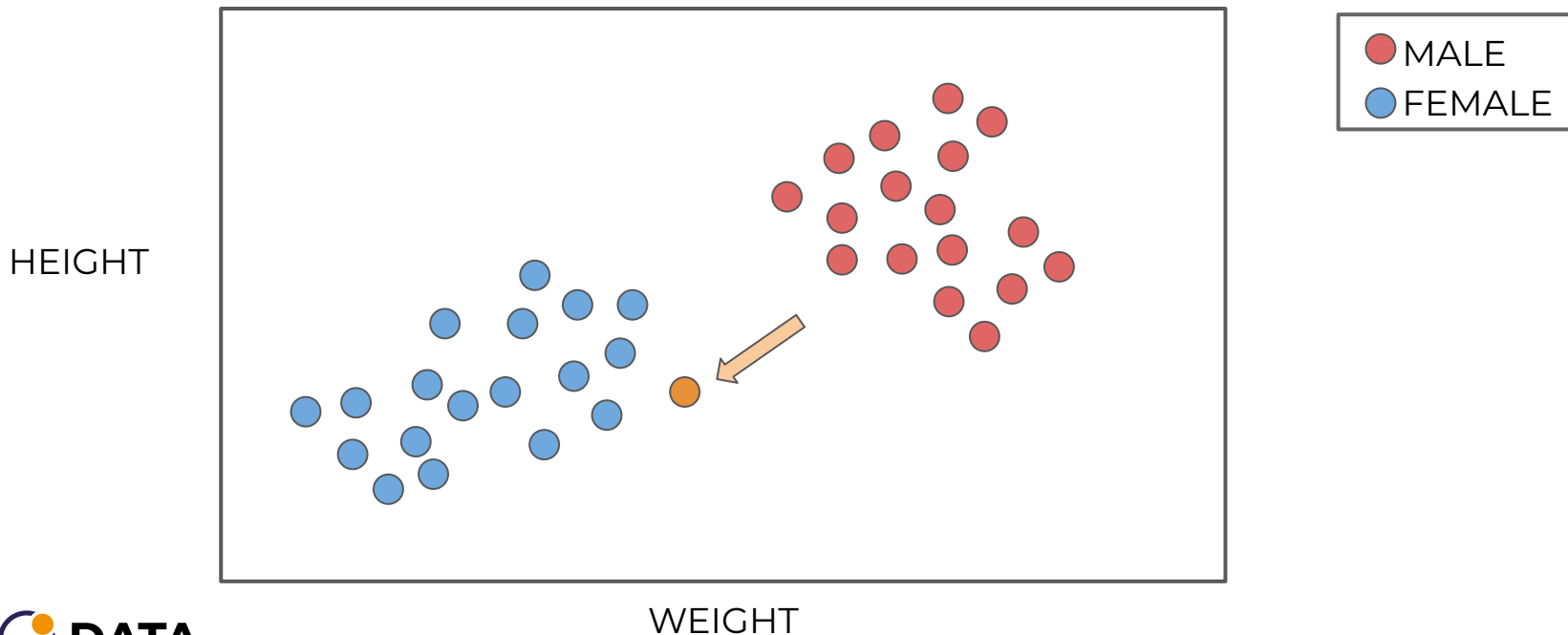
- We historically know the sex of the chicks:



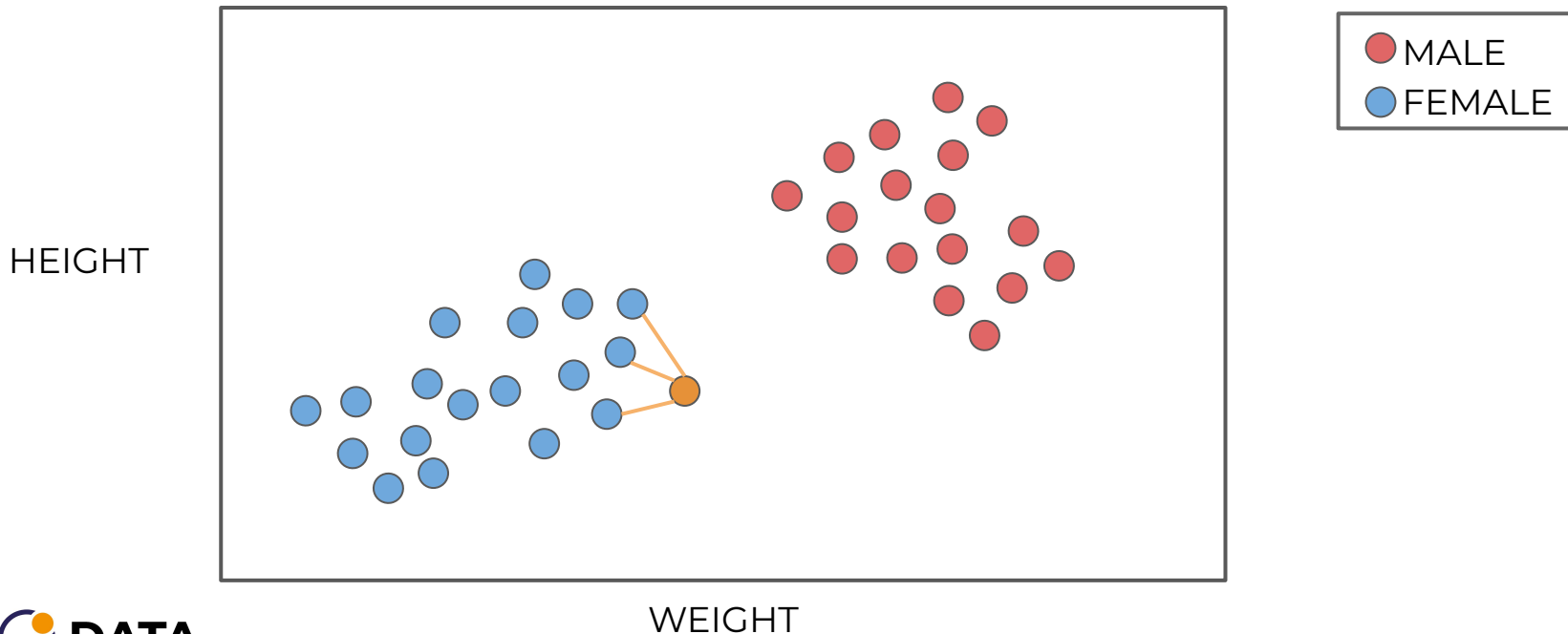
- How would we assign sex to a new point?



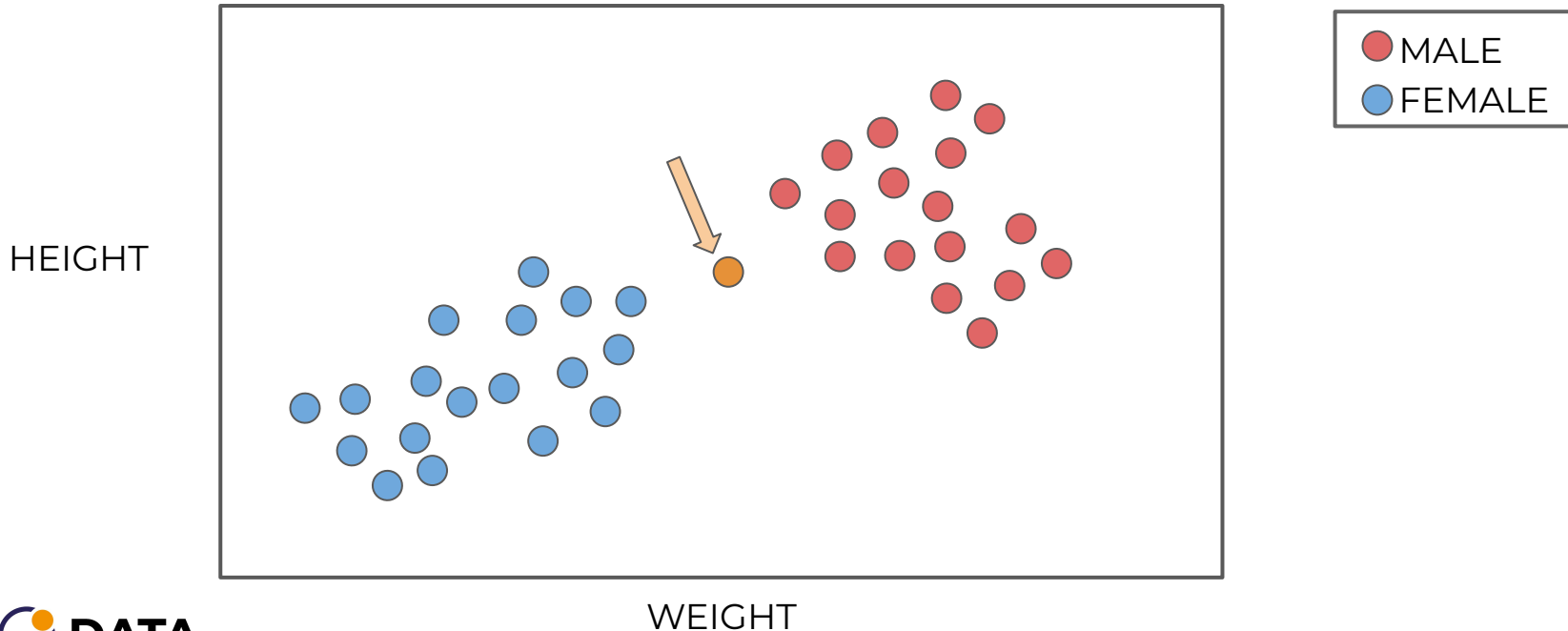
- We intuitively “know” this is likely female.



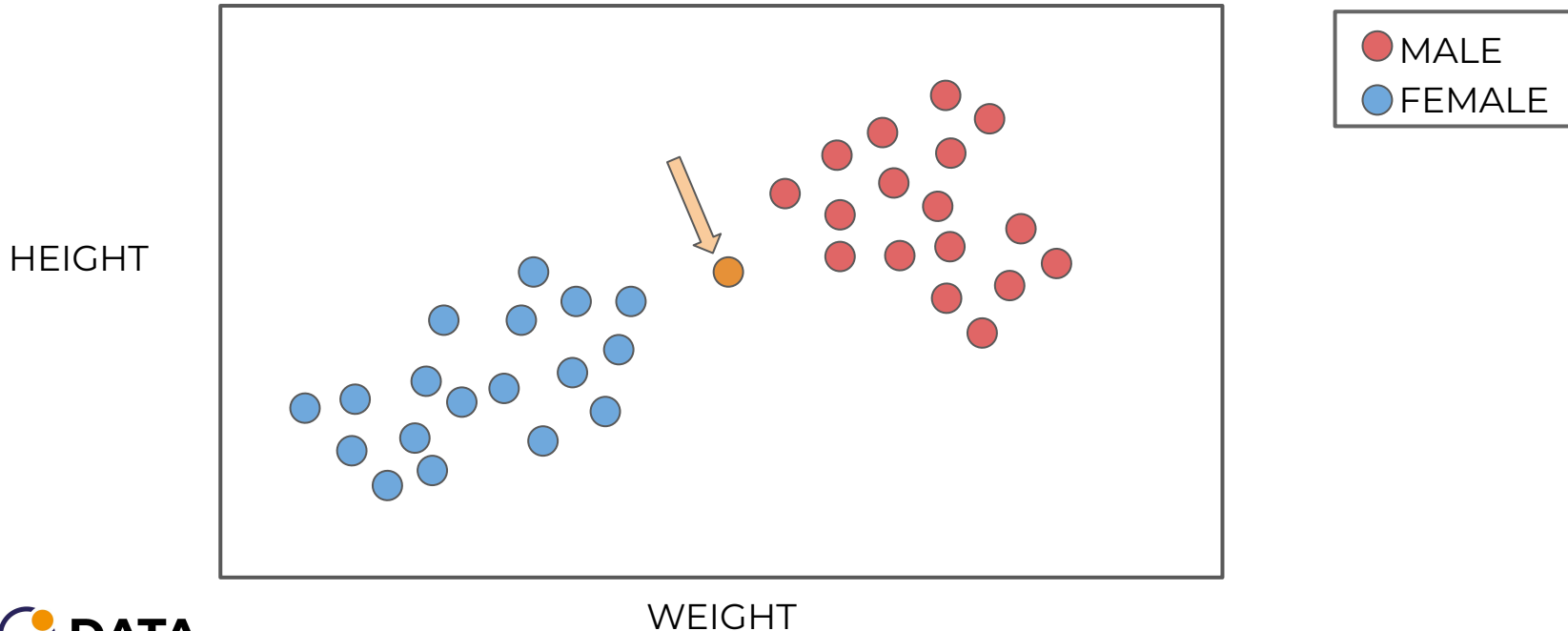
- Intuition comes from **distance** to points!



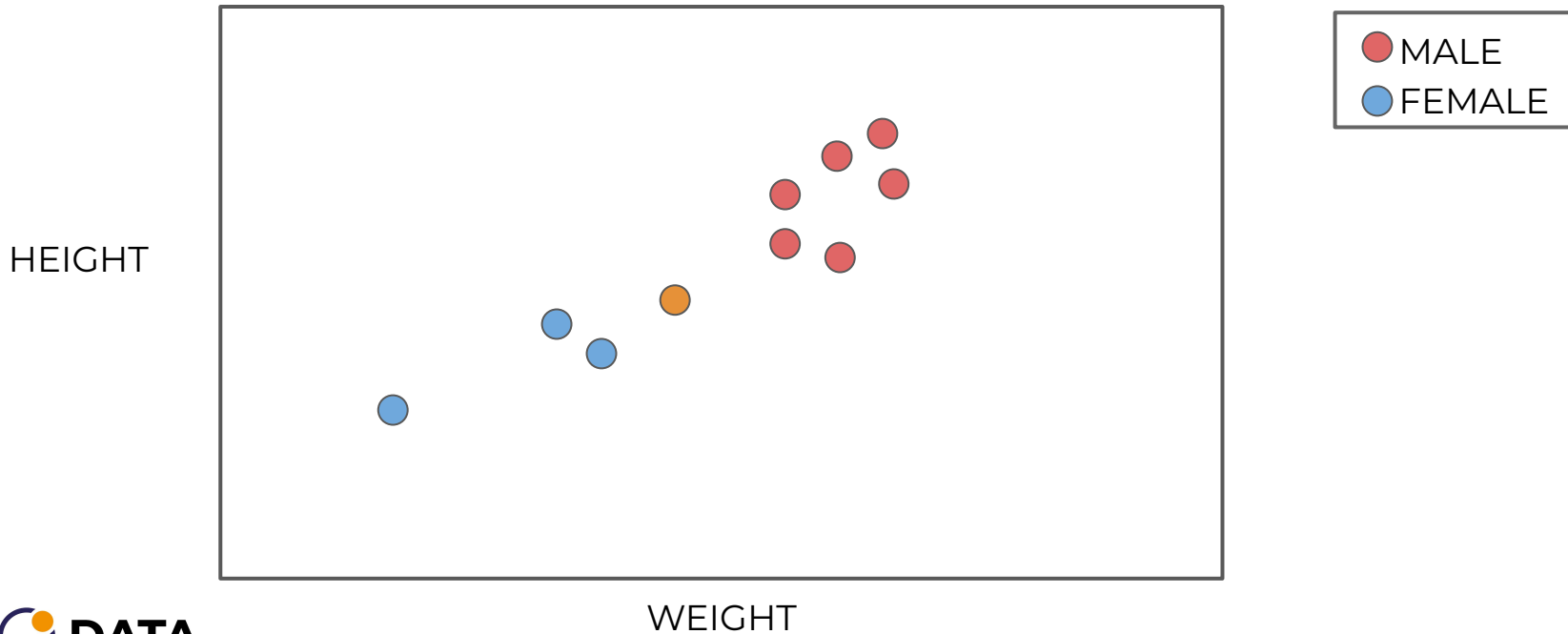
- What about a less obvious point?



- How many points to we consider?



- Let's imagine a situation like this:

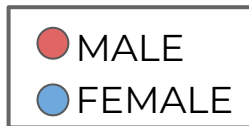
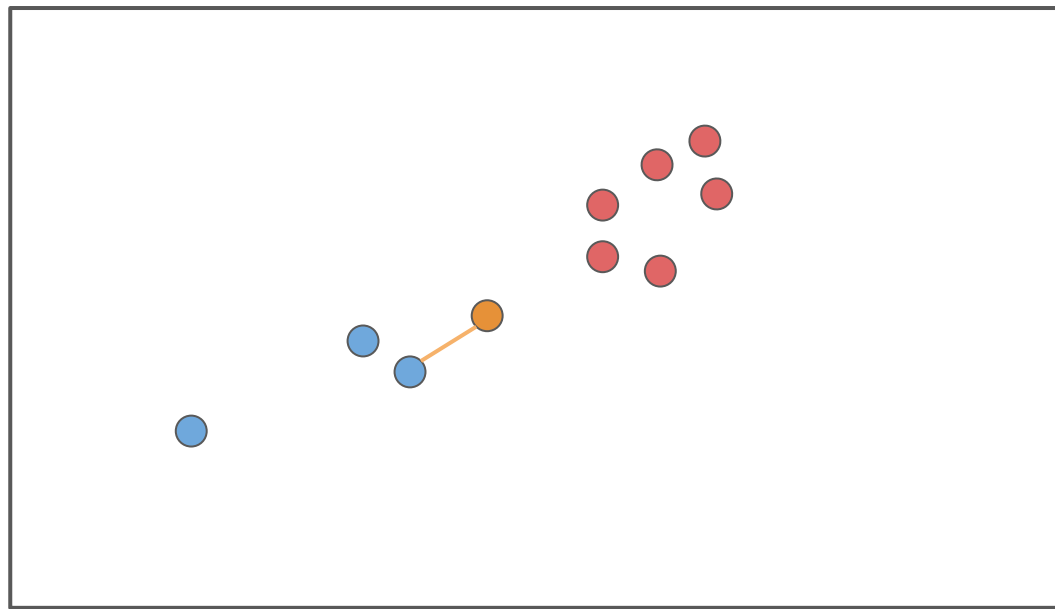






- $K=1$

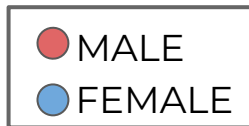
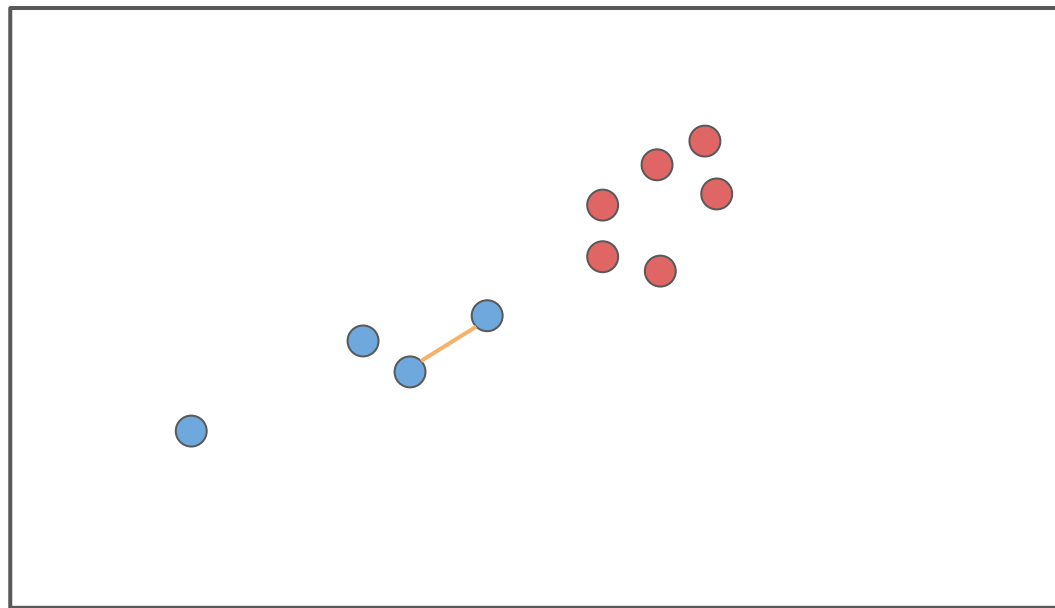
HEIGHT





- $K=1$

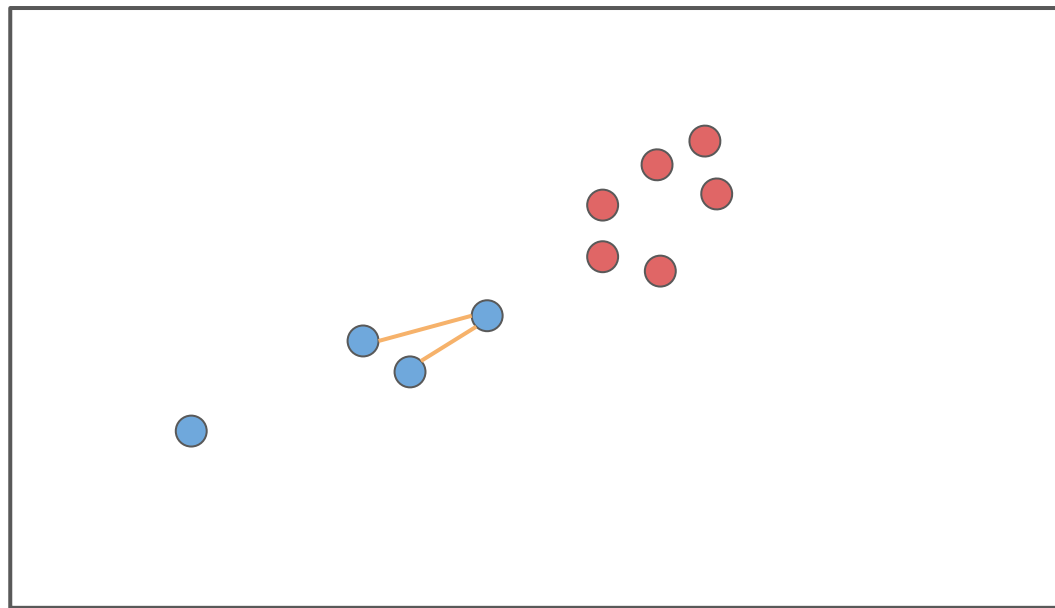
HEIGHT





- $K=2$

HEIGHT

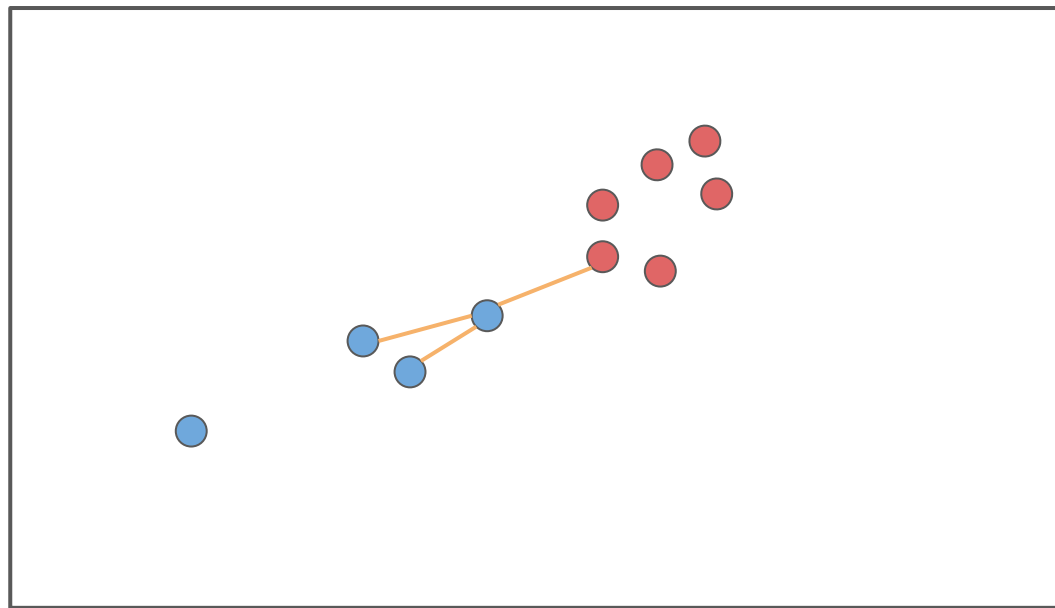


MALE  
FEMALE



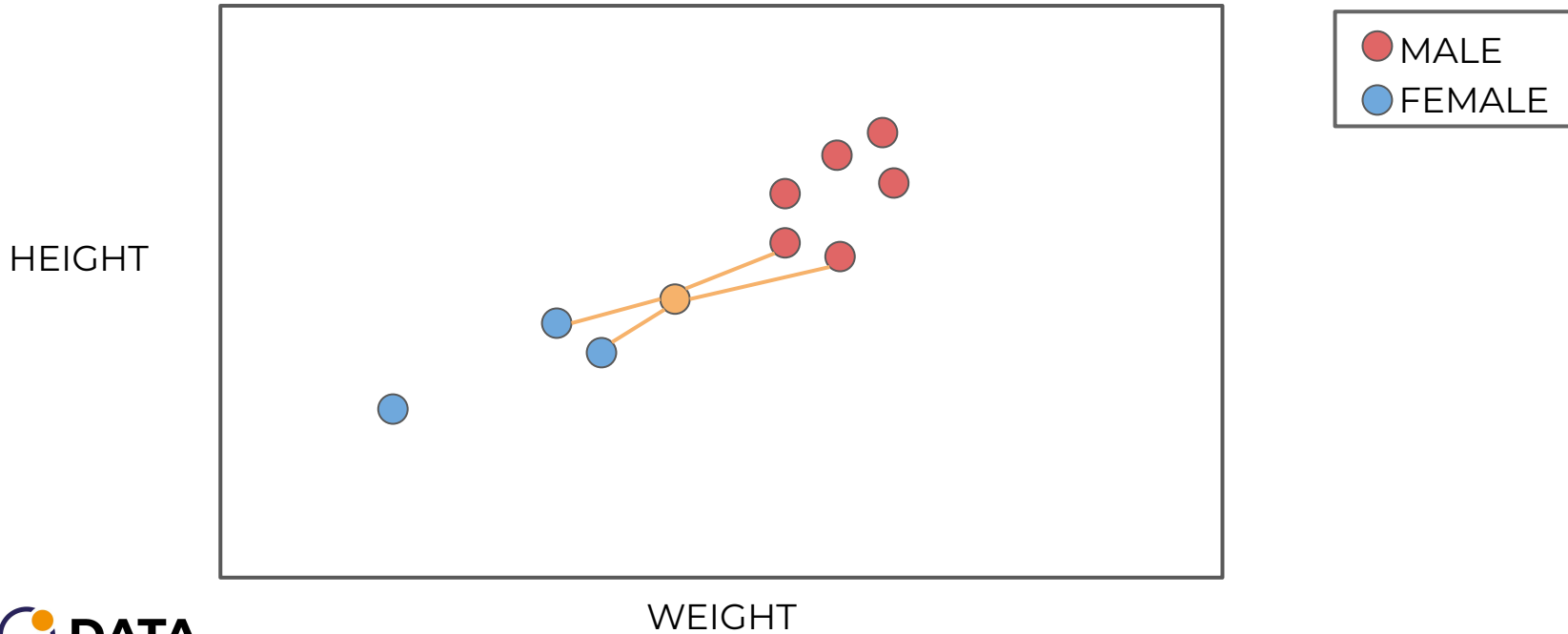
- $K=3$

HEIGHT



MALE  
FEMALE

- K=4 leads to a tie!





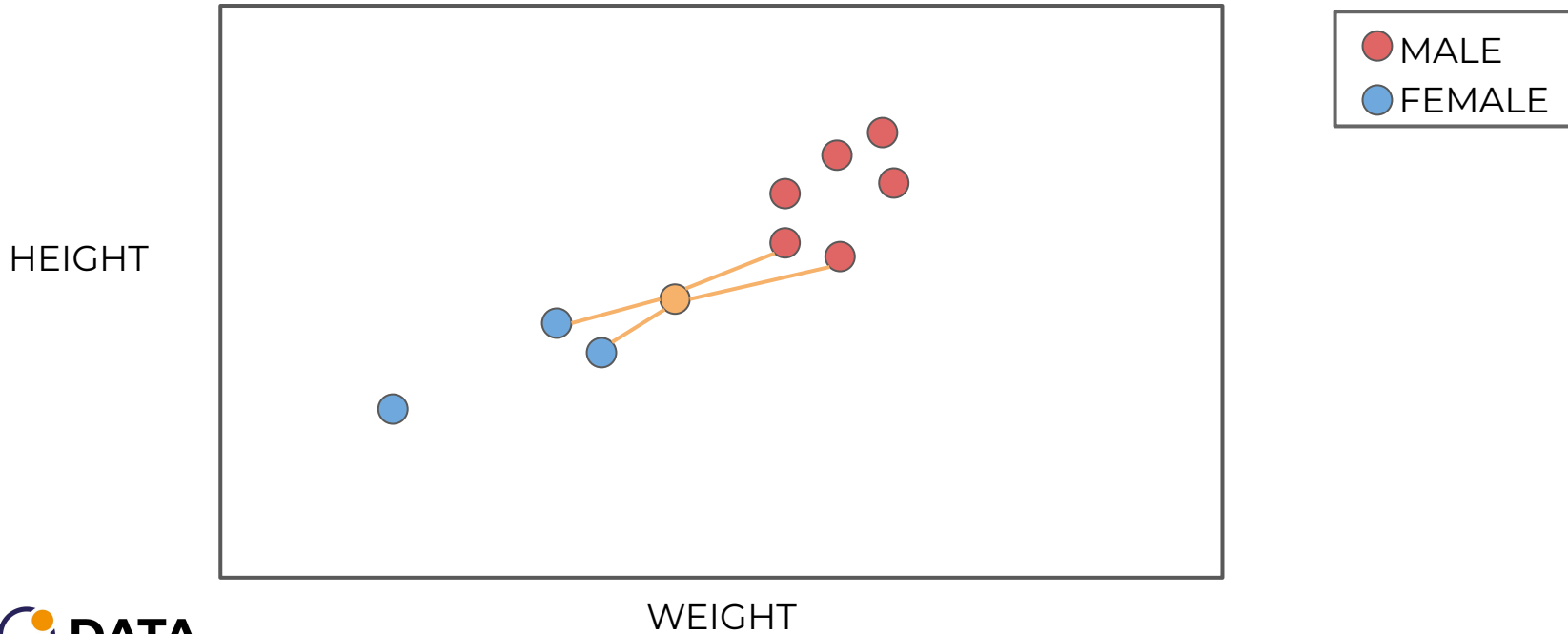
- Tie considerations and options:
  - Always choose an odd K.
  - In case of tie, simply reduce K by 1 until tie is broken.
  - Randomly break tie.
  - Choose nearest class point.

- What does Scikit-Learn do in case of tie?
  - *Warning: Regarding the Nearest Neighbors algorithms, if it is found that two neighbors, neighbor  $k+1$  and  $k$ , have identical distances but different labels, the results will depend on the ordering of the training data.*

- What does Scikit-Learn do in case of tie?
  - *In the case of ties, the answer will be the class that happens to appear first in the set of neighbors.*
  - *Results are ordered by distance, so it chooses the class of the closest point.*

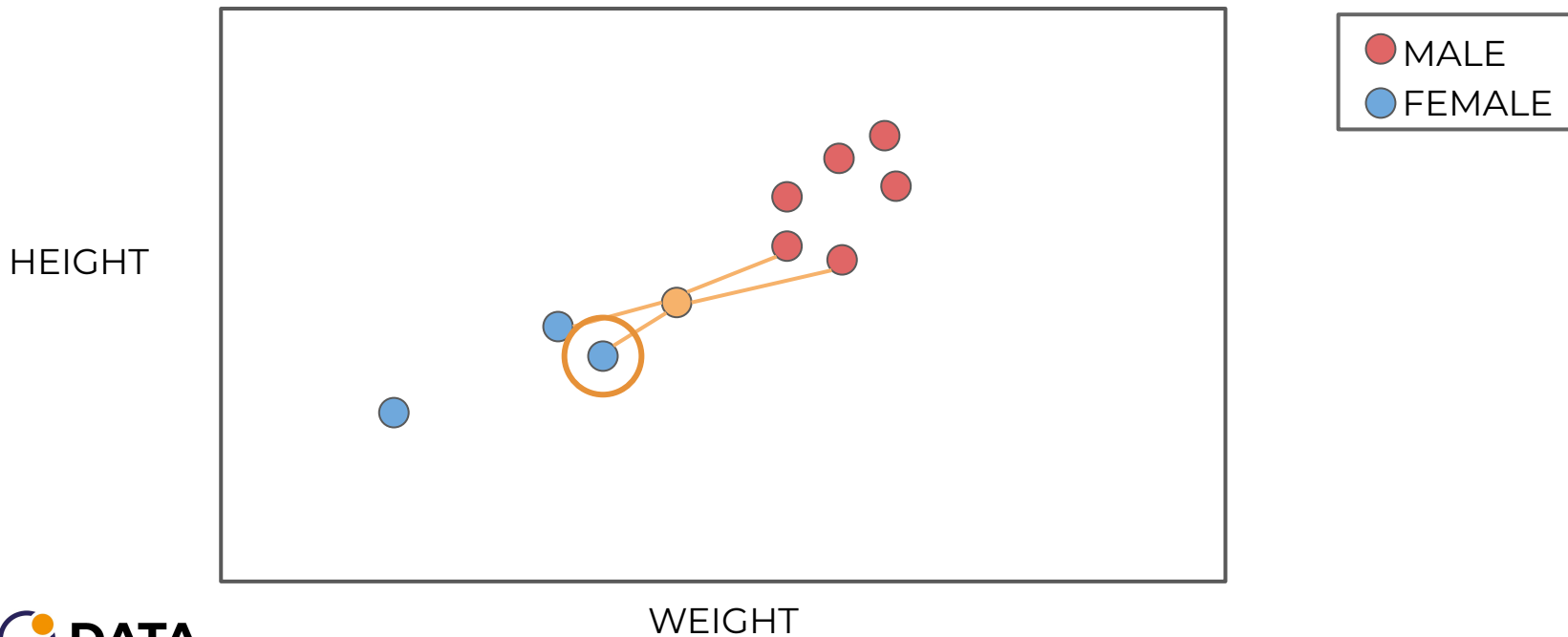


- $K=4$  leads to a tie!



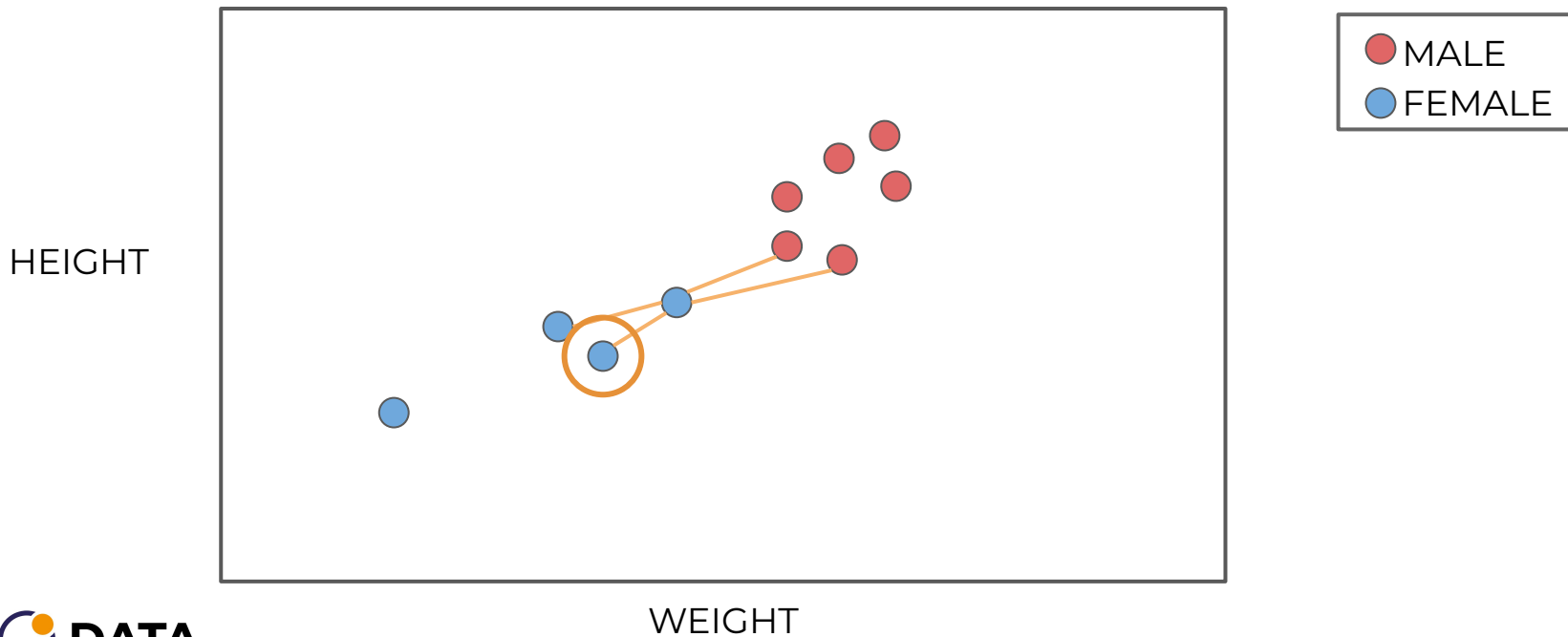


- Choose closest K

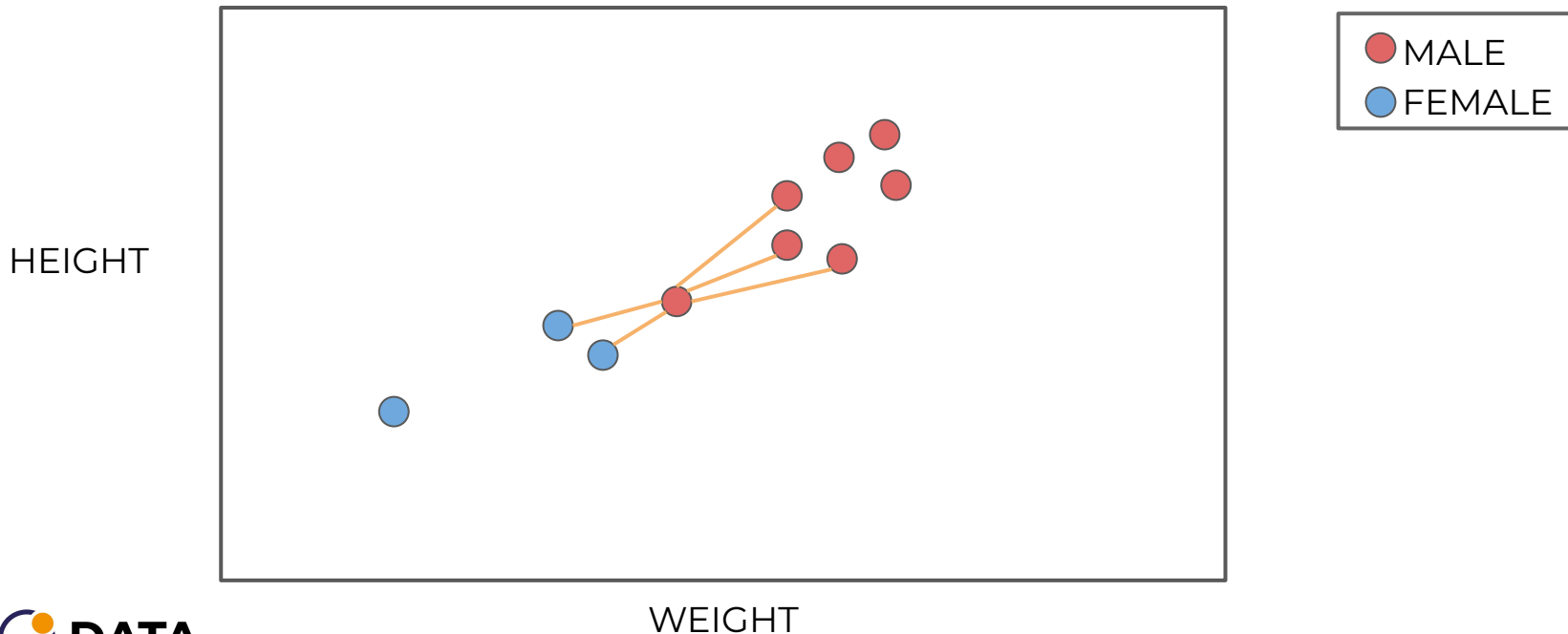




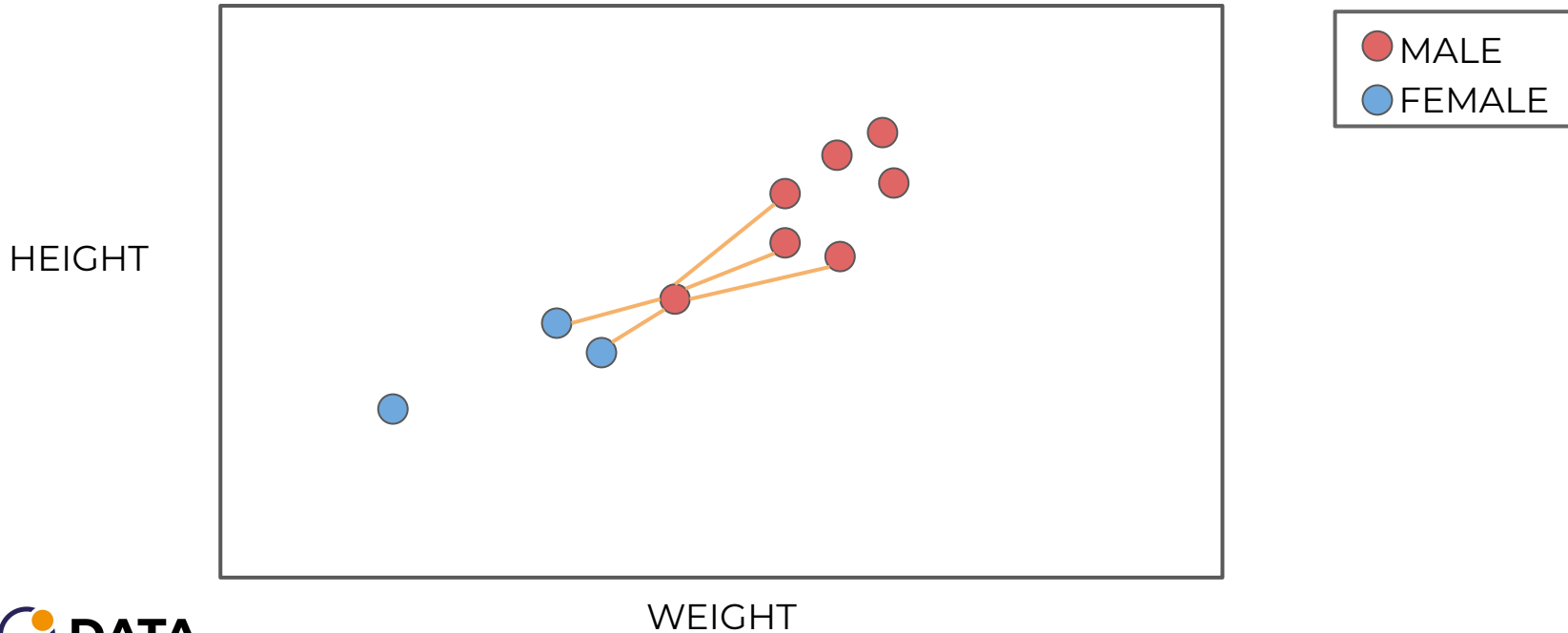
- Choose closest K



- K=5 causes a switch from previous K values.



- How to choose best K value?

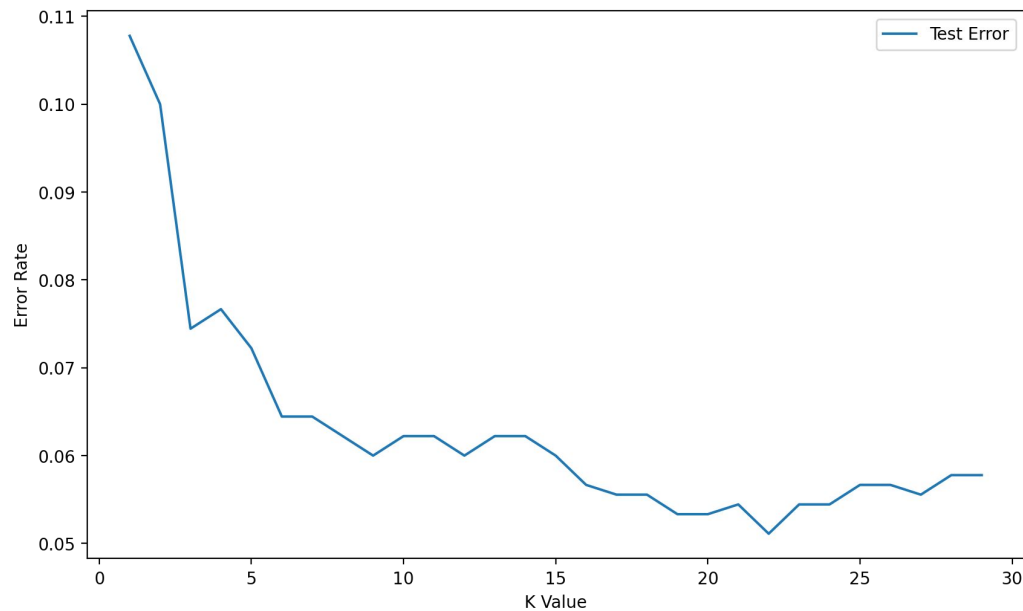




- We want a K value that **minimizes** error:
  - $\text{Error} = 1 - \text{Accuracy}$
- Two methods:
  - Elbow method.
  - Cross validate a grid search of multiple K values and choose K that results in lowest error or highest accuracy.

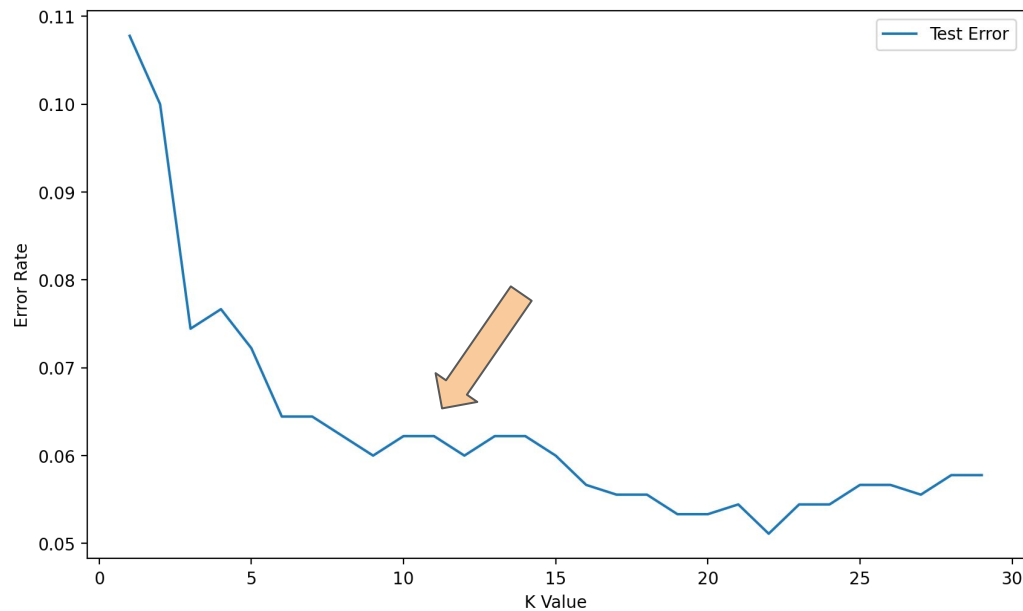


- Elbow method:





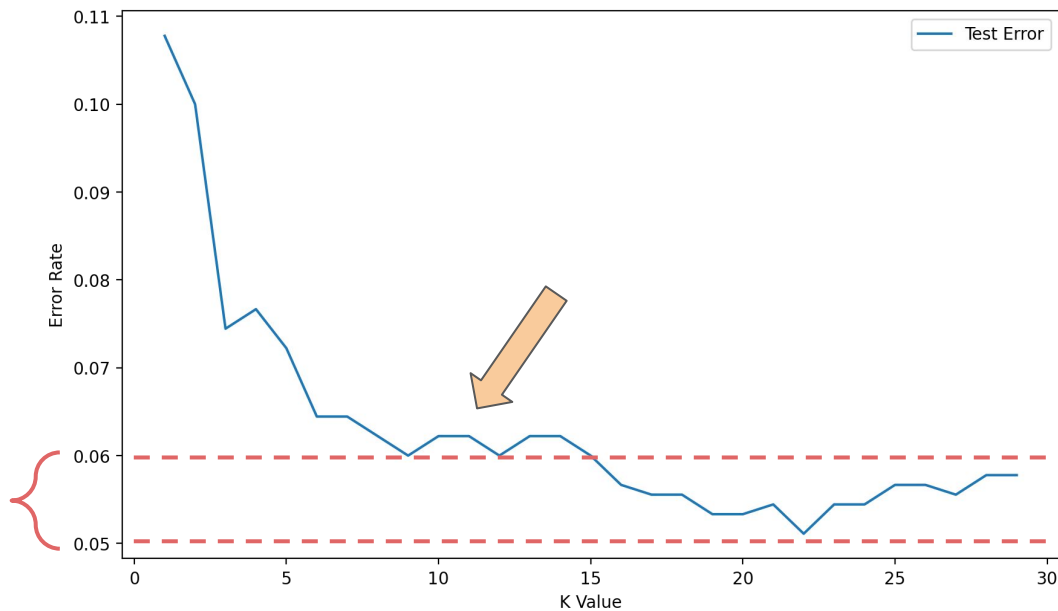
- Elbow method:







- Elbow method:





- Cross validation only takes into account the K value with the lowest error rate across multiple folds.
- This could result in a more complex model (higher value of K).
- Consider the context of the problem to decide if larger K values are an issue.



- KNN Algorithm
  - Choose K value.
  - Sort feature vectors (N dimensional space) by distance metric.
  - Choose class based on K nearest feature vectors.



- KNN Considerations:
  - Distance Metric
    - Many ways to measure distance:
      - Minkowski
      - Euclidean
      - Manhattan
      - Chebyshev



- KNN Considerations:
  - Scaling for Distance
    - Features could have vastly different value ranges!

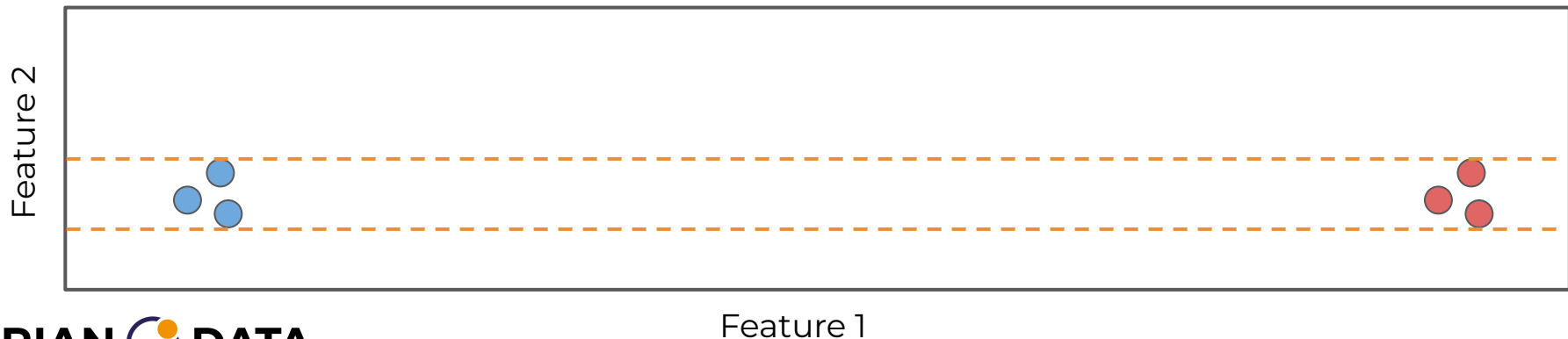


- KNN Considerations:
  - Scaling for Distance
    - Features could have vastly different value ranges!



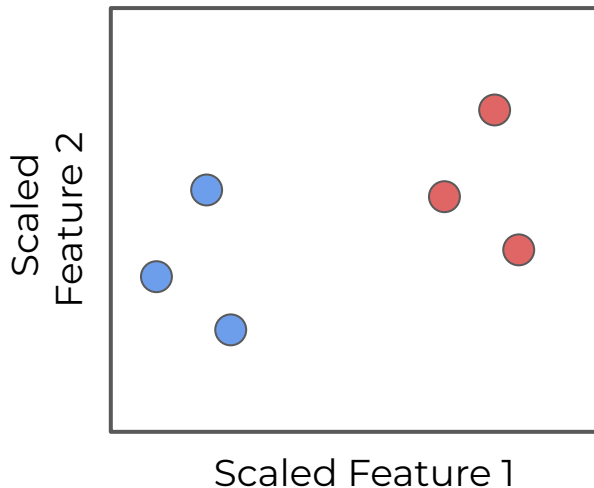


- KNN Considerations:
  - Scaling for Distance
    - Features could have vastly different value ranges!





- KNN Considerations:
  - Scaling is necessary for KNN.







- While the KNN Algorithm is relatively simple, keep in mind the following considerations:
  - Choosing the optimal K value.
  - Scaling features.
- Let's continue to explore how to perform KNN for classification!



# KNN Classification

Coding Part One: Data and Model



# KNN Classification

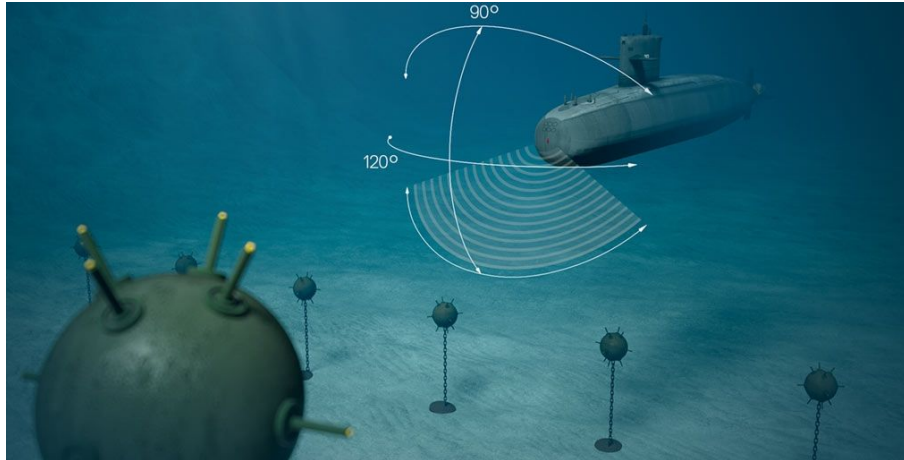
Coding Part Two: Choosing K

- A Pipeline object in Scikit-Learn can set up a sequence of repeated operations, such as a scaler and a model.
- This way only the pipeline needs to be called, instead of having to repeatedly call a scaler and a model.



# KNN Exercise Overview

- Let's test your new skills on a real data set.
- We'll be analyzing sonar frequencies to help distinguish between rocks or sea mines!





# KNN Exercise Solutions