K Nearest Neighbors

KNN

- 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

KNN

- 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.

KNN

- 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!

KNN

- 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

KNN

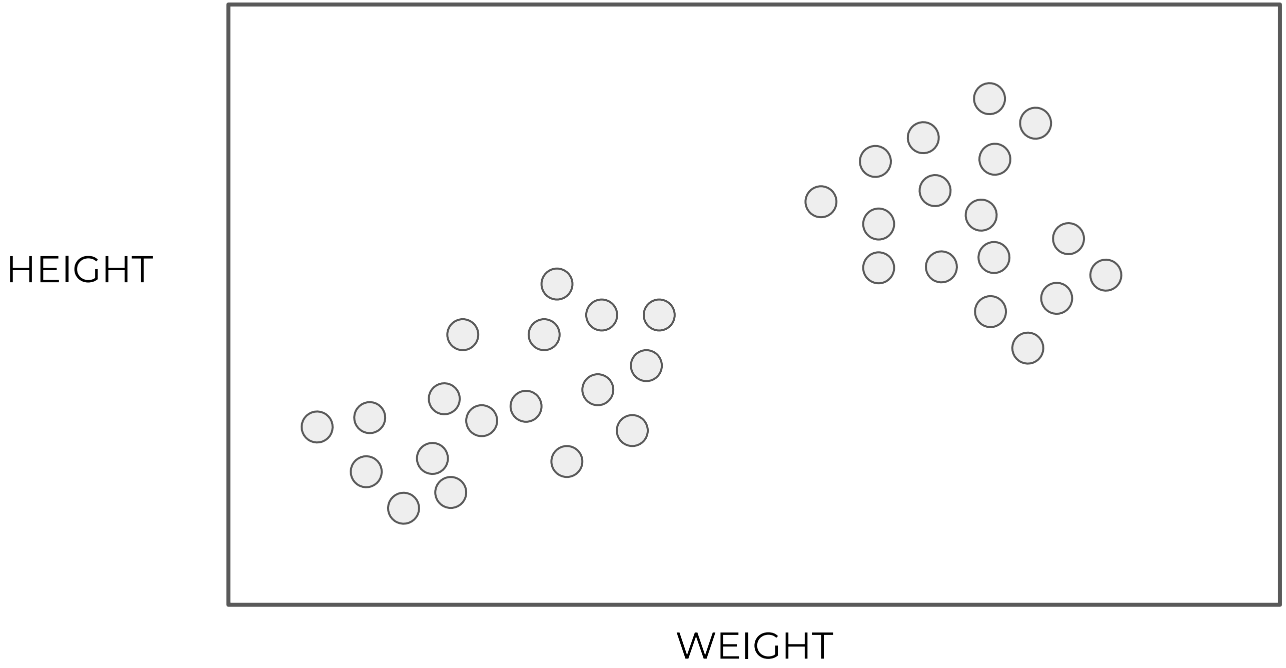
- K nearest neighbors is one of the simplest machine learning algorithms.
- It simply assigns a label to new data based on the **distance** between the old data and new data.
- Let’s go through the intuition with an example use case...

KNN

- Sexing chicks is still a very manual process:
  - 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?

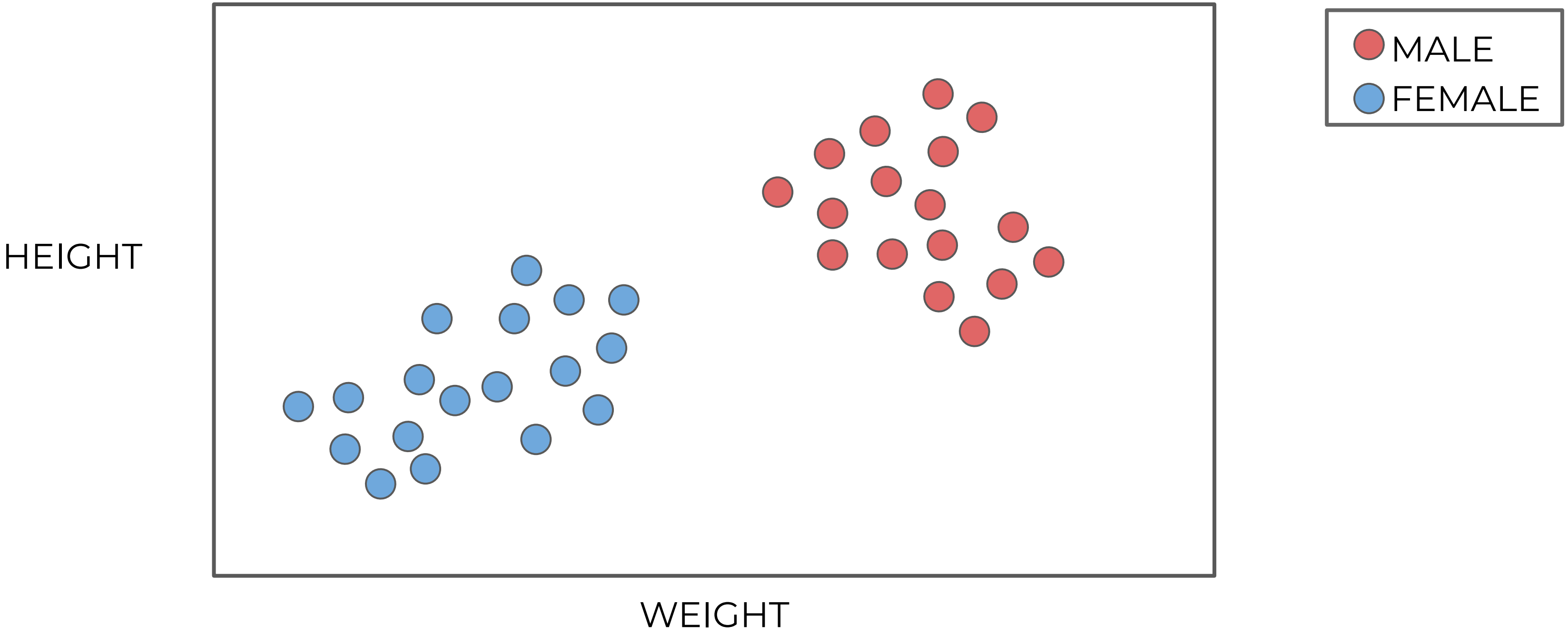
KNN

- Imagine a height and weight data set



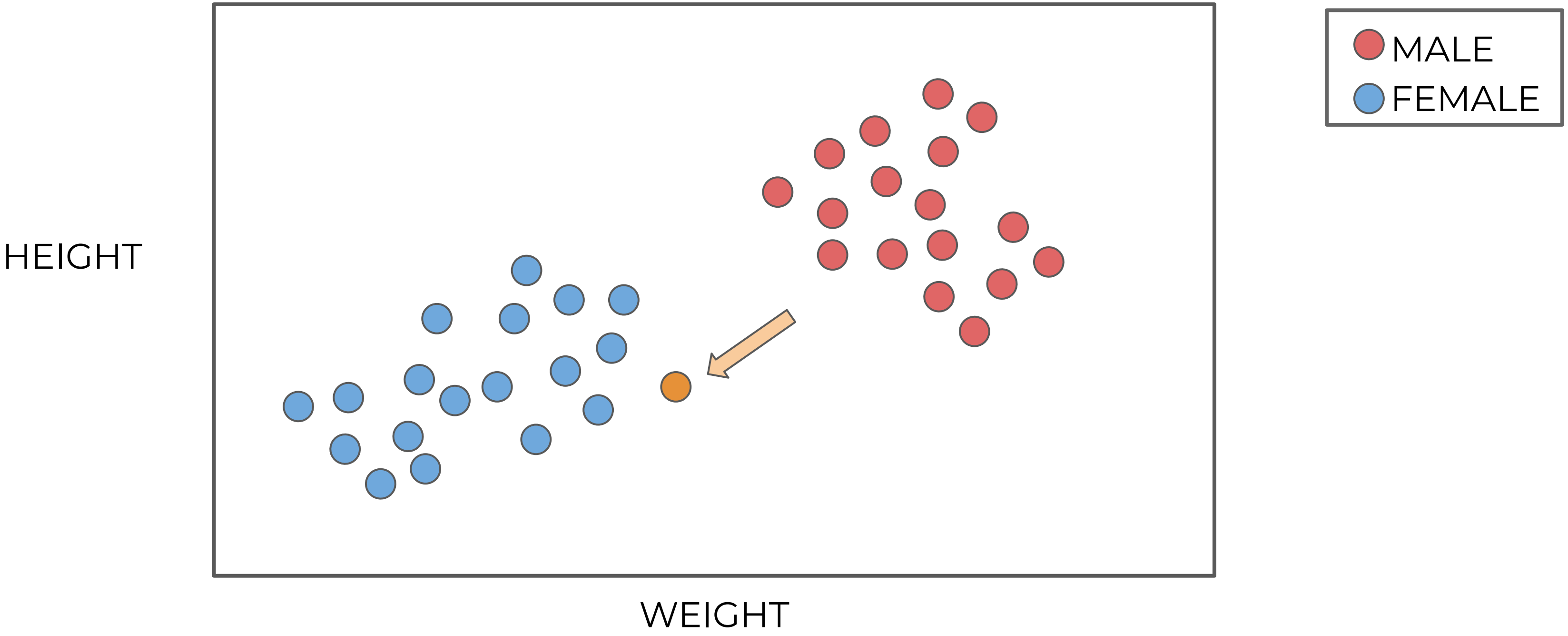
KNN

- We historically know the sex of the chicks:



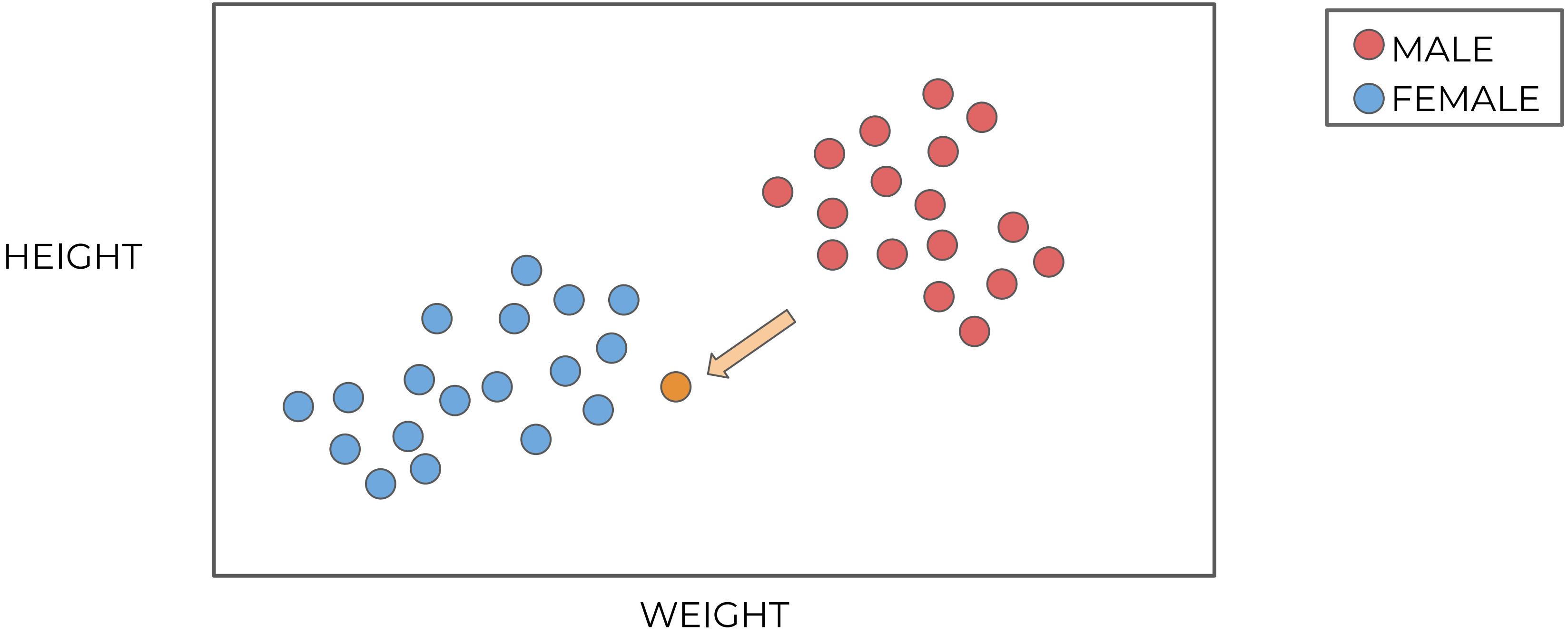
KNN

- How would we assign sex to a new point?



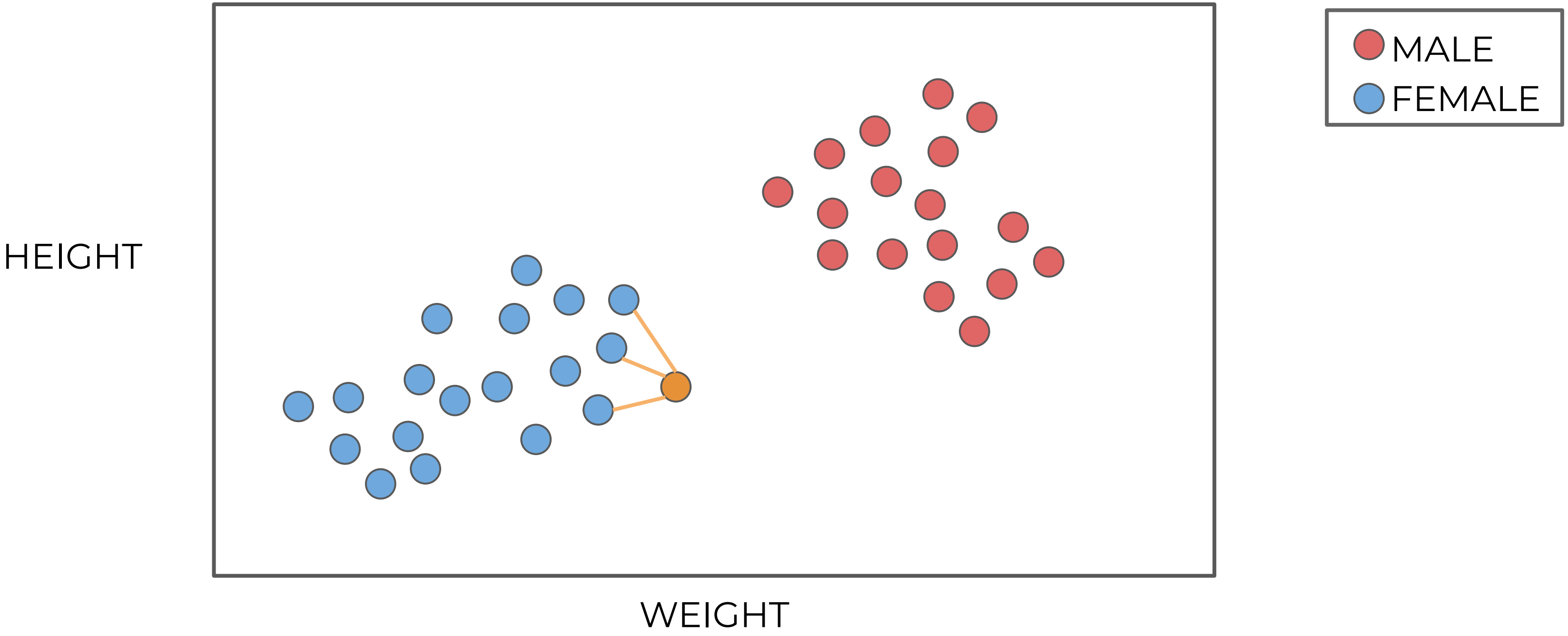
KNN

- We intuitively “know” this is likely female.



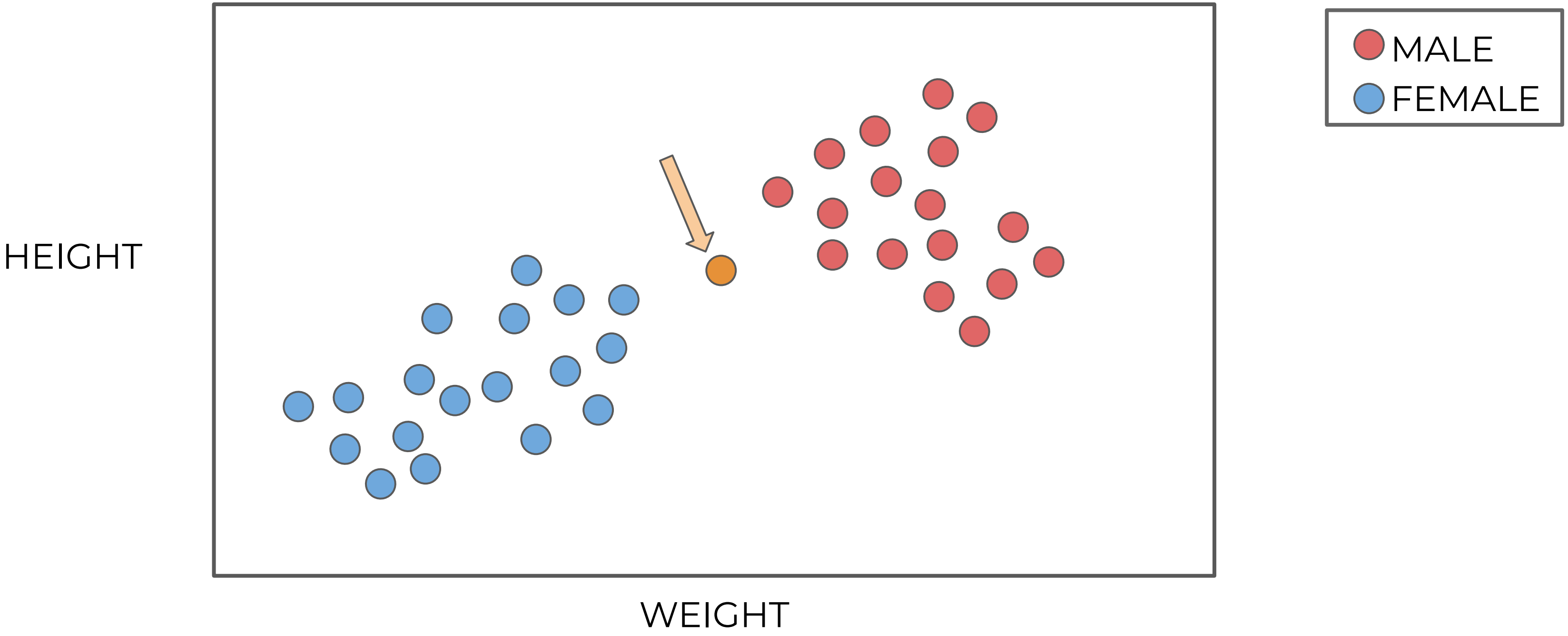
KNN

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



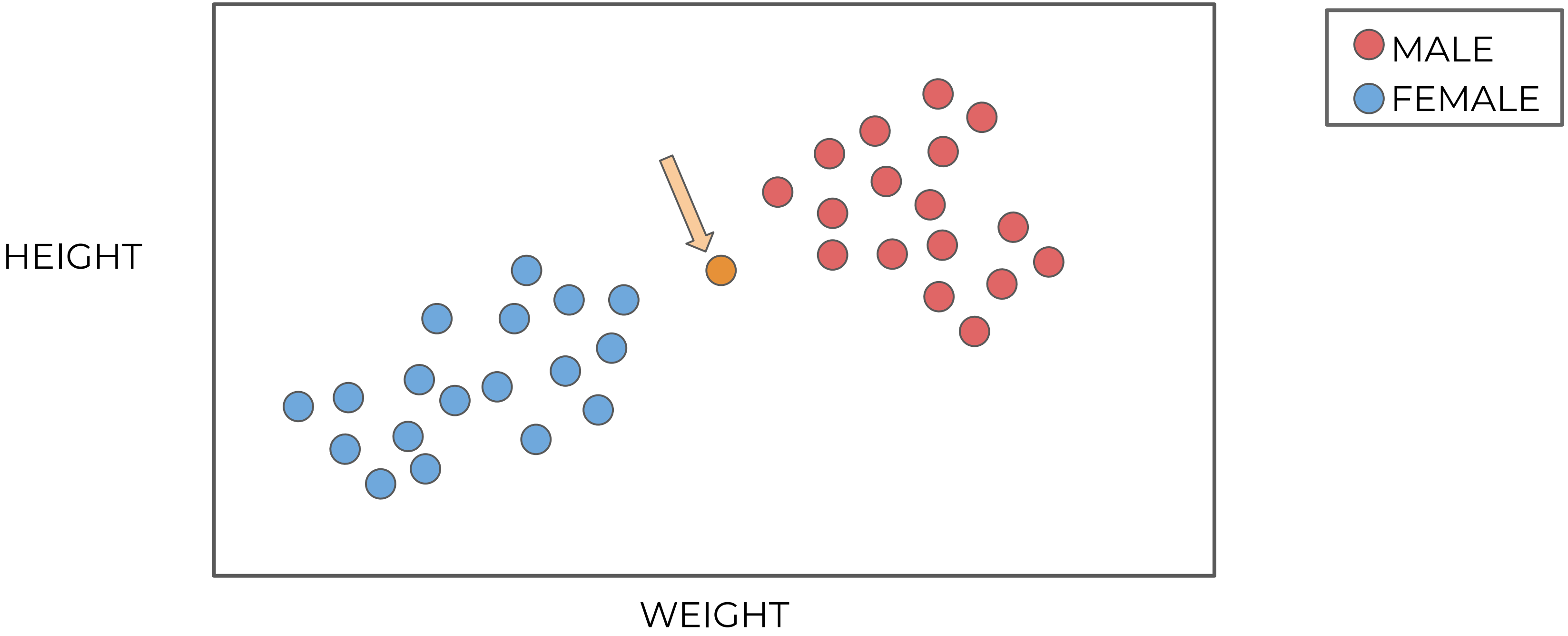
KNN

- What about a less obvious point?



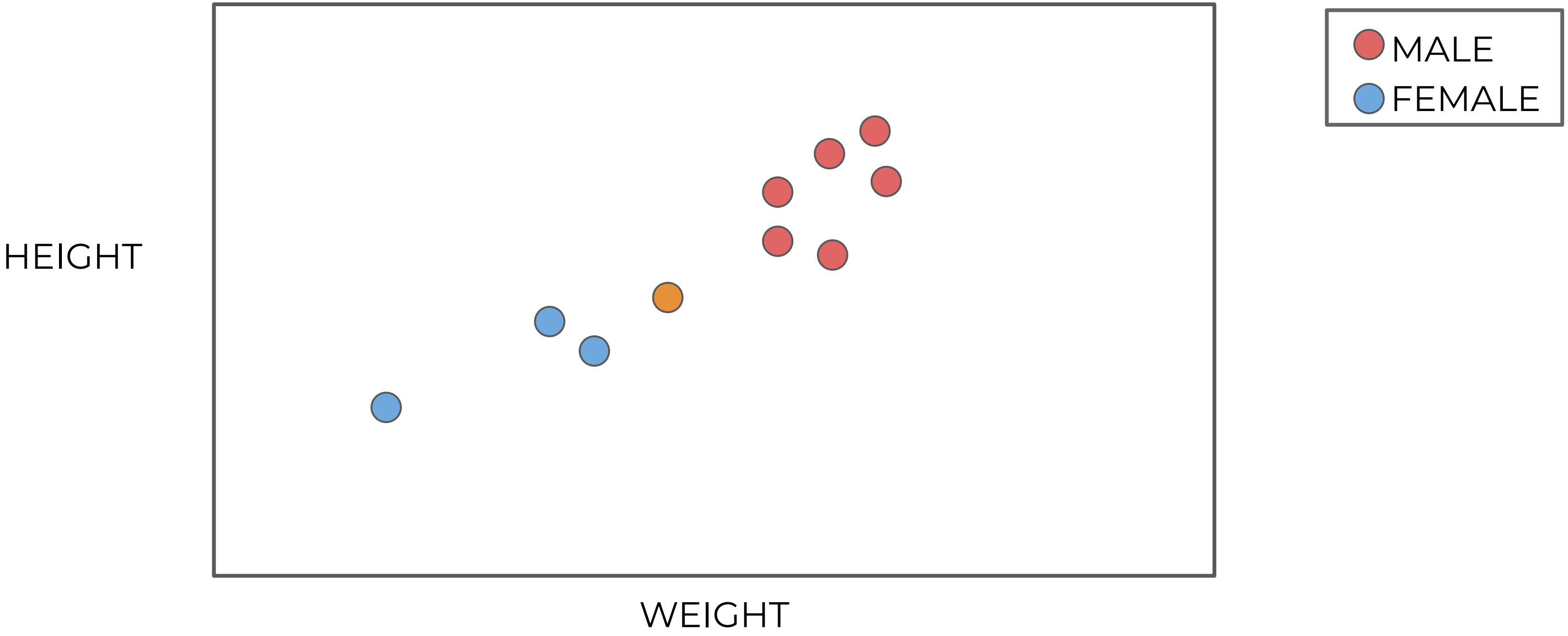
KNN

- How many points to we consider?



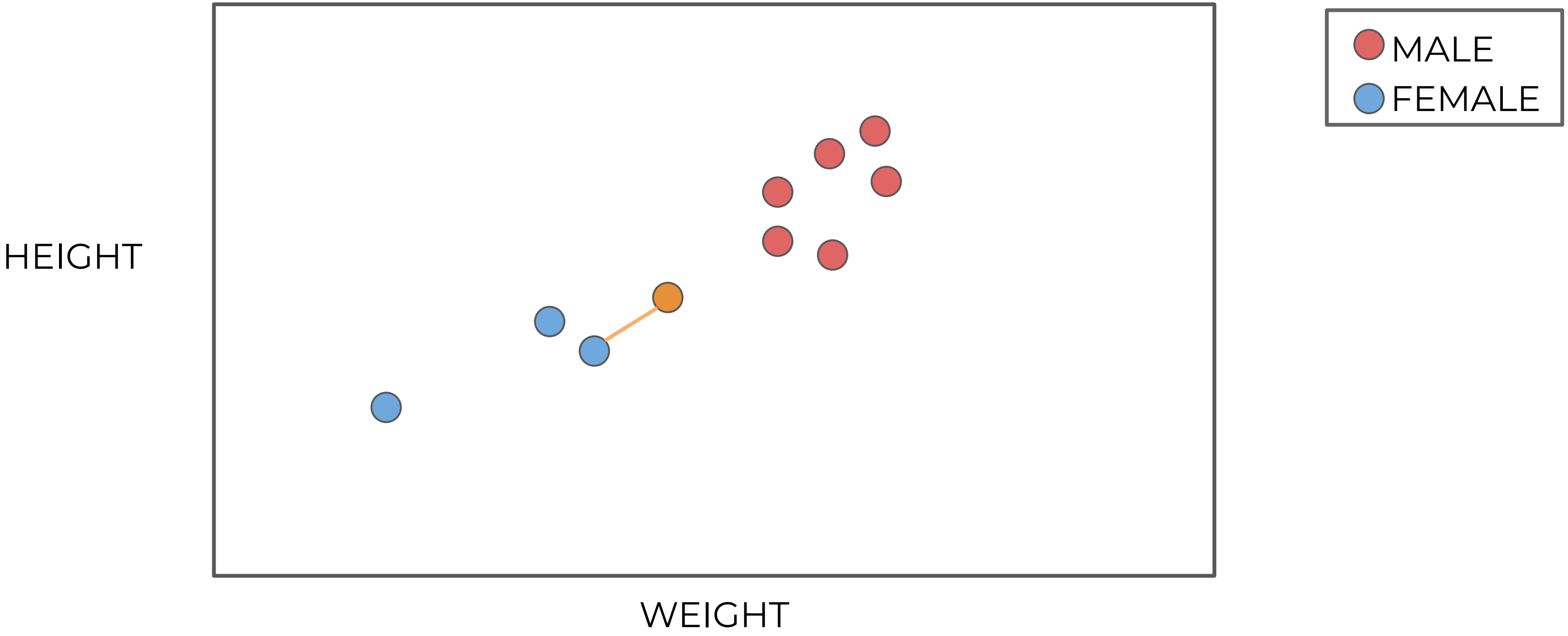
KNN

- Let’s imagine a situation like this:



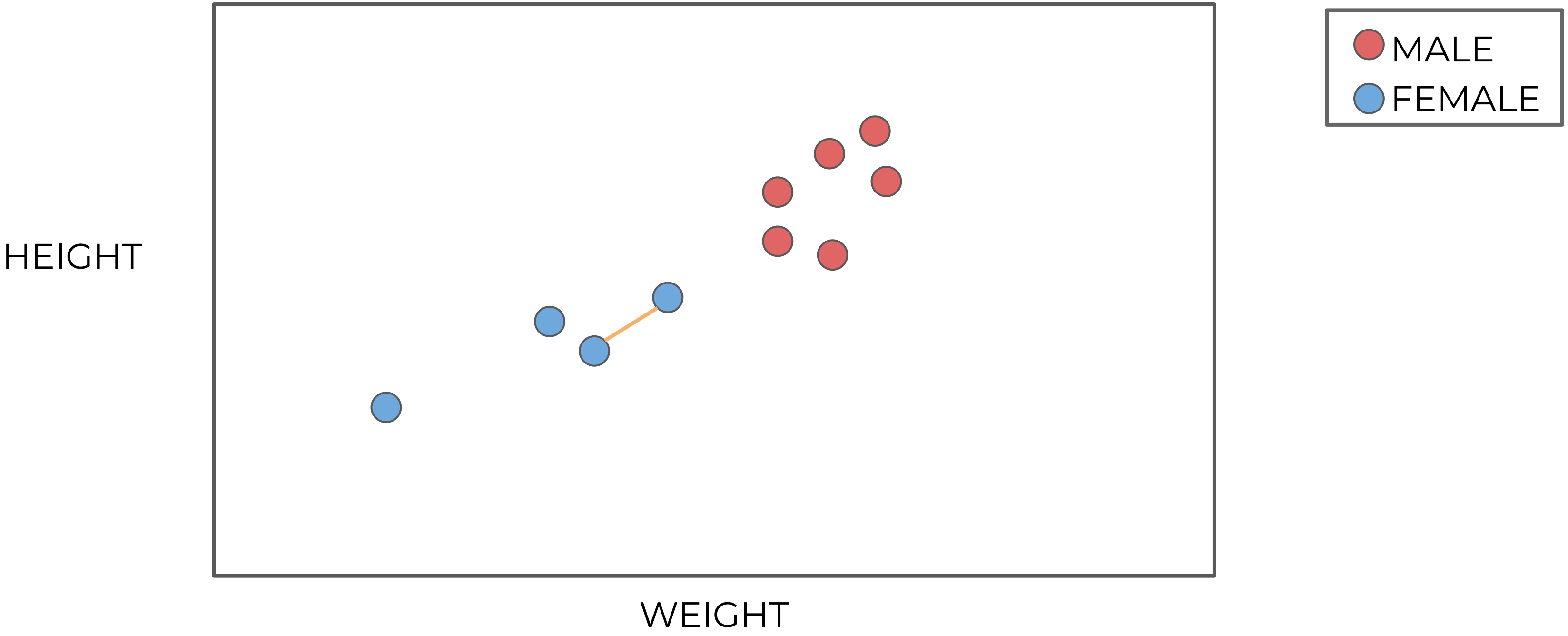
KNN

- K=1



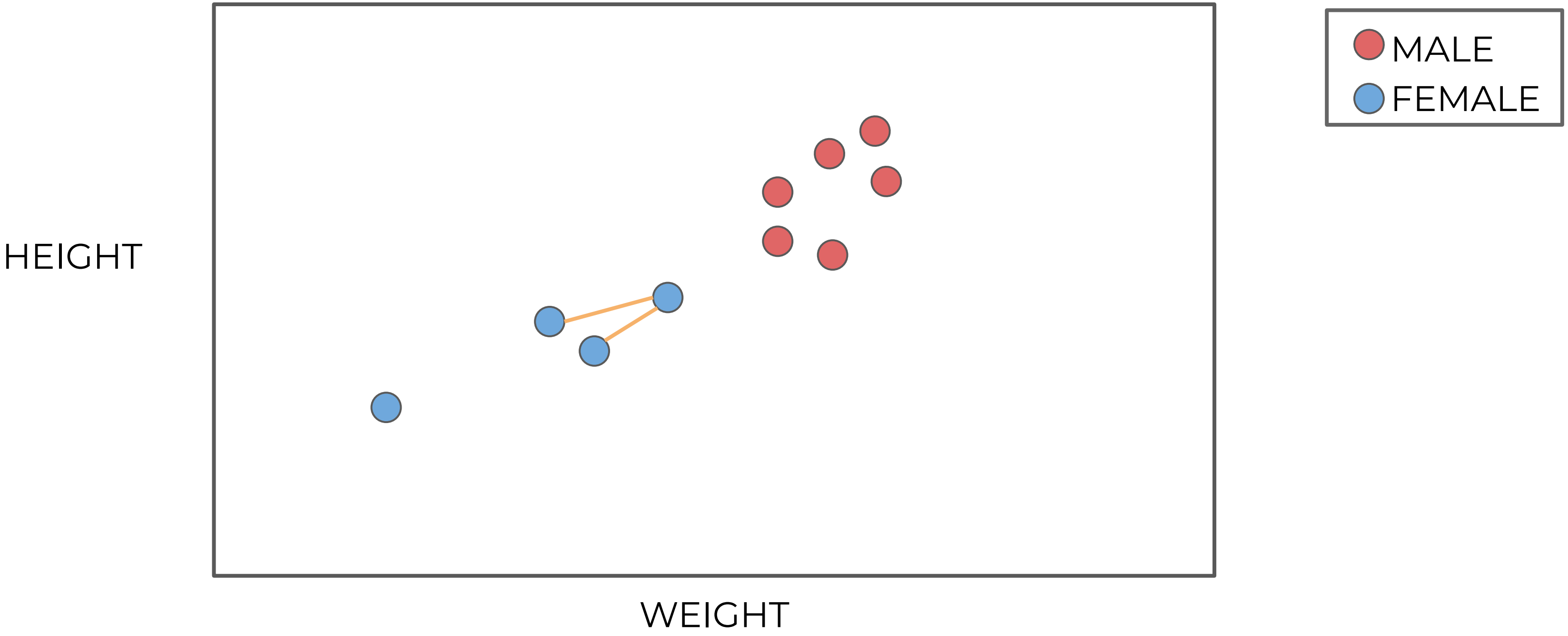
KNN

- K=1



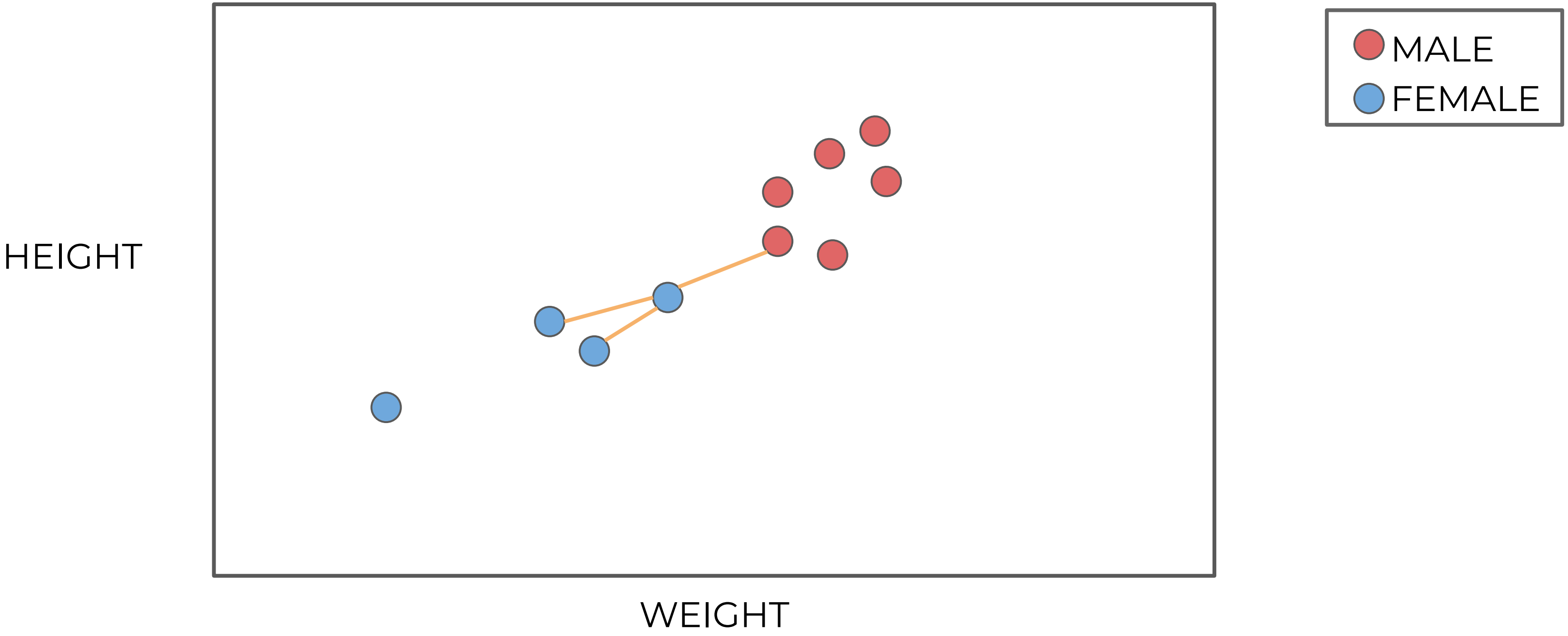
KNN

- K=2



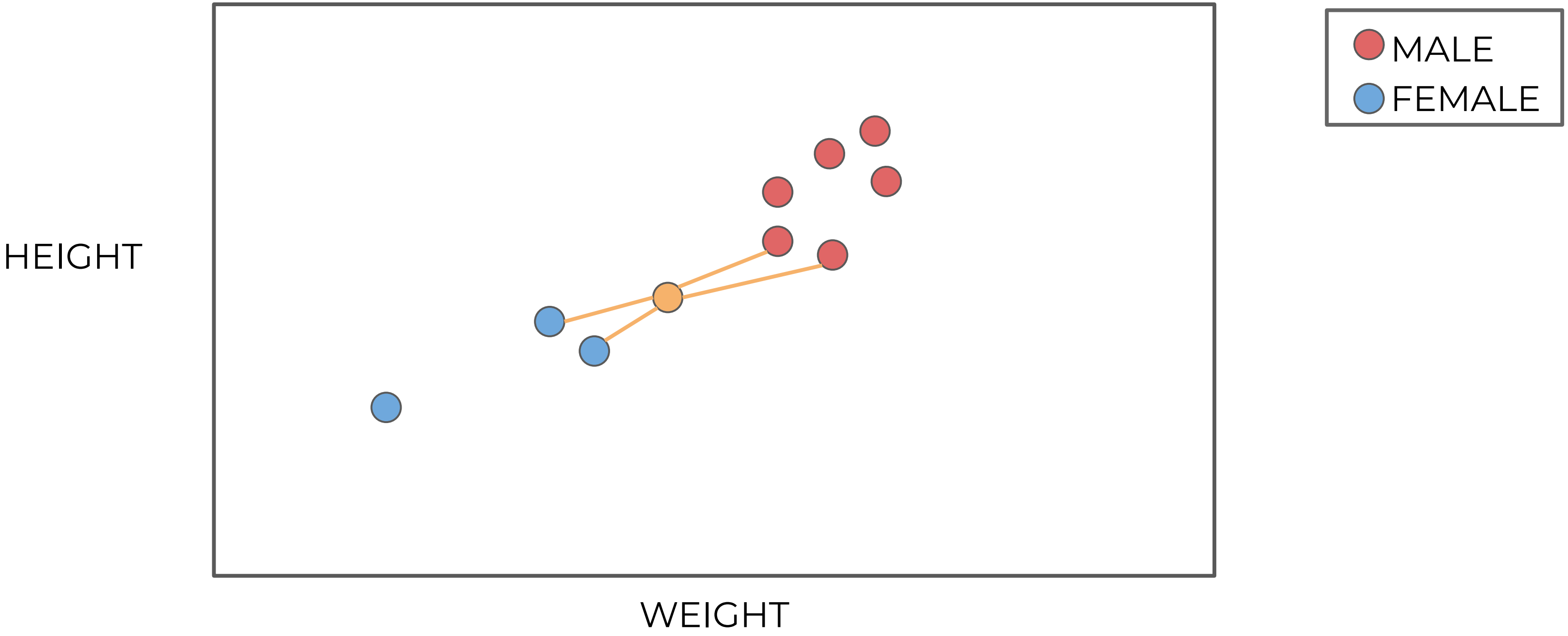
KNN

- K=3



KNN

- K=4 leads to a tie!



KNN

- 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.

KNN

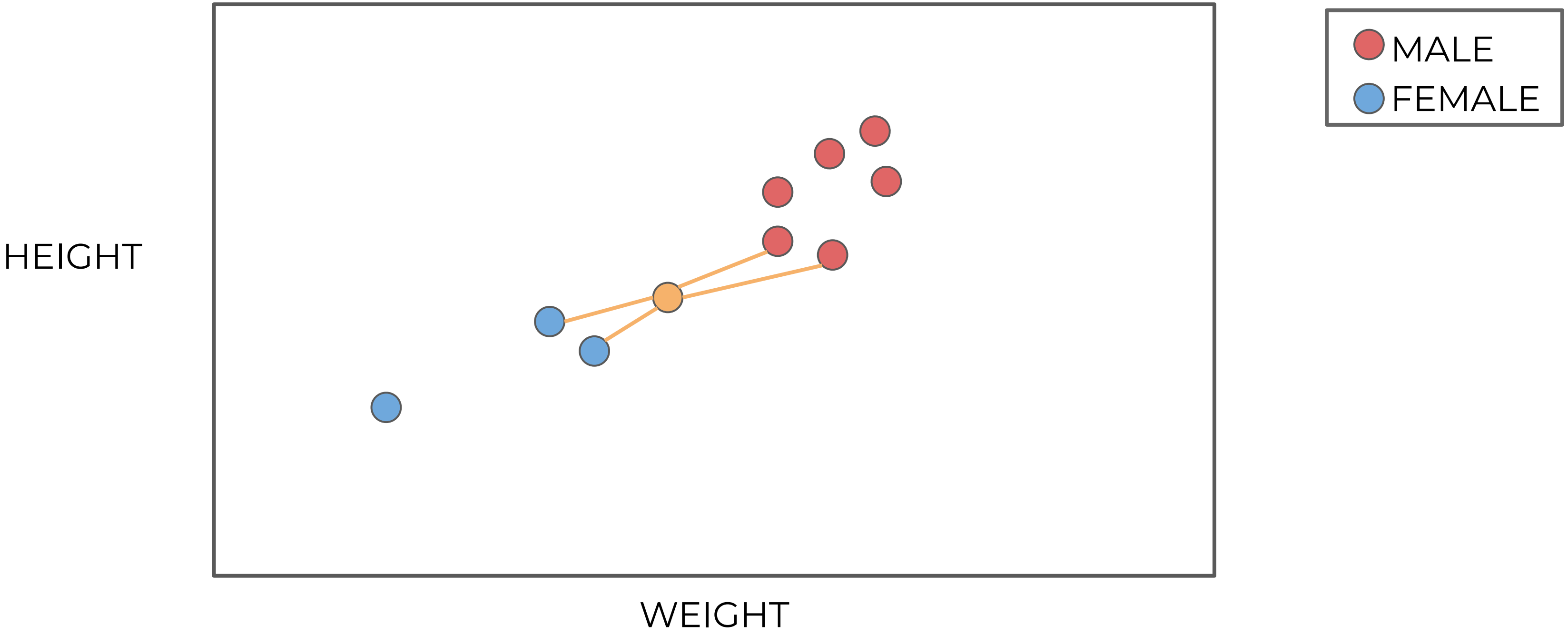
- What does Scikit-Learn do in case of tie?
  - *Warning: Regarding the Nearest Neighbors algorithms, if it is found that two neighbors, neighbor k+1 and k, have identical distances but different labels, the results will depend on the ordering of the training data.*

KNN

- 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.*

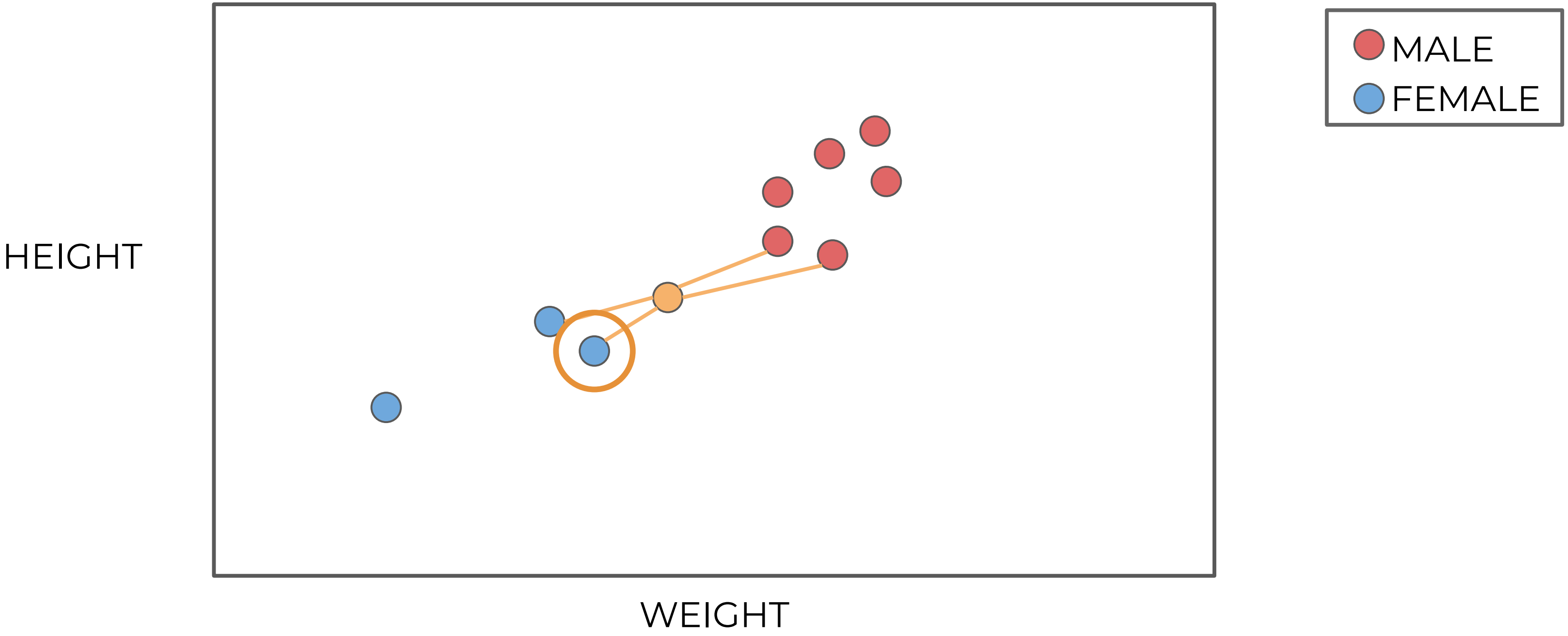
KNN

- K=4 leads to a tie!



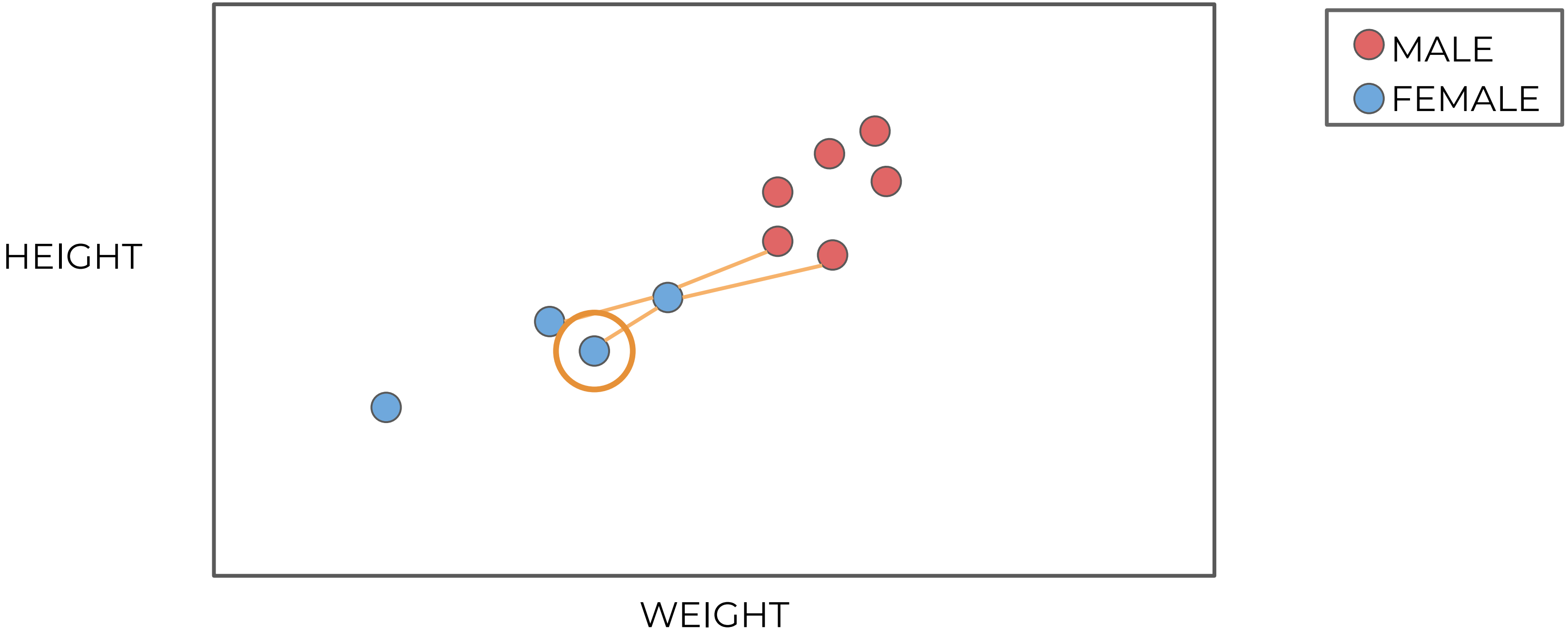
KNN

- Choose closest K



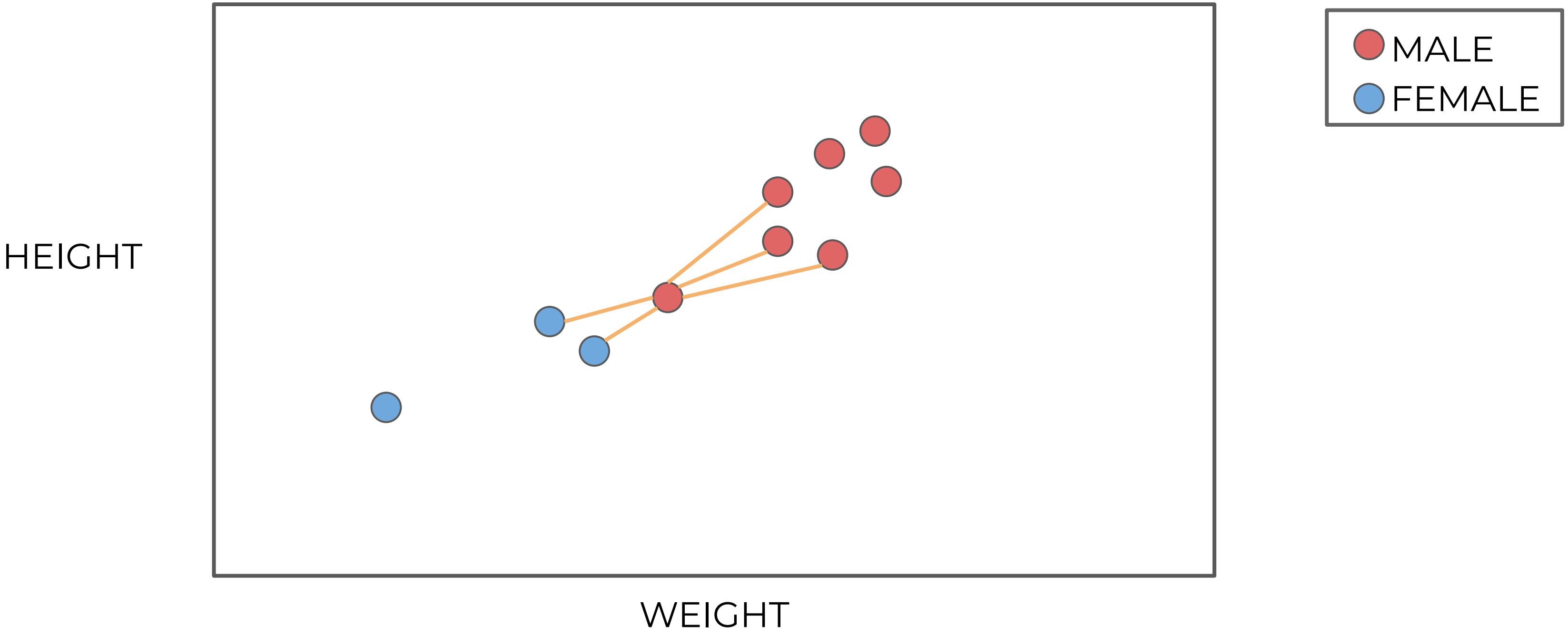
KNN

- Choose closest K



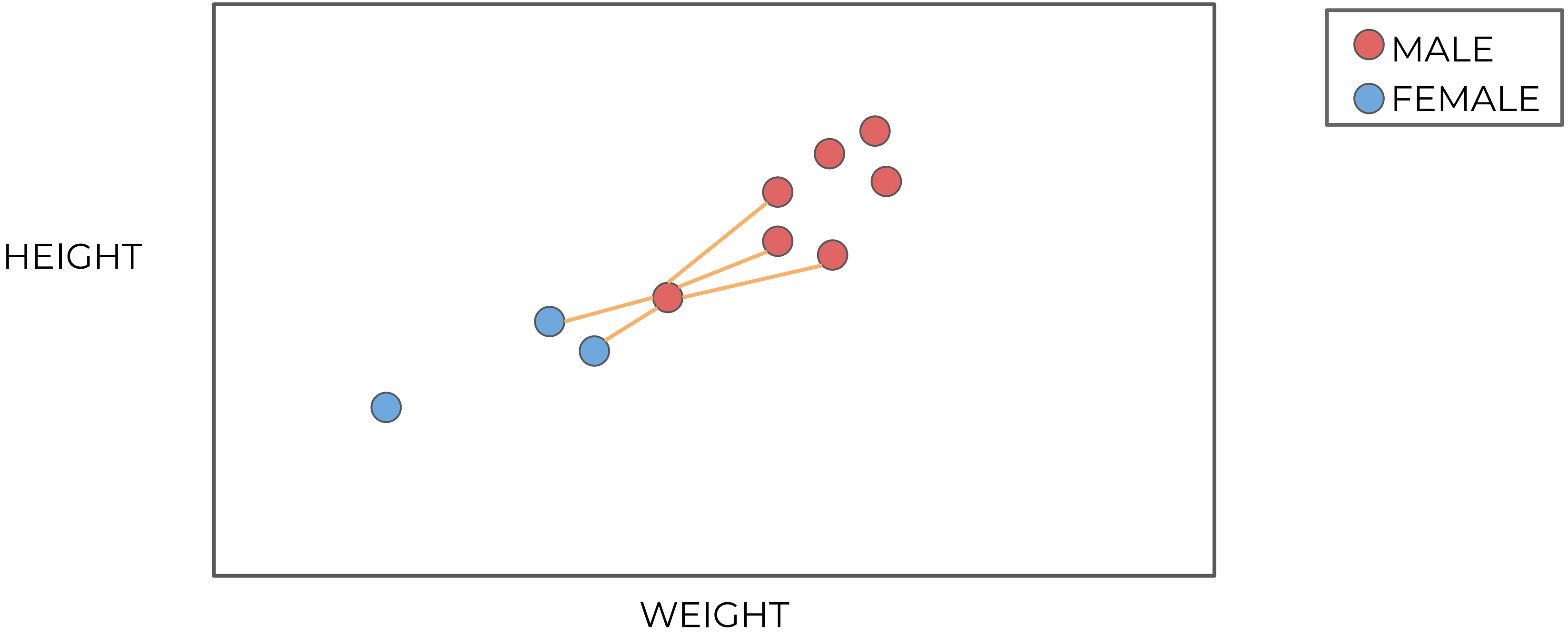
KNN

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



KNN

- How to choose best K value?

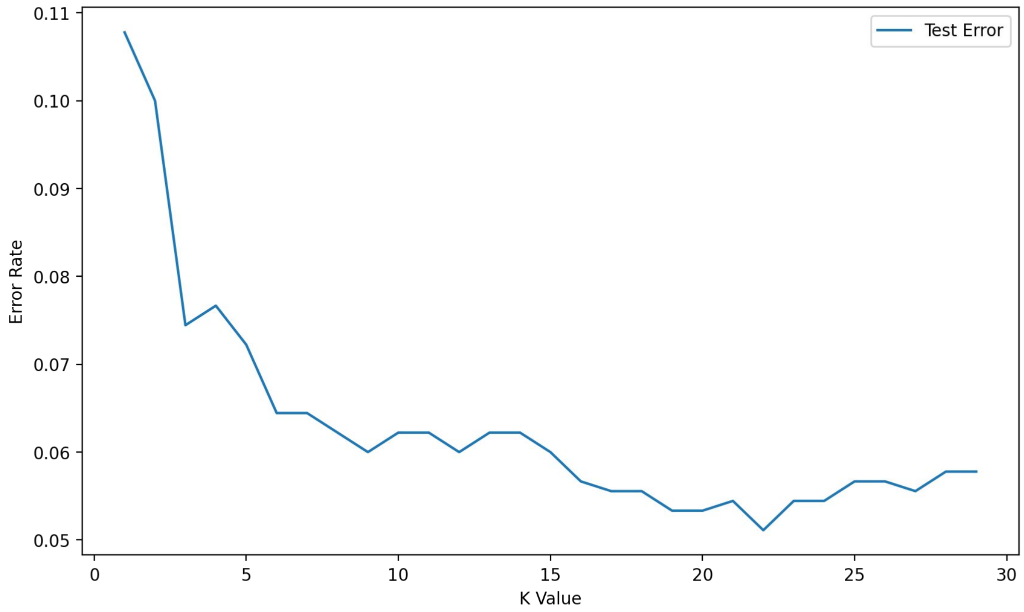


KNN

- We want a K value that **minimizes** error:
  - Error = 1 - 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.

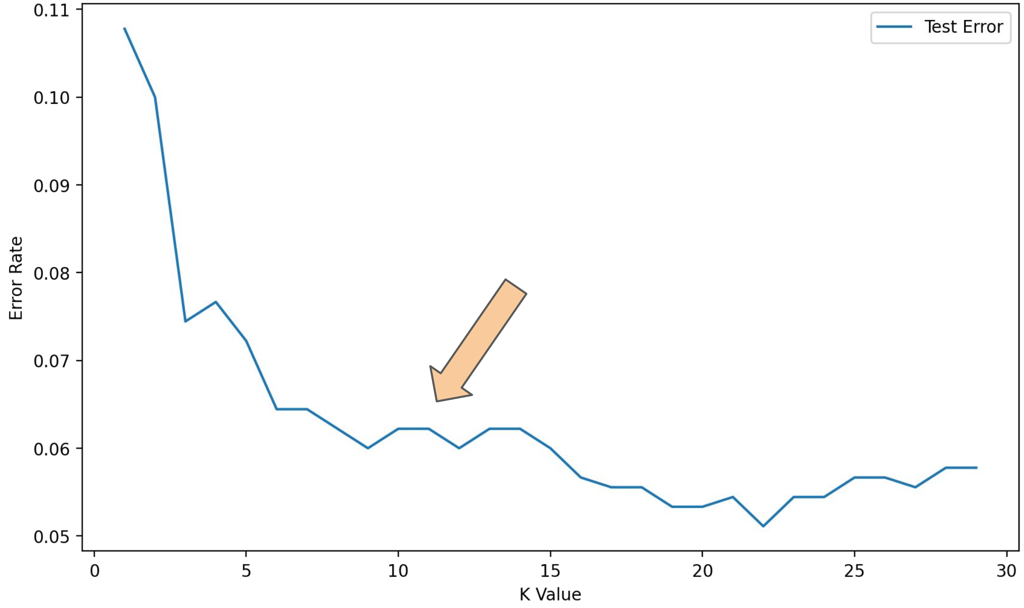
KNN

- Elbow method:



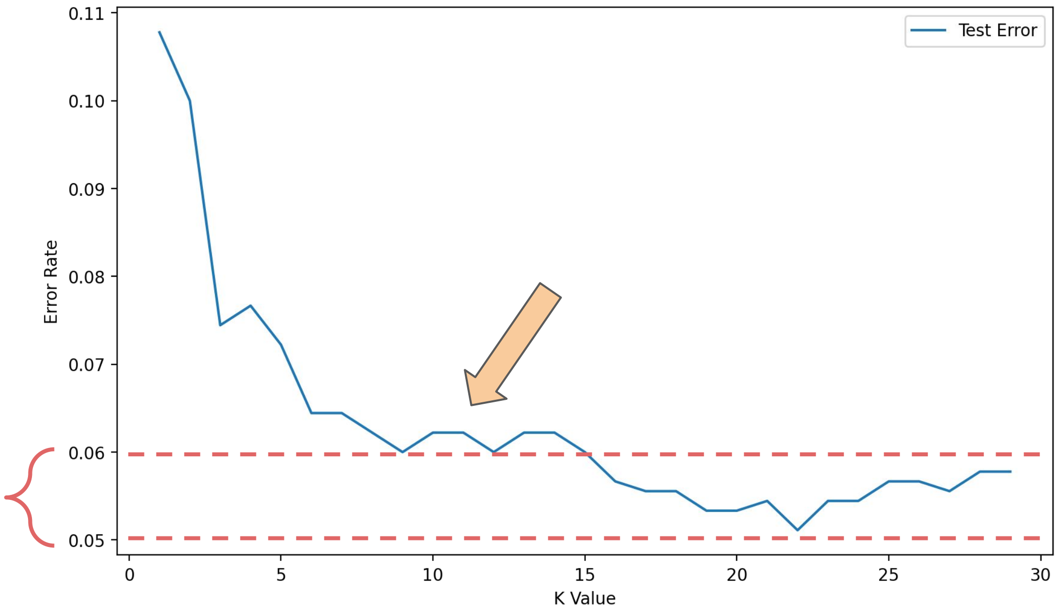
KNN

- Elbow method:



KNN

- Elbow method:



KNN

- 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

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

KNN

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

KNN

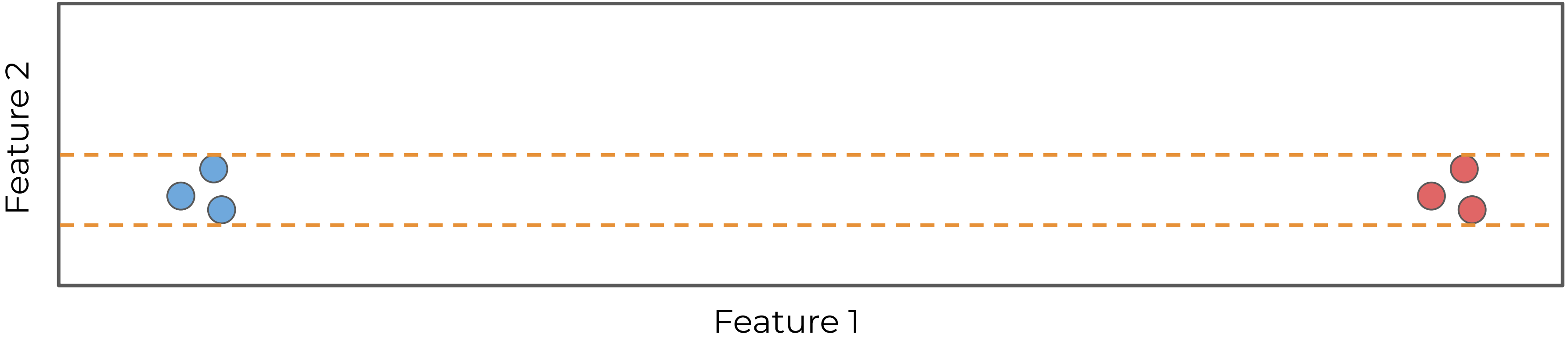
- KNN Considerations:
    - Scaling for Distance
        - Features could have vastly different value ranges!

KNN

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

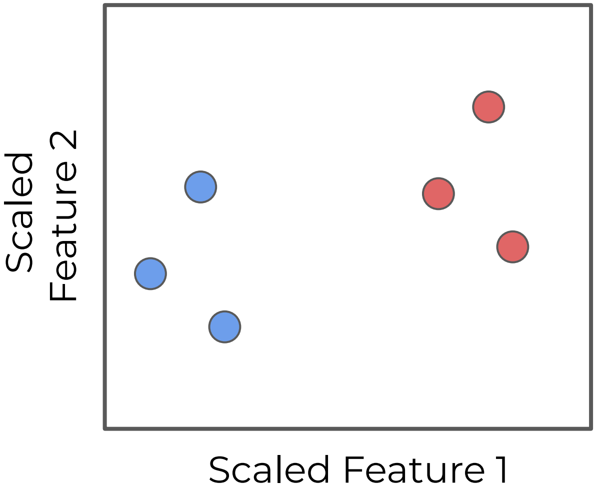
KNN

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



KNN

- KNN Considerations:
  - Scaling is necessary for KNN.



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 Two: Choosing K

KNN

- 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.