Nearest Neighbours

Max Turgeon

DATA 2010–Tools and Techniques in Data Science

Lecture Objectives

- Explain the nearest neighbour algorithm.
- Fit nearest-neighbour classifiers in Python.
- Evaluate the model using different metrics.

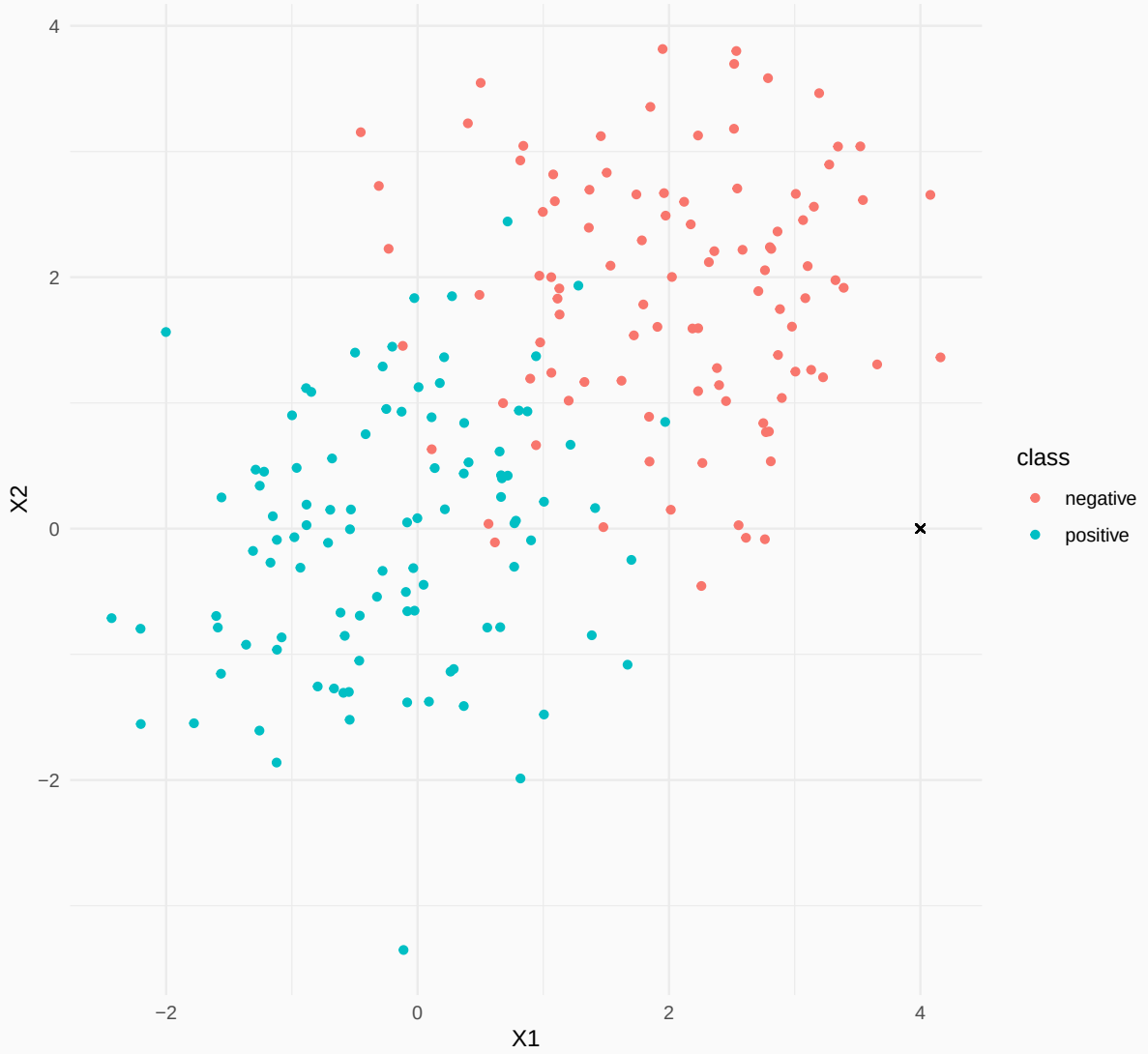
Motivation

- We started our discussion of machine learning with linear and logistic regression.
- We discussed how flexible they are:
    - Combined with splines.
    - Using regularization.
- Nonetheless, they still assume linearity and additivity (i.e. the regression equation).
- The next approaches we will discuss don't make any such assumption.

Nearest neighbours i

- Let's assume we want to classify a new observation based on training data.
- For visualization purposes, we will assume we have two covariates.

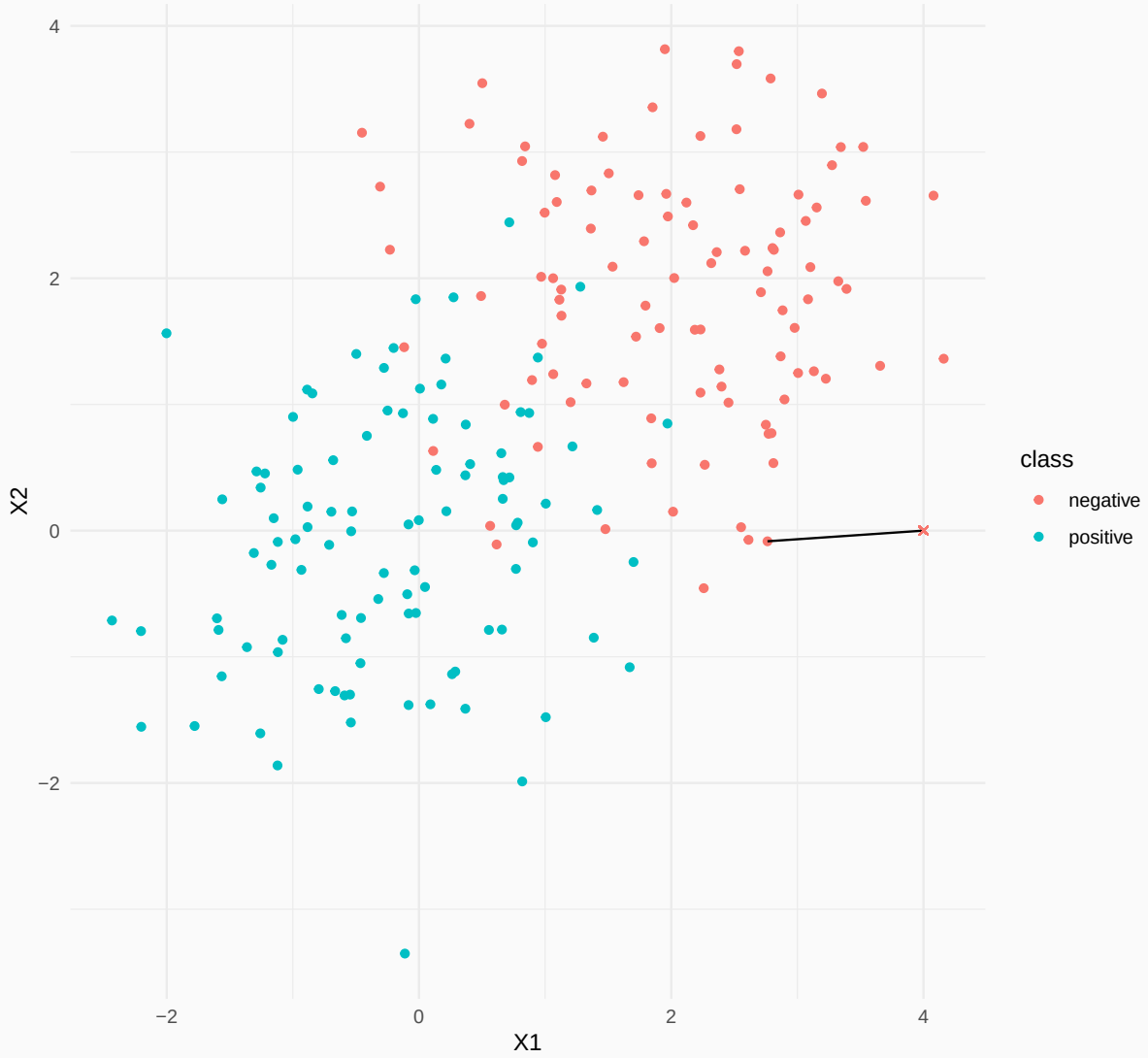
Nearest neighbours ii



Nearest neighbours iii

- How should we classify the new observation?
- The new observation is surrounded by negative observations, so it would make sense to classify it as negative.
- In fact, the *closest* training data point (in Euclidean distance) is negative.

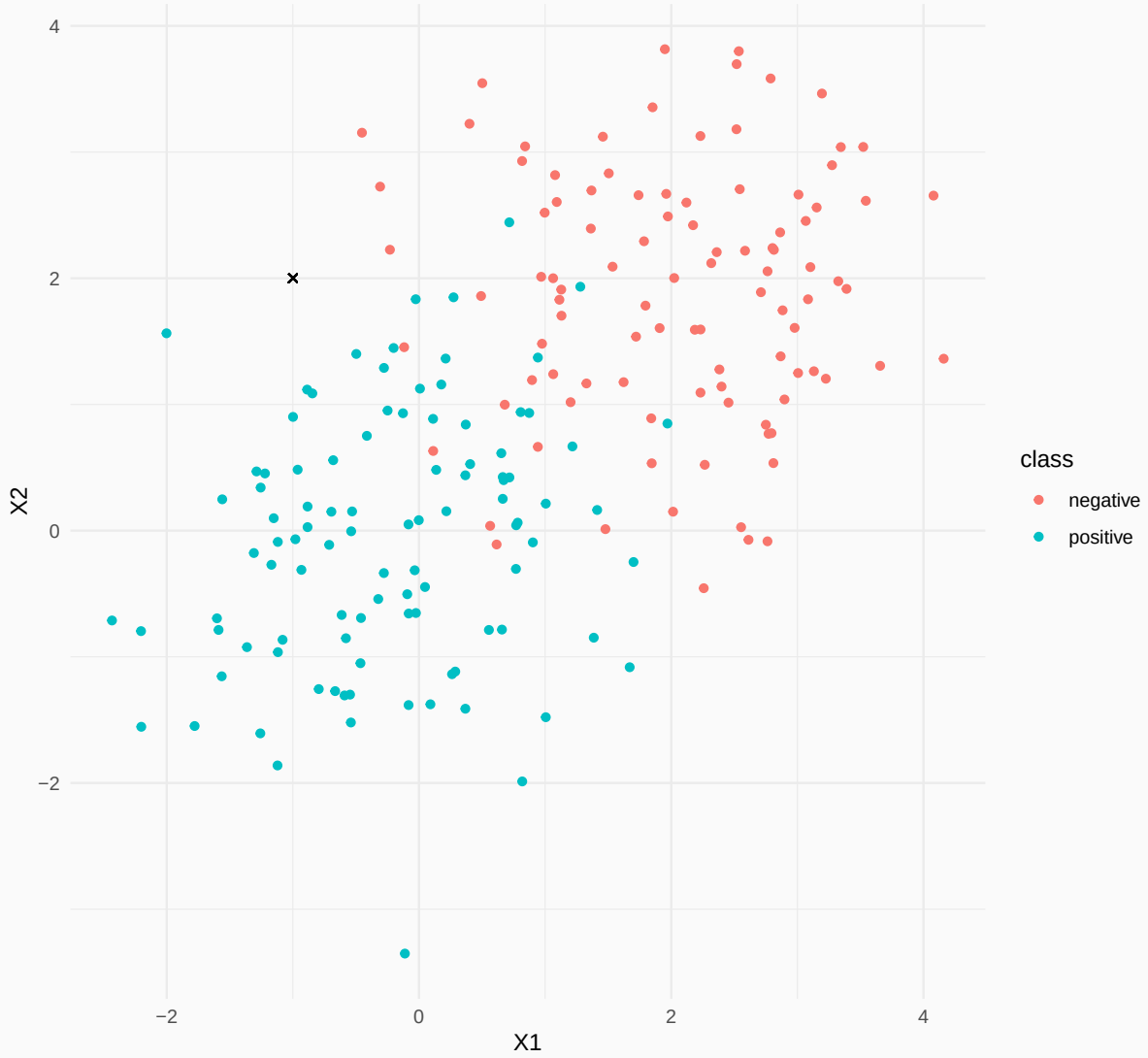
Nearest neighbours iv



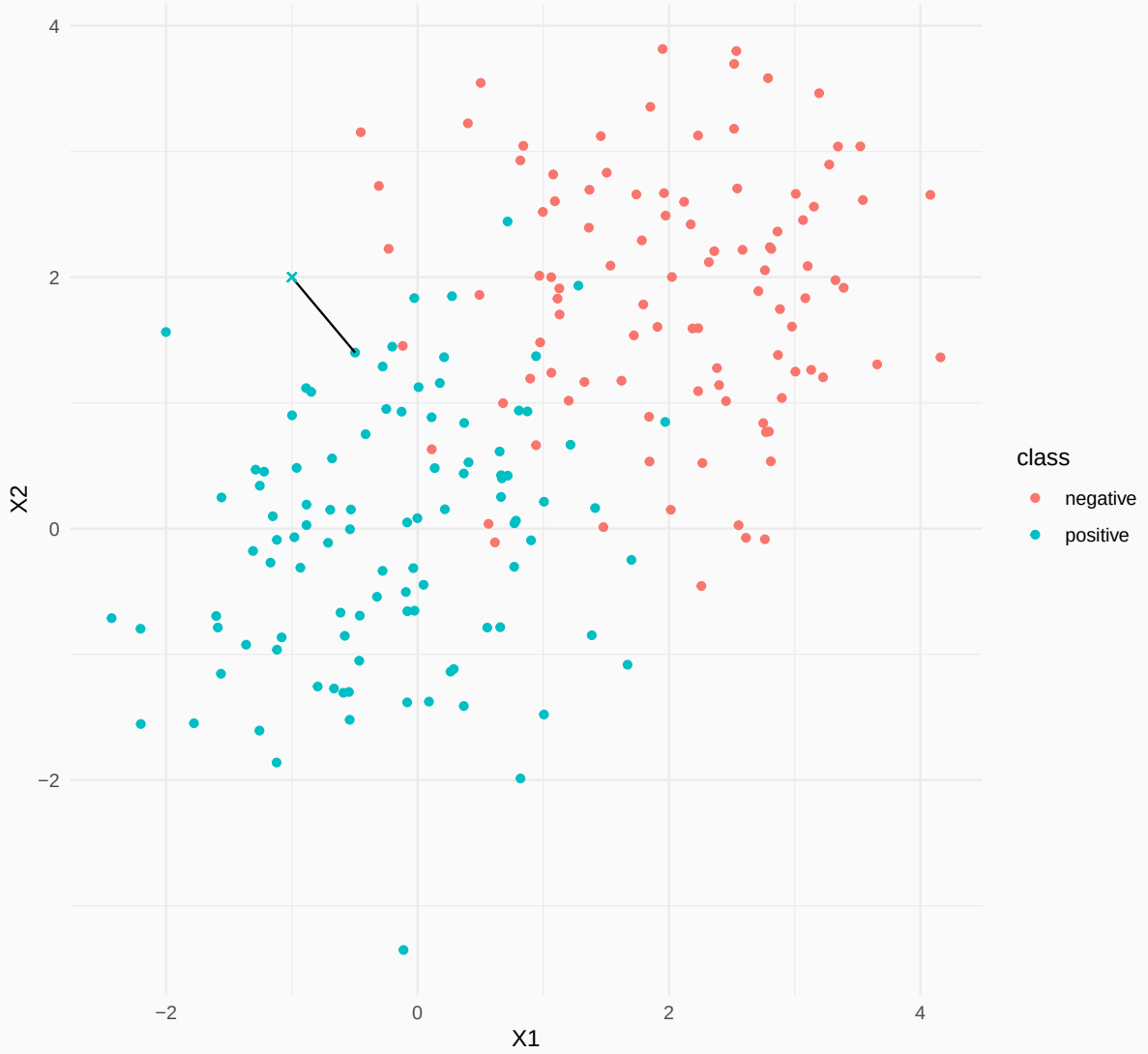
Nearest neighbours v

- What about an observation that is not surrounded by a single class?
- We can still find its nearest neighbour:
    - If it's negative, classify as negative.
    - If it's positive, classify as positive.

Nearest neighbours vi



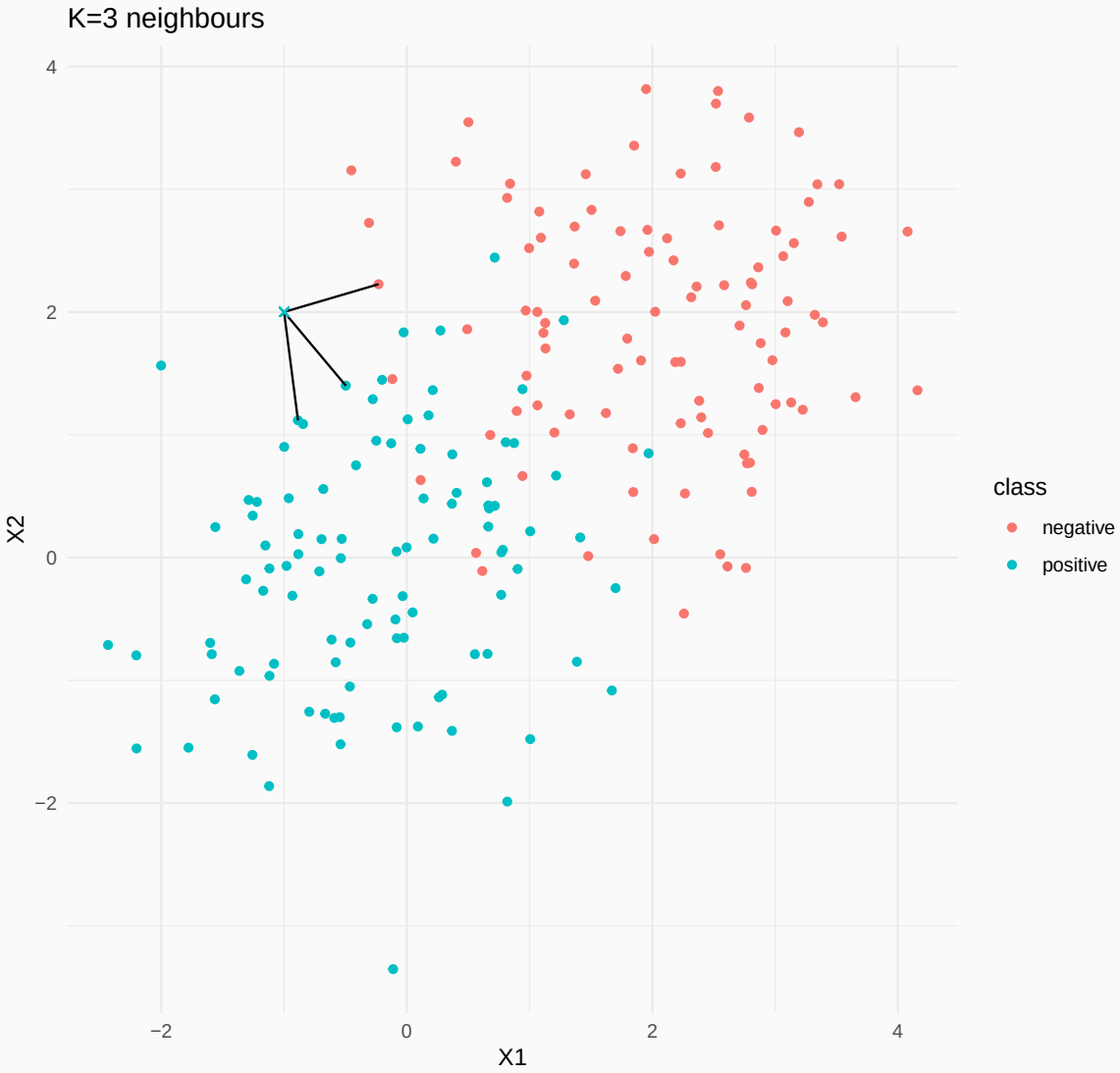
Nearest neighbours vii



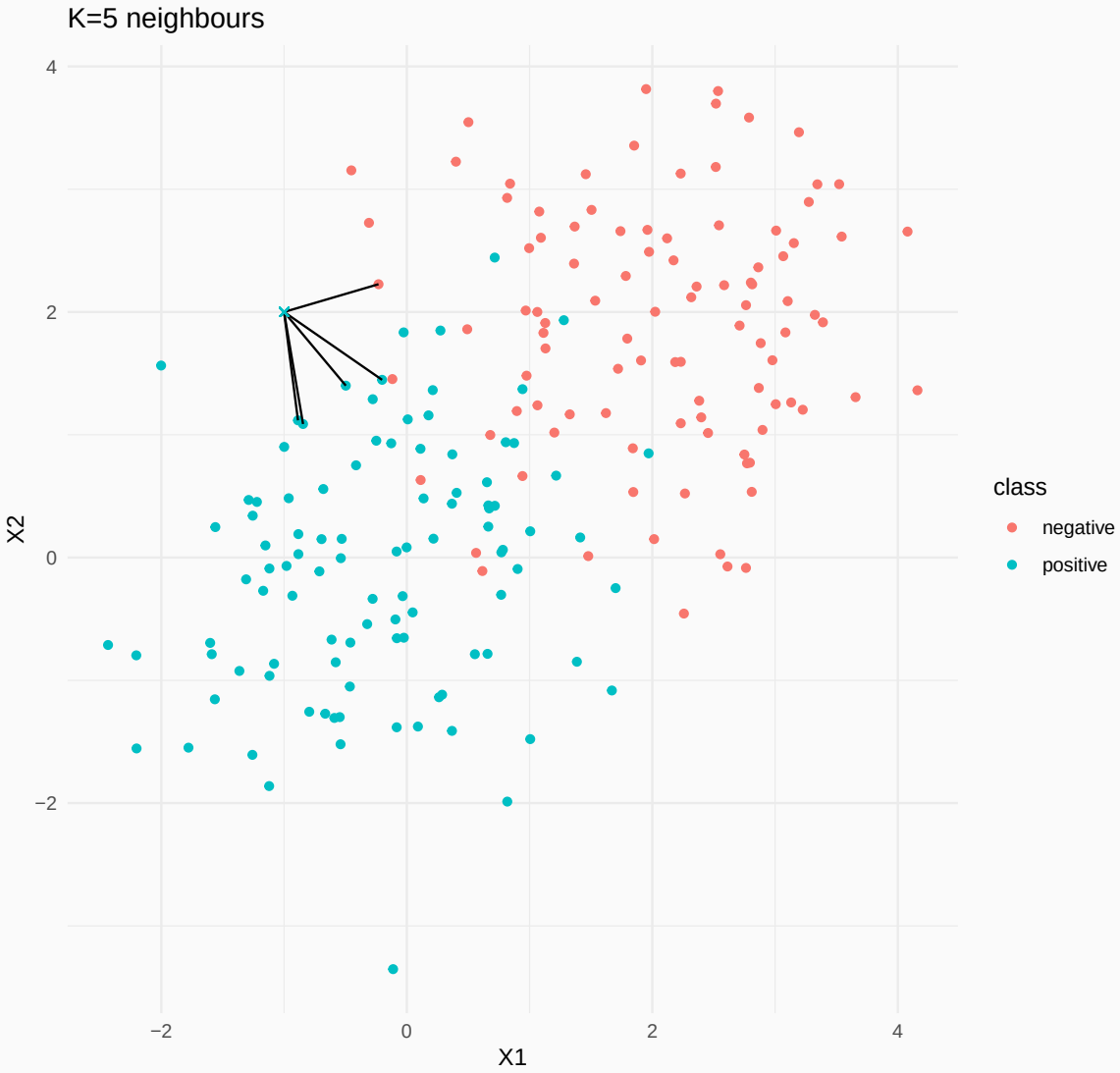
Nearest neighbours viii

- This is a simple algorithm, but there is one problem:
  - What if the nearest neighbour is an outlier?
- To increase robustness, we can instead look at multiple neighbours:
  - If a *majority* are negative, classify as negative.
  - If a *majority* are positive, classify as positive.

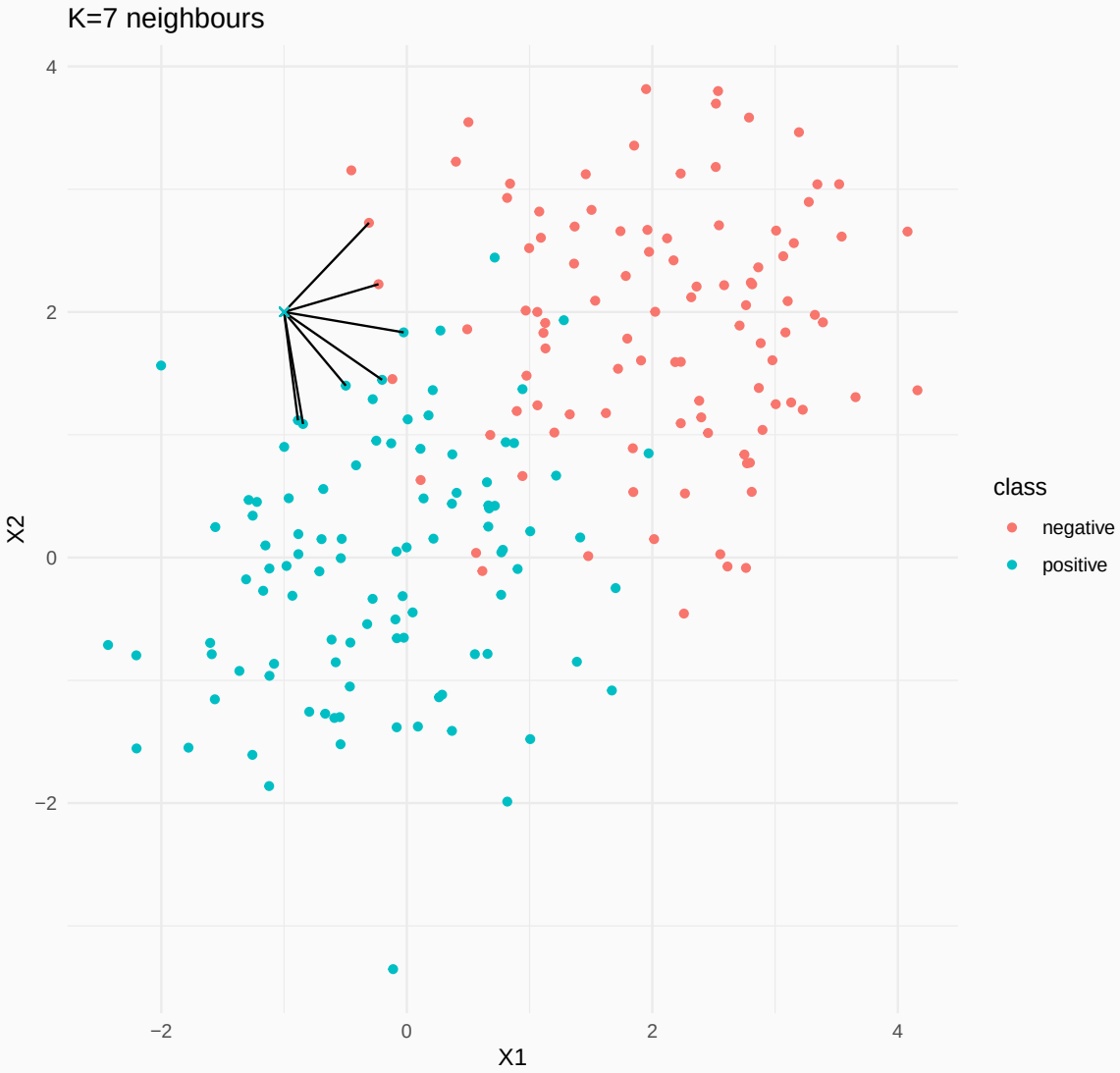
Nearest neighbours ix



Nearest neighbours x



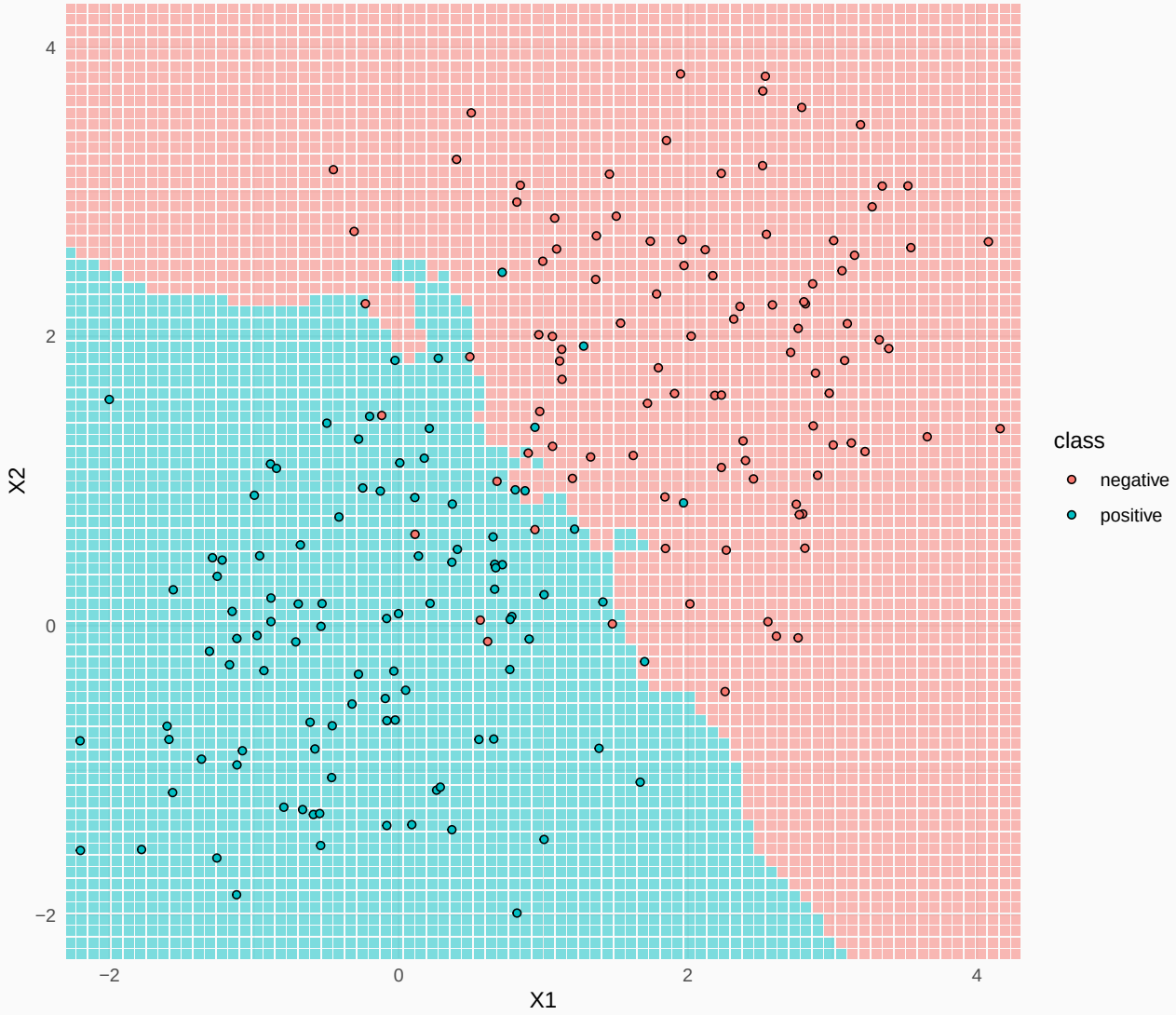
Nearest neighbours xi



Nearest neighbours xii

- **Question**: Why did I only choose odd values of K?
- We can visualize our *decision boundary* and see that it is not linear.
    - In fact it doesn't even have to create connected regions for each class.

Nearest neighbours xiii



Algorithm

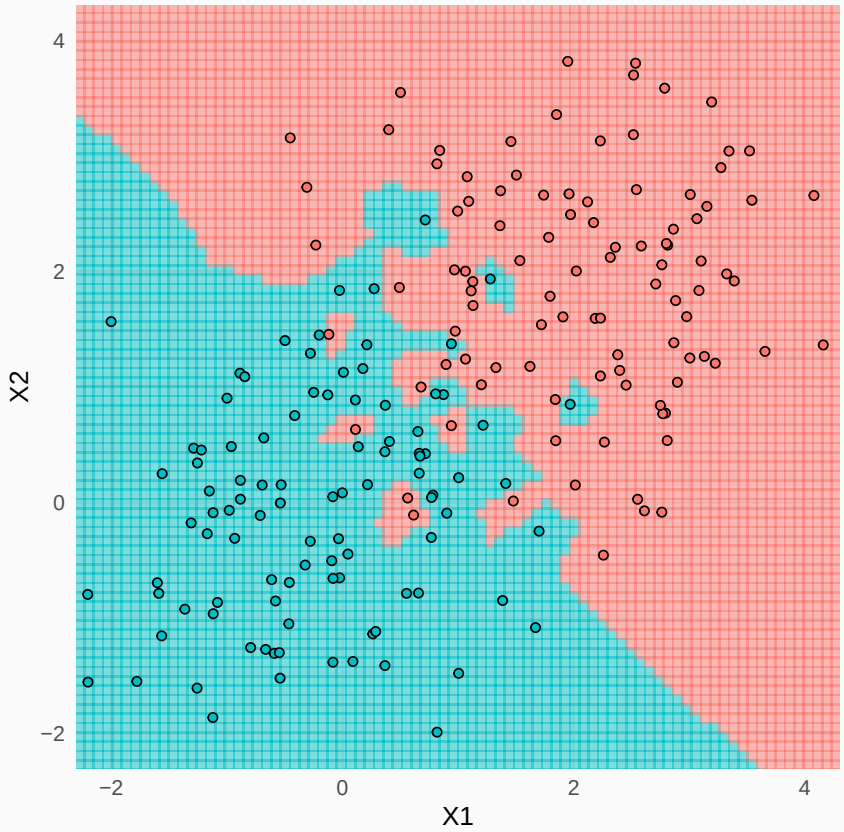
1. Fix a positive integer K.
2. For a new observation, compute its distance (using the predictors) to all the data points in the training data.
3. Find the K training data points with the smallest distance to the new observation.
4. Use the majority rule to classify the new observation.

Comments i

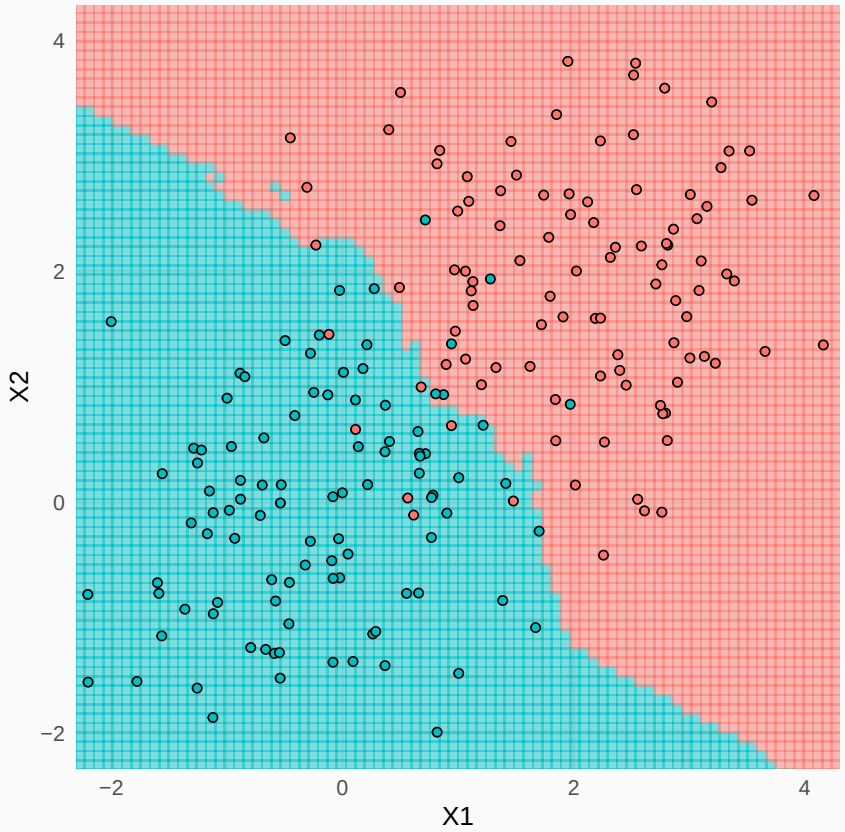
- To avoid ties in binary classification, we usually take K odd.
- We typically use the Euclidean distance for its simplicity, but any distance function would work.
- The optimal K depends on the data. Larger K reduces the impact of outliers, but it makes the decision boundary less distinct.
    - Bias-variance trade-off
- The same approach can be used for classification with more than 2 classes.

Comments ii

K=1



K=9



Exercise

Open the Jupyter notebook on nearest-neighbour classification and follow the instructions.

Nearest Neighbour Regression

- For classification, we found the nearest neighbours are used the majority class to make a prediction.
- For *regression*, we can take the mean (or median) of the neighbours.
- In this way, we can also create regression models using the same general approach.
- **Exercise**: Build a nearest-neighbour regression model for the prostate cancer dataset. Evaluate your model.

Summary

- The idea is simple: use the neighbours from the training data to inform the classification.
- There is no value of K that will work best all the time.
    - It's a hyper-parameter than can be tuned.
- Nearest neighbour can perform poorly with unbalanced training data.
    - *Solution*: Downsample or upsample to create a balanced dataset (out of scope for this course).
- Next lecture, we will start our discussion of classification and regression trees.