Machine learning fundamentals

Cian Scannell

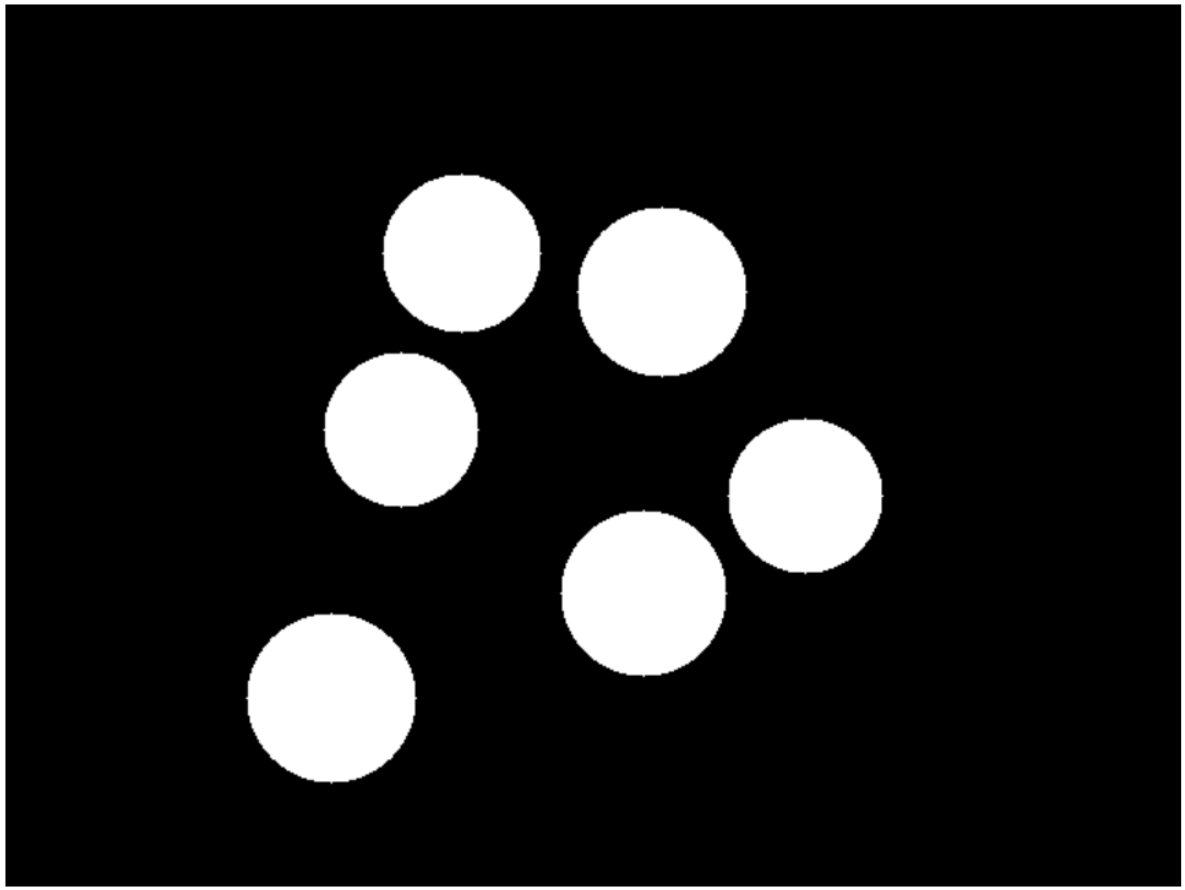
Eindhoven University of Technology
Department of Biomedical Engineering

2024

Intended Learning Outcomes

- Introduce machine learning
- Distinguish between *supervised* and *unsupervised* machine learning
- Distinguish between *parametric* and *non-parametric* models
- Describe an intuitive method for *classification* (k-NN)
- Introduce the concept of *generalisation* and *complexity* with respect to machine learning models

How would you approach this problem?

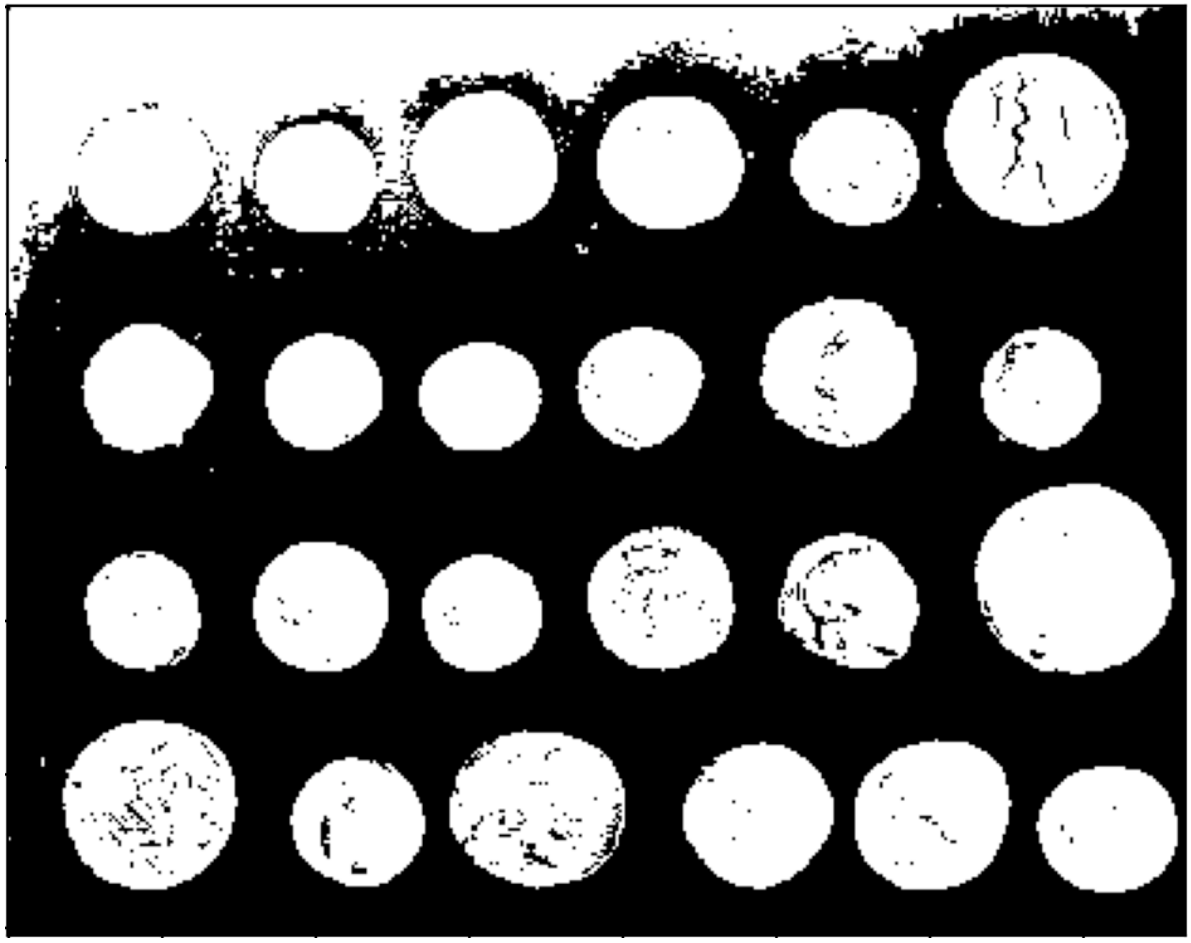


Count and measure all circles in the image.

How would you approach this problem?

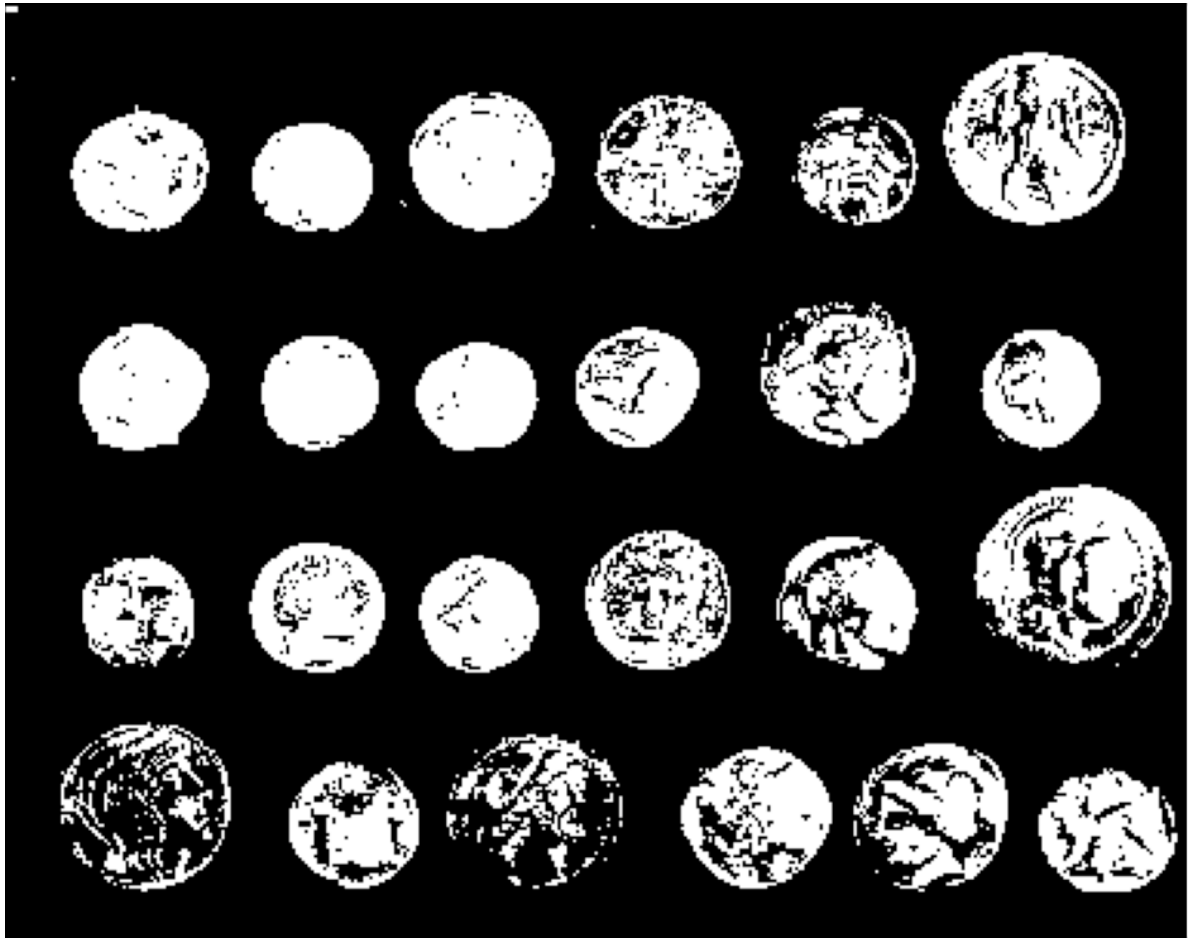
Count and measure all coins in the image.

How would you approach this problem?



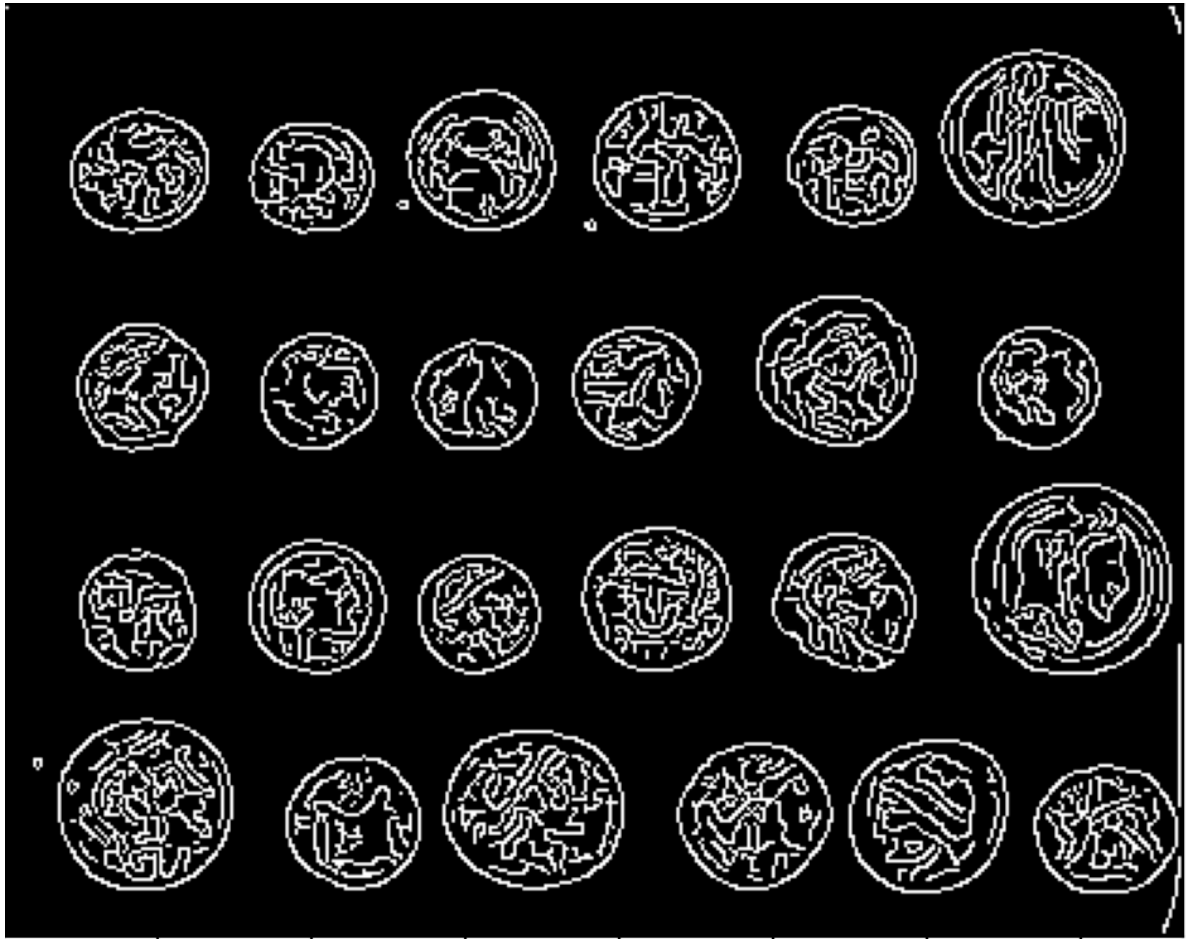
$threshold(image, 100)$

How would you approach this problem?



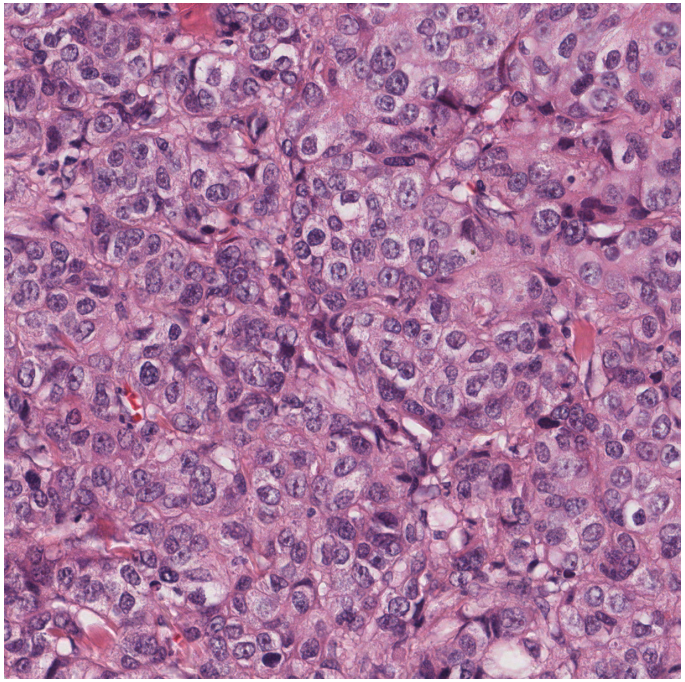
$threshold(image, 140)$

How would you approach this problem?



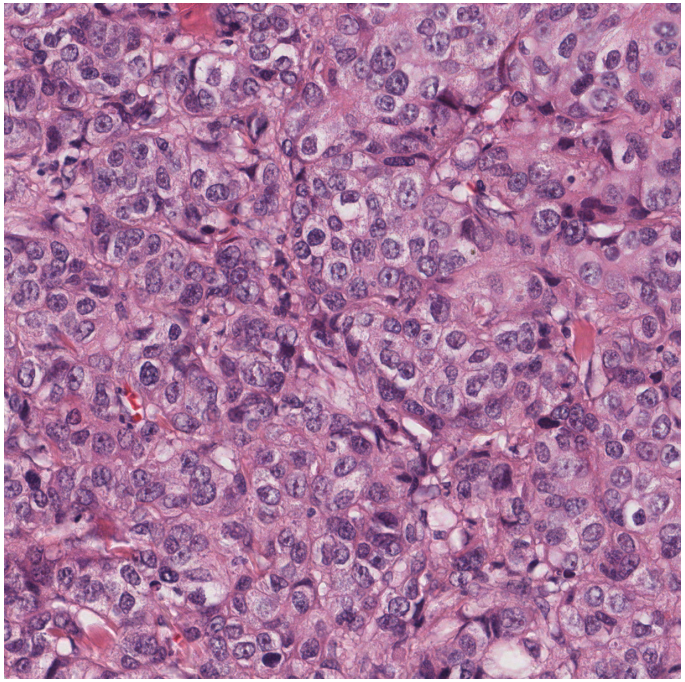
canny_edge_detection(*image*)

How would you approach this problem?



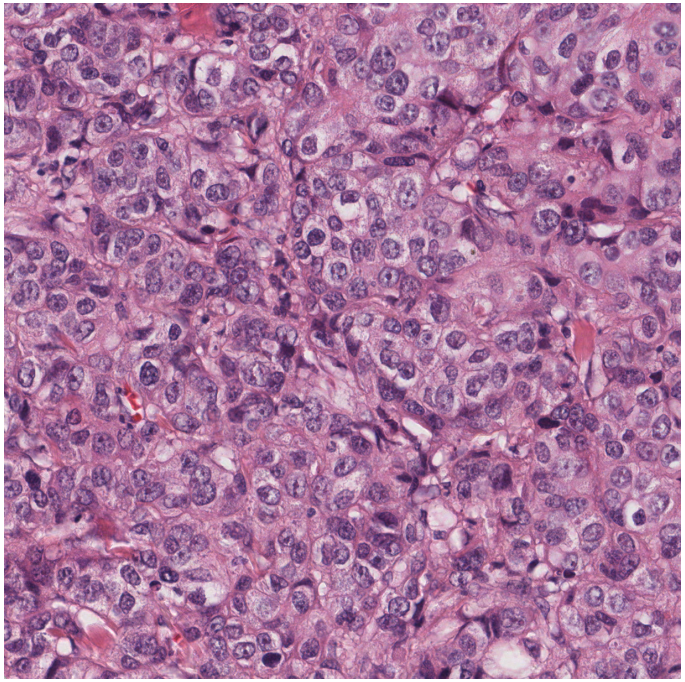
- Detect, segment and classify all cell nuclei.

How would you approach this problem?



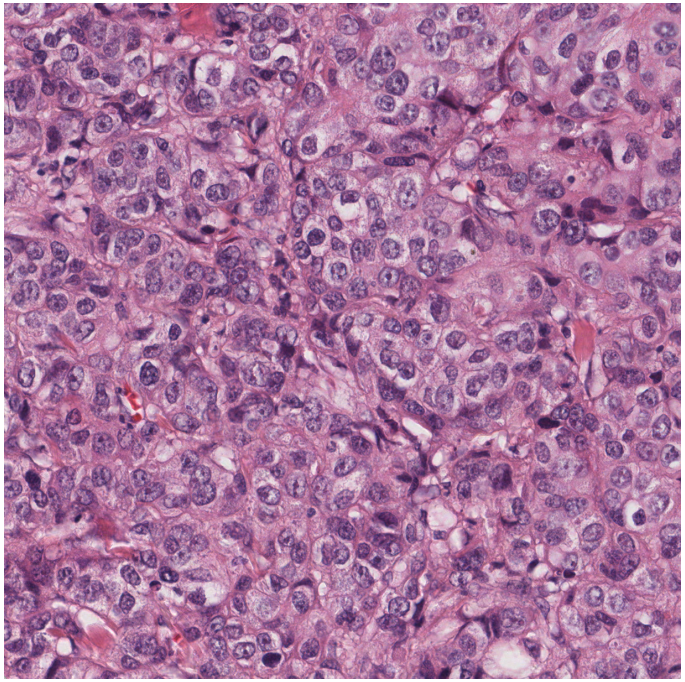
- Detect, segment and classify all cell nuclei.
- Classify as normal, benign or malignant.

How would you approach this problem?



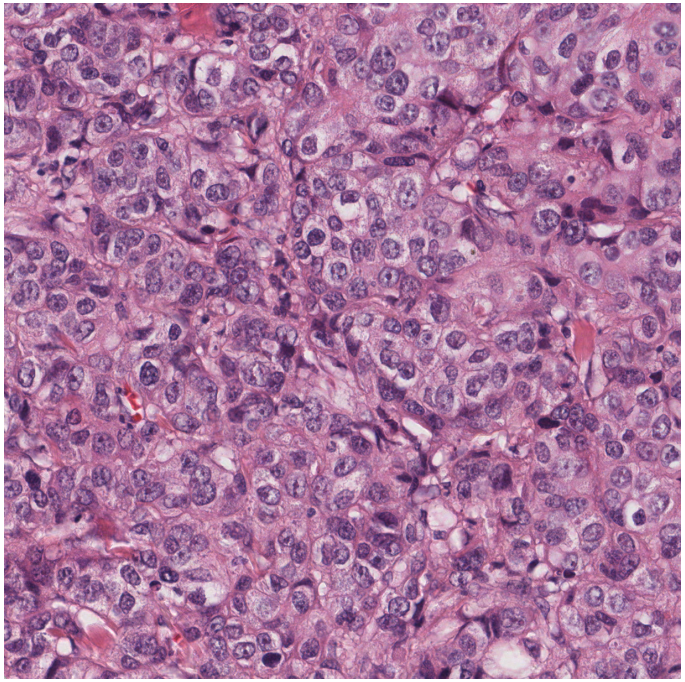
- Detect, segment and classify all cell nuclei.
- Classify as normal, benign or malignant.
- Classify as low, intermediate or high grade cancer.

How would you approach this problem?



- Detect, segment and classify all cell nuclei.
- Classify as normal, benign or malignant.
- Classify as low, intermediate or high grade cancer.
- Predict the 5-year disease free survival of this patient.

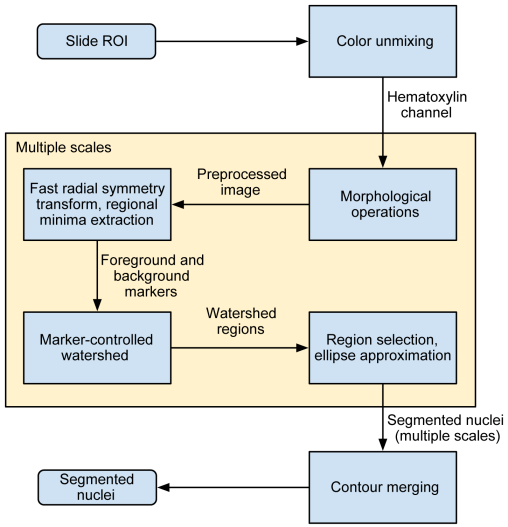
How would you approach this problem?

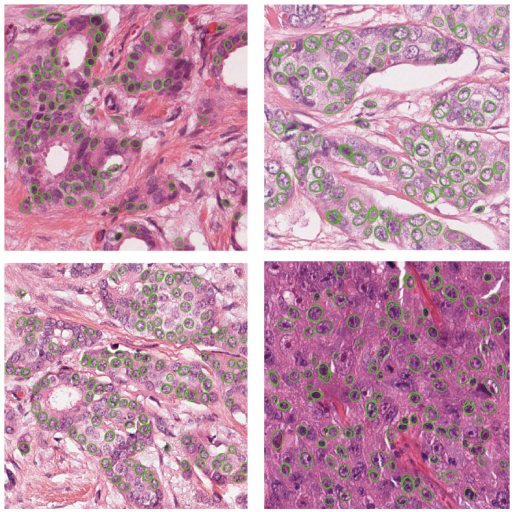


- Detect, segment and classify all cell nuclei.
- Classify as normal, benign or malignant.
- Classify as low, intermediate or high grade cancer.
- Predict the 5-year disease free survival of this patient.
- Predict if this patient will respond to a specific treatment.

An example from Mitko's past work: nuclei area measurement

2010-2011: An image processing pipeline of (mainly) mathematical morphology operators (e.g. the watershed algorithm).



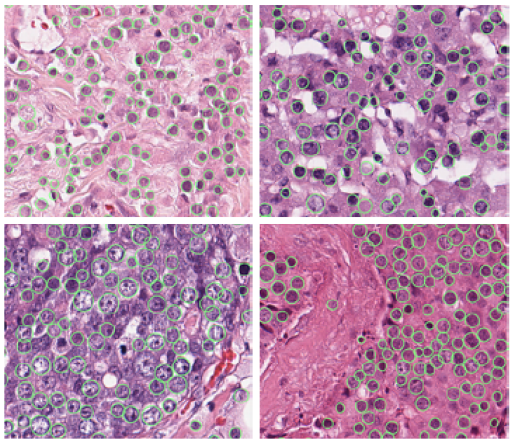


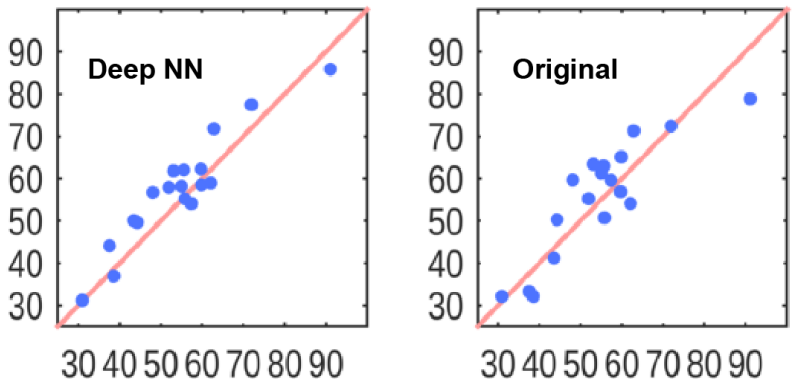
The design and validation of the processing pipeline took the better part of a year.

Figure source: Veta et al. PLOS ONE 2012

An example from Mitko's past work: nuclei area measurement

2015: A deep neural network for nuclei area measurement.





The the training and validation of the deep neural network model took less than a week.

The results were more accurate than the the original method.

Figure source: Veta et al. MICCAI 2016

An example from Mitko's past work: nuclei area measurement

In the first case, he translated the domain knowledge of (medical) experts about nuclei appearance into a series of **manually written rules** that perform nuclei segmentation.

In the second case, he took a dataset of nuclei segmentations and fed it to a (deep) machine learning algorithm that **learned** how to directly measure nuclei size **from the provided examples**.

The central premise of machine learning

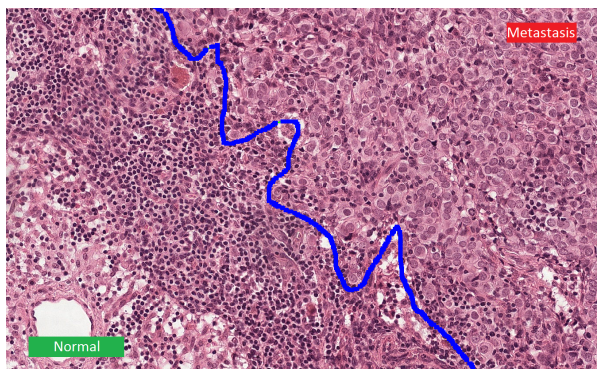
Learn "computer programs" from data instead of manually writing rules.

The central premise of machine learning

Learn "computer programs" from data instead of manually writing rules.

Advantage: the same method (e.g. a neural network) can be used to solve a variety of different problems.

Siberian hustky vs. eskimo dog



Normal vs. metastases

Figures source: (left) Szegedy et al. arXiv 2014, (right) camelyon16.grand-challenge.org

Types of machine learning

- Unsupervised machine learning: given a dataset x_i, find "some interesting properties".
  - Clustering: find groupings of x_i
  - Density estimation: find $p(x_i)$
  - Generative models.
  - ...
- Supervised machine learning: given a training dataset $\{x_i, y_i\}$, predict $\hat{y}_i$ of previously unseen samples.
  - Regression: the target variables y_i are continuous.
  - Classification: the target variables y_i are categorical.
  - ...
- ...

Supervised learning

- Outcome measurement Y (also called dependent variable, response, target)
- Vector of p predictor measurements $X = (X_1, X_2, \ldots, X_p)^T$ (also called inputs, regressors, covariates, features, independent variables).
- In the regression problem, Y is quantitative (e.g price, blood pressure).
- In the classification problem, Y takes values in a finite, unordered set (survived/died, digit 0-9, cancer class of tissue sample).
- We have training data $(x_1, y_1), \ldots, (x_N, y_N)$. These are pairs of observations (also referred to as examples or instances).

Objectives

Machine learning aims to 'learn' a model f that predicts the outcome Y given the input X:

$$Y = f(X) + \epsilon$$

where ϵ captures measurement errors and other discrepancies.

With a good f we can make predictions of Y for new inputs $X = x$.

Objectives

On the basis of the training data we would like to:

- Accurately predict unseen test cases.
- Understand which inputs affect the outcome, and how.
- Assess the quality of our predictions.

In summary...

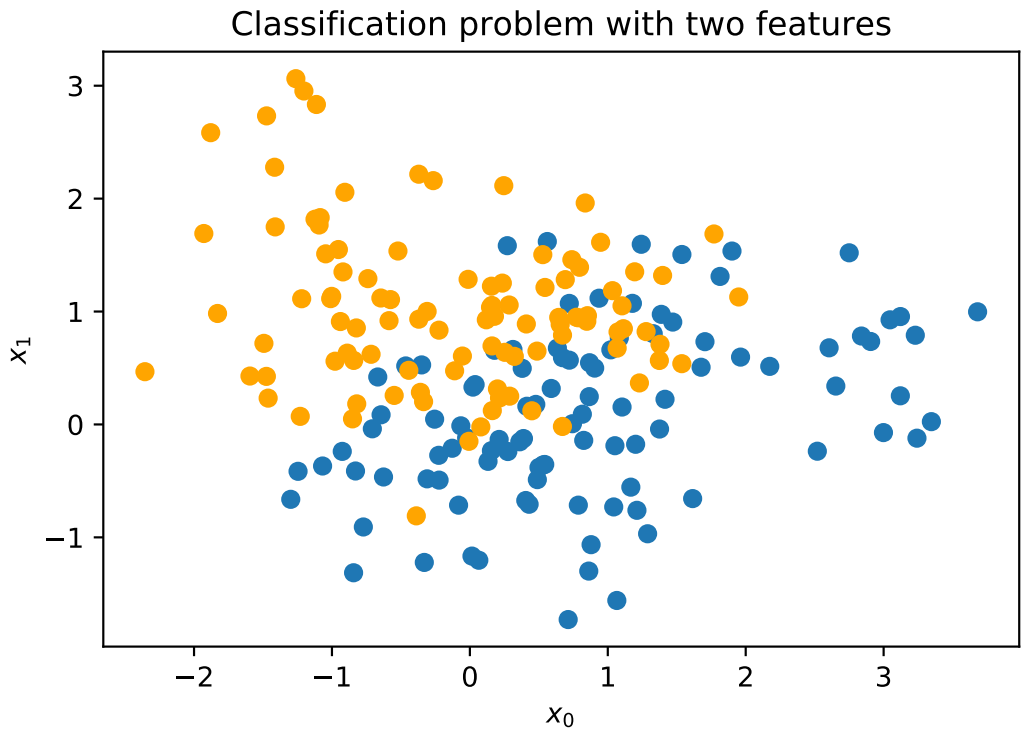
In order to design a machine learning algorithm for a specific task we are given a dataset of examples represented by x_i.

Each example is (optionally) associated with a target y_i.

The target can be categorical, such as class membership (e.g. $y_i = \{healthy, diseased\}$) (this is a classification problem), or continuous (e.g. area, volume etc.) (this is a regression problem).

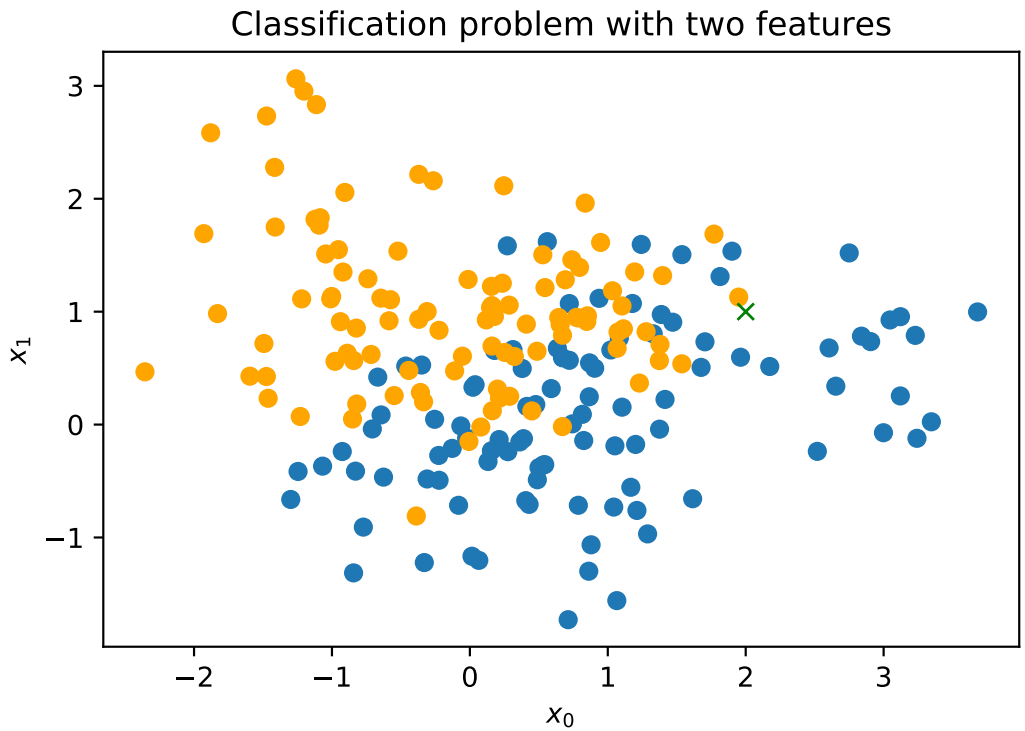
An intuitive model for classification

Assume we have a dataset with two classes (e.g. "benign" and "malignant") and each example in our dataset (e.g. a CT scan) is represented with two features: x_0 and x_1. For the purpose of this discussion, it does not matter which specific features are in question.



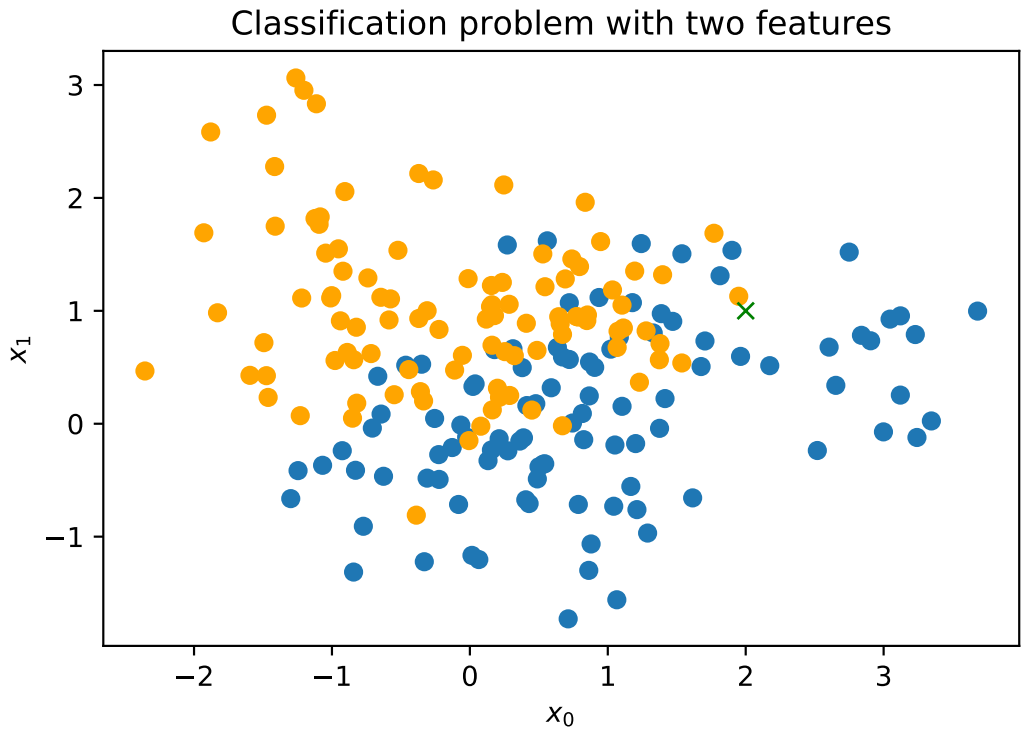
An intuitive model for classification

The goal of the machine learning model: when a new data point comes (e.g. a CT scan from a new patient) assign it to one of the two classes (i.e. classify the CT scan as "benign" or "malignant").

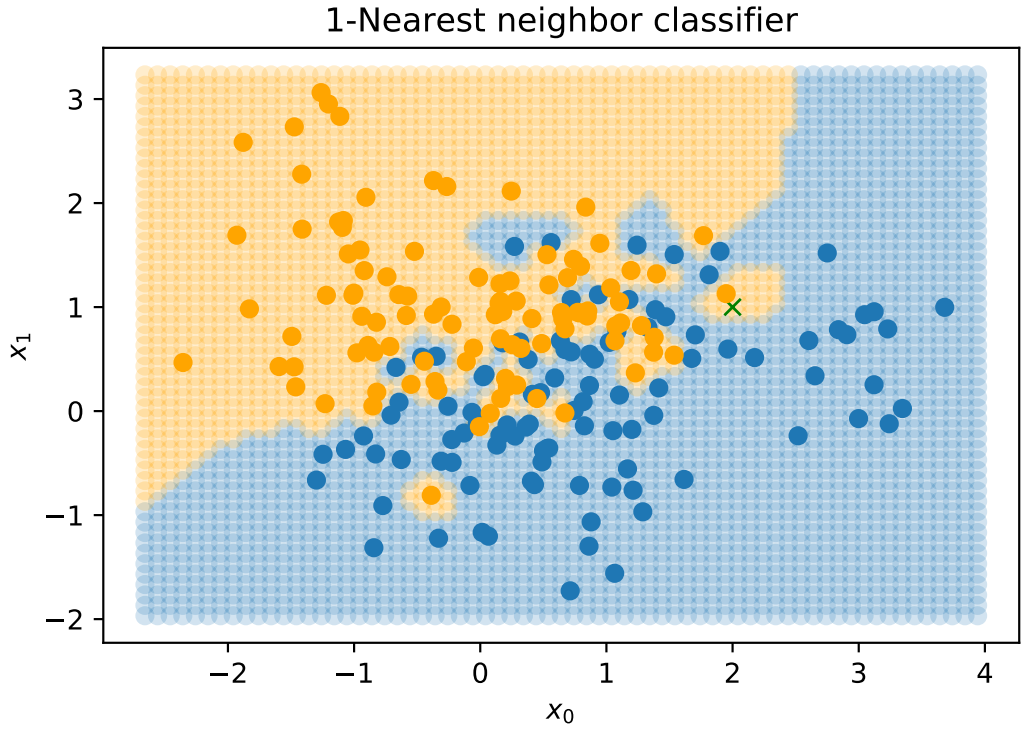


Nearest neighbour classifier

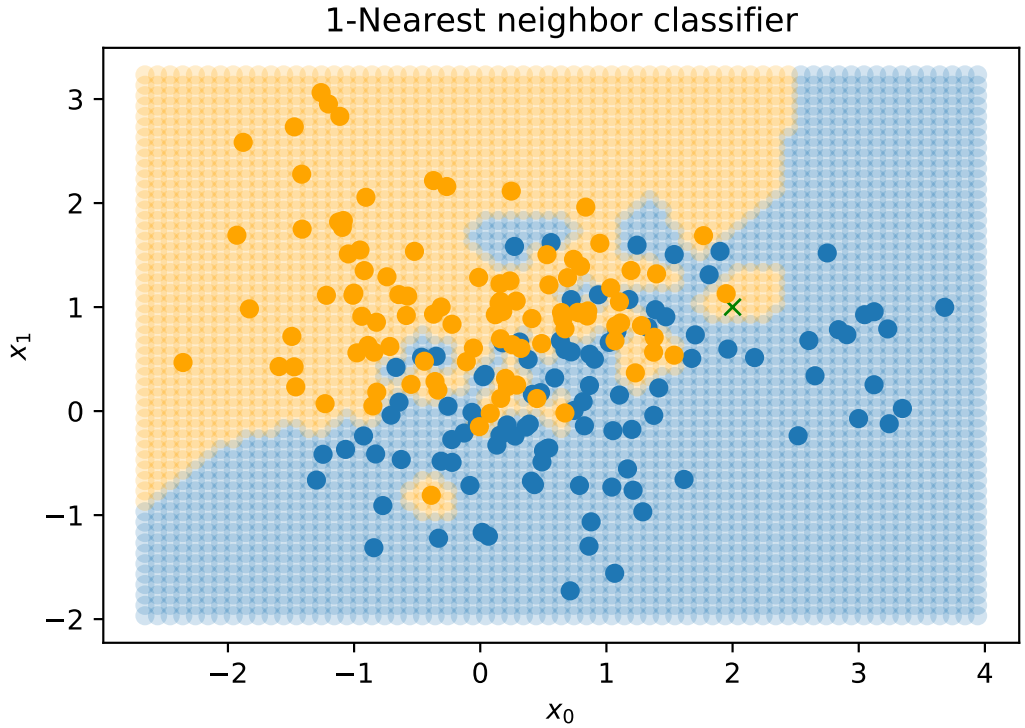
Assign the new data point the class of its nearest neighbour in feature space.



Decision boundary of a NN classifier

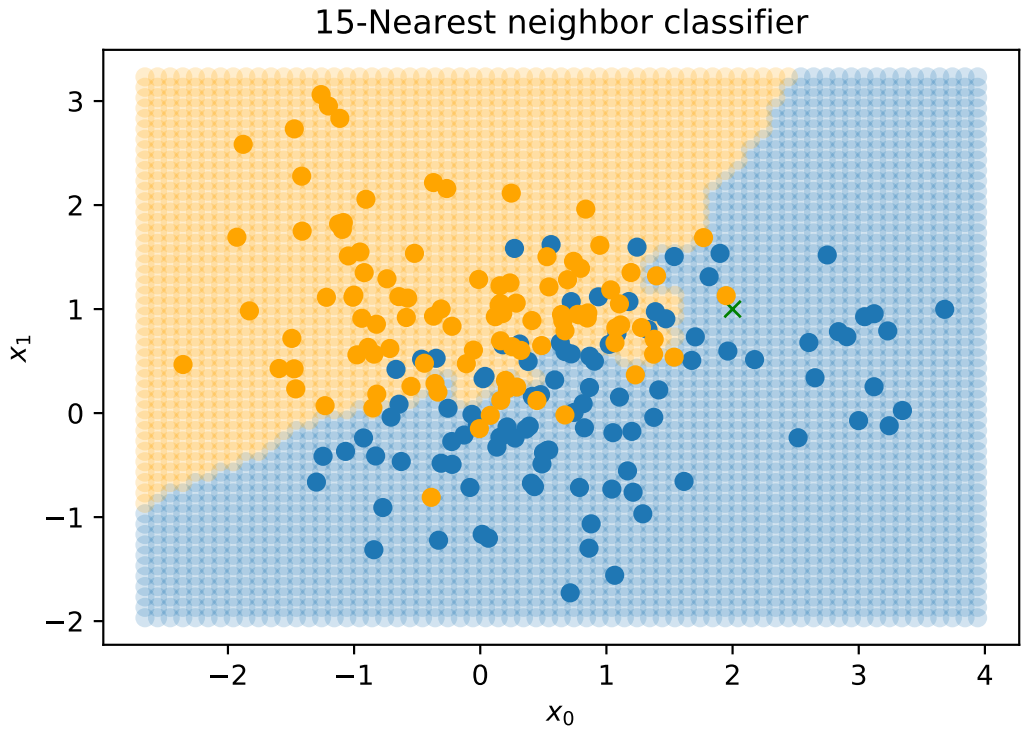


Decision boundary of a NN classifier



Is this a good decision boundary?

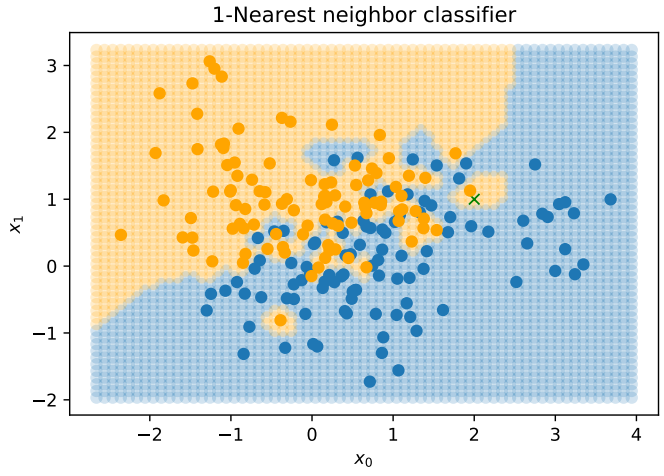
A more general method

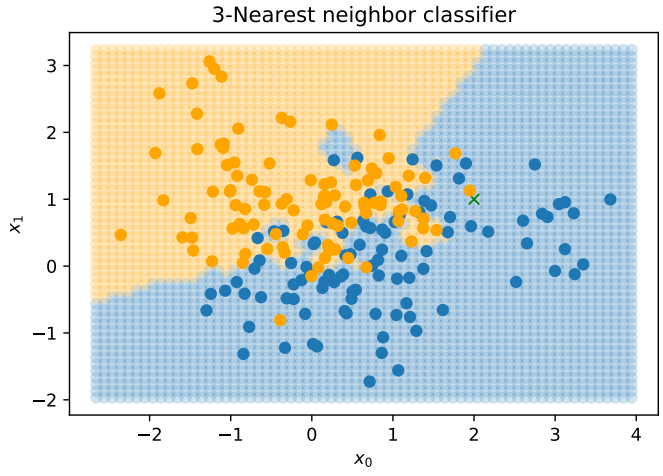


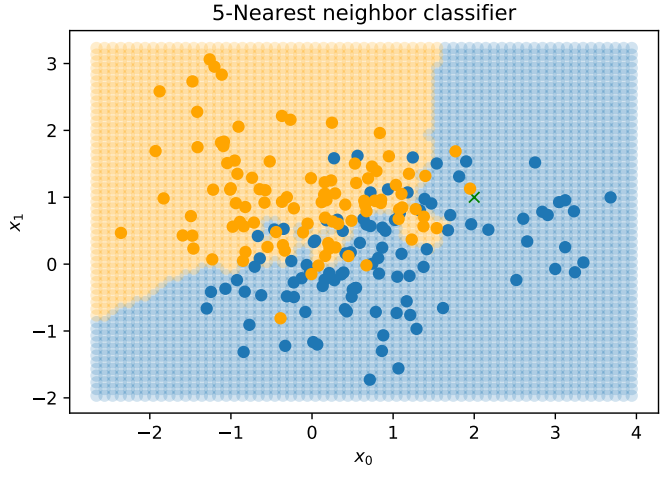
Is this a better decision boundary?

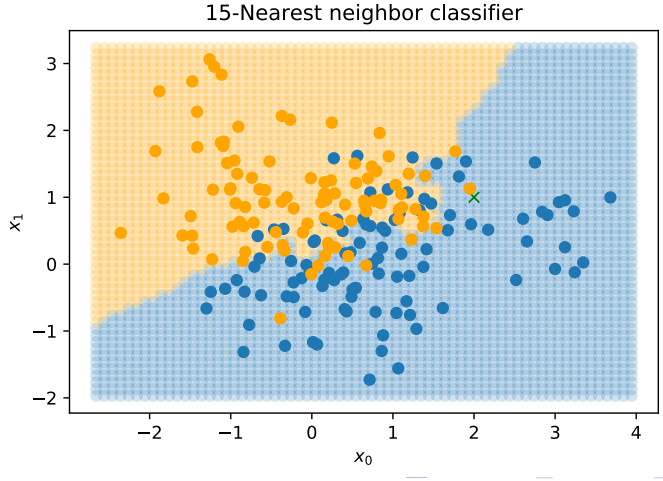
k-Nearest neighbours classifier

Assign the new data point the majority class of its k nearest neighbour in feature space.









How is the class determined with k-NN?

For a new example with features $\boldsymbol{x}_{\text{new}} = [x_0, x_1]$, we want to predict the class $\hat{y_{\text{new}}}$.

- Compute the distance to all training samples $\boldsymbol{x}_i$.
  - Most commonly, we use the Euclidean distance:
  - $d(\boldsymbol{x}_{\text{new}}, \boldsymbol{x}_i) = \sqrt{(x_{\text{new},0} - x_{i,0})^2 + (x_{\text{new},1} - x_{i,1})^2}$
- Sort the training samples based on the distance and pick the k nearest ones to the new example.
- Determine the class of the k nearest training samples.
- Assign to $\boldsymbol{x}_{\text{new}}$ the majority class of its nearest training samples (neighbours).

Some notes on extending k-NN

- Extension to more than two classes is trivial.
- Using k-NN for regression is also possible (e.g. instead of computing the majority class of the nearest neighbours we compute the average target value y).
- Using different distance metric is common, e.g. the L_1-distance:
- $d(\boldsymbol{x}_{\text{new}}, \boldsymbol{x}_i) = |x_{\text{new},0} - x_{i,0}| + |x_{\text{new},1} - x_{i,1}|$

Some remaining questions...

Q: We are in principle free to choose the value for k. But how to do this?

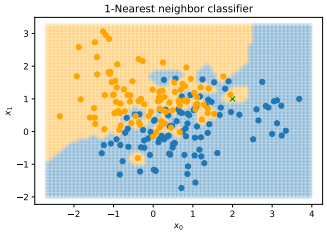
Some remaining questions...

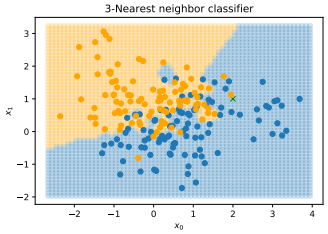
Q: We are in principle free to choose the value for k. But how to do this?

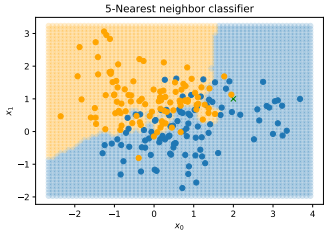
A: "Pick the value for k that gives the best performance."

Performance as a function of k

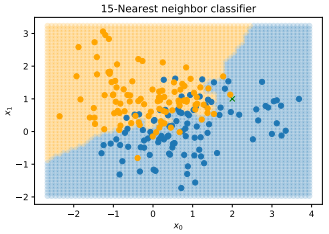
"Pick the value for k that gives the best performance."



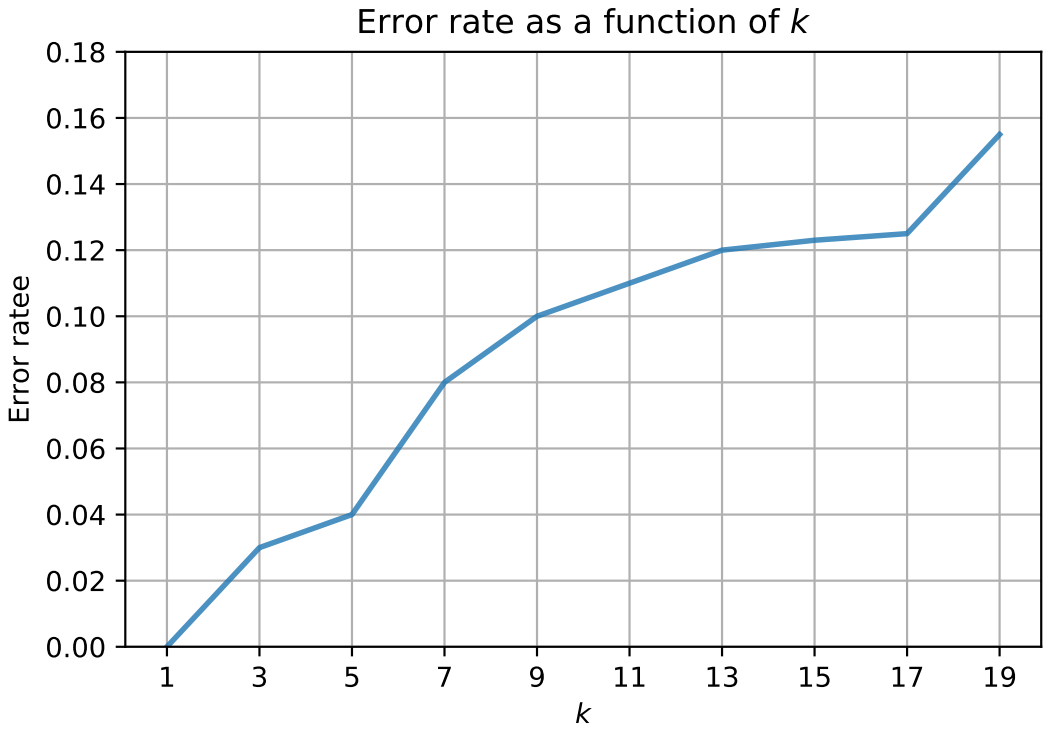




...



...



Choosing the optimal value of k

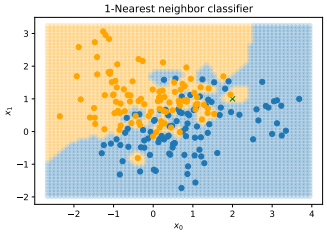
However, if we choose the optimal value of k based on the performance on the training set, we will always select $k = 1$ since in that case the training error is 0.

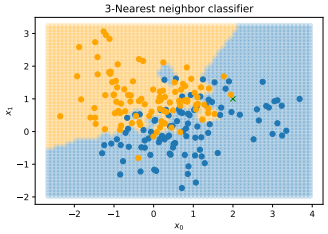
We need to choose k based on the performance on an *independent* test set.

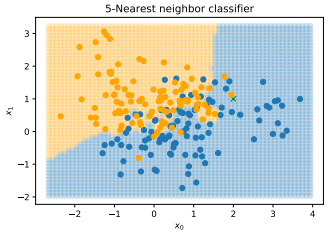
The test set should be independent in the sense that the examples that it contains should in no way be related to the ones in the training set.

Performance as a function of k

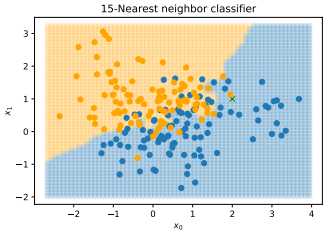
"Pick the value for k that gives the best performance."



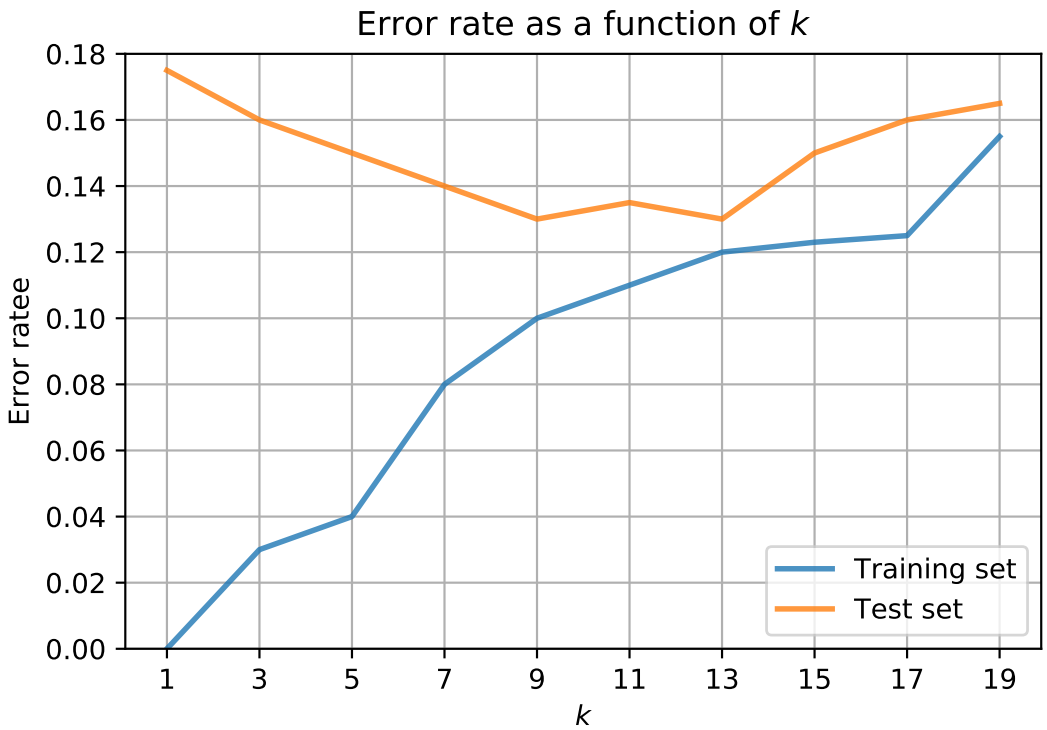




...



...



Generalisation of machine learning methods

The error on the independent test dataset is called the *generalisation* error. It tell us how well we can expect our classifier to generalise its performance on new, previously unseen examples.

In machine learning, we only care about the generalisation error since it is always trivial to design a machine learning model that has perfect performance on the training set (e.g. use 1-NN).

Generalisation and complexity

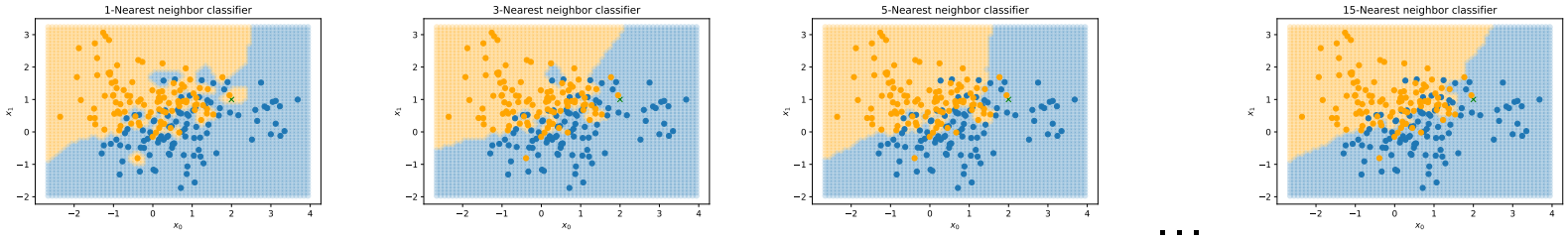
There is a relationship between the complexity of the machine learning model and its generalisation ability.

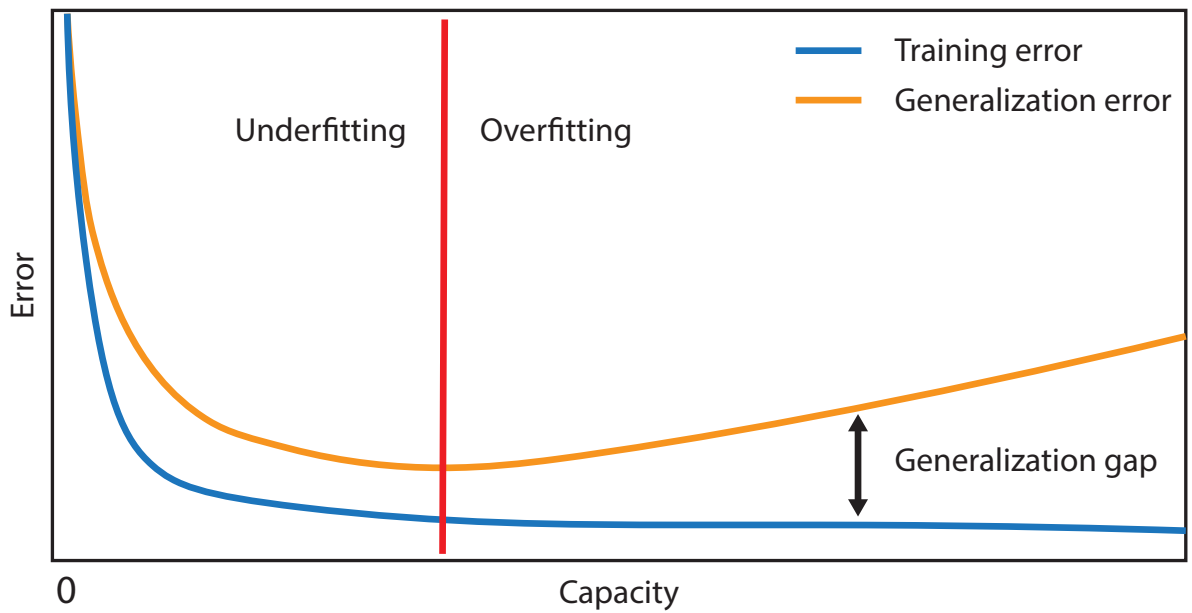
Classifiers that produce **simple** decision boundaries can have higher training errors but usually generalise better to new samples.

Classifiers that produce **complex** decision boundaries can have lower training errors but usually generalise worse to new samples.

Generalisation and complexity

Note that complexity **decreases** with k.





Types of machine learning models

Previously: supervised vs unsupervised.

Also:

- Parametric models
  - The number of parameters is fixed, i.e. it does not grow with the number of training samples
  - Once the model is trained (the parameters of the model are determined), we can "throw away" the training dataset.
  - Linear regression is an example (see next week).
- Non-parametric models
  - The number of parameters is not fixed, and it grows with the number of training samples.
  - k-NN is an example of a non-parametric machine learning model.

Questions?