Welcome!

A Functional Machine Learning Classifier

Paul Brabban

sheffieldml.org.uk & (def shef)

The Problem

Meet the Irises

Iris Setosa

Image: Денис Анисимов via Wikimedia

Iris Versicolor

Image: Danielle Langlois via Wikimedia

Iris Virginica

Image: Frank Mayfield via Wikimedia

The Problem

Which is this?

Iris...?

Image: Danielle Langlois via Wikimedia

Machine Learning

Techniques that let machines learn from experience, without being explicitly programmed.

Machine Learning

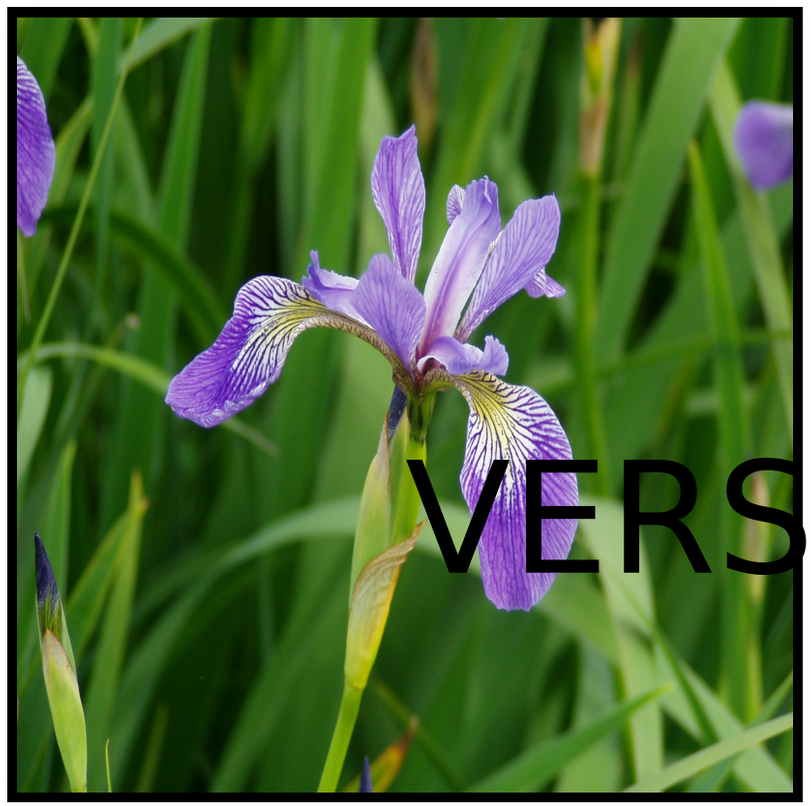
Machine learning models predict things.

- how much a house will sell for
- which numeric digit a digitised photo shows
- whether an applicant will pay back a loan
- whether a image of cells is normal or cancerous

and uncountably more...

Machine Learning

Even what species a particular iris belongs to!



Iris...?

Image: Danielle Langlois via Wikimedia

Classification

We're solving a classification problem

is this iris versicolor, setosa or virginica?

There are other kinds...

- regression, like predicting a numeric house price
- unsupervised, when we don't know the answers

Examples

To learn a supervised problem, we need examples

5.1,3.5,1.4,0.2,Iris-setosa

- Four Features:
  - Petal width & length
  - Sepal width & length
- One Label (Setosa | Versicolor | Virginica)

Training and Testing

Training

- training examples are fed to the ML algorithm
- a model is built that can predict for new examples

Testing

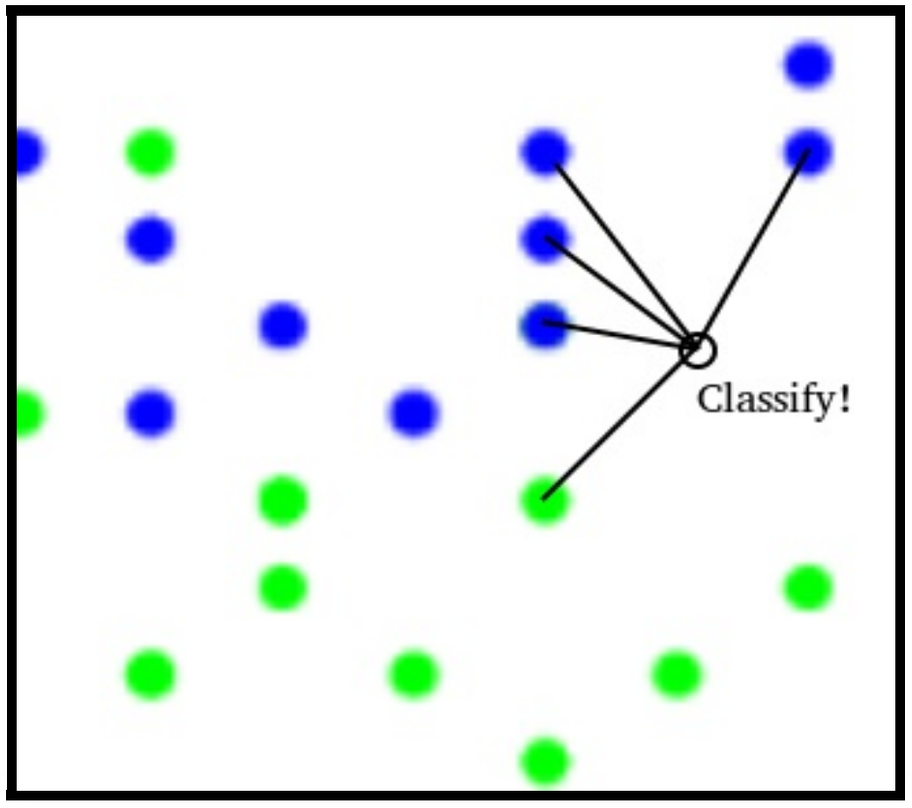
- the model predicts for examples it hasn't seen before
- "goodness" of the model is assessed

The kNN classifier

We're going to use a classic algorithm:

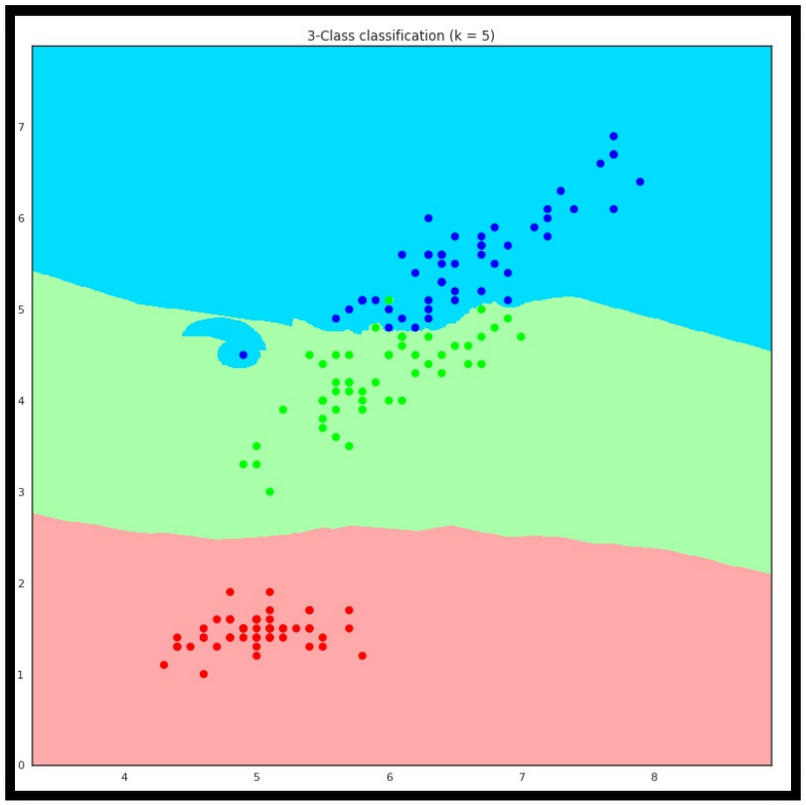
a k-Nearest Neighbours classifier.

The kNN classifier



A new data point's nearest neighbours with k=5. Classify as Blue!

The kNN classifier



Scatter plot of training data and predictions. Features plotted are petal length vs. sepal length.

Training

Most algorithms have a 'training' phase where they deduce and optimise a target function.

The k-NN classifier doesn't really have a training phase as it just 'memorizes' the training data.

Validation

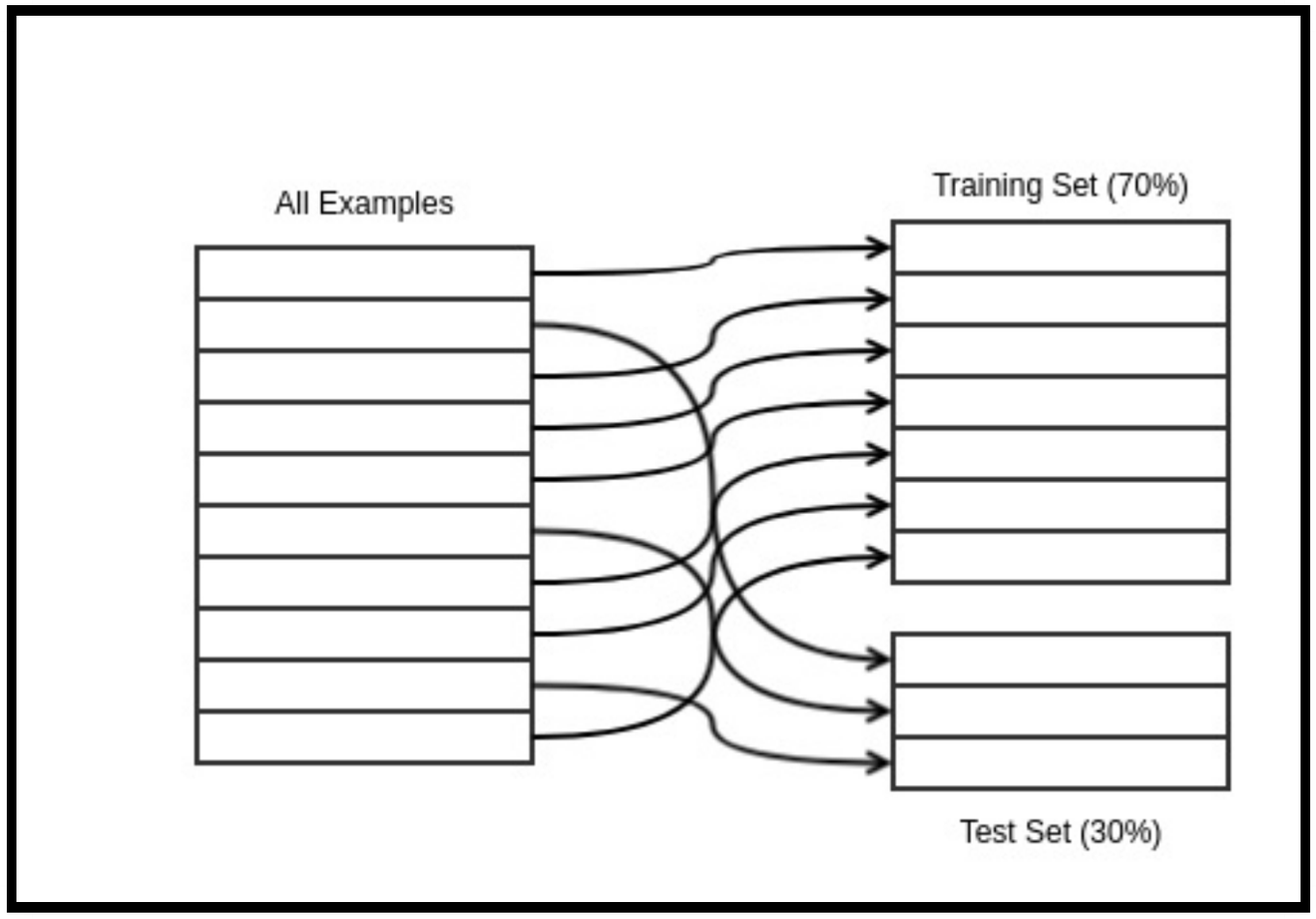
...but is it good at predicting the species?

How do we measure the effectiveness of our algorithm?

Train/Test Split

- assign some examples to a "training" set (say 70%)
- and the rest to a "test" set (say 30%)
- have the algorithm memorize the training set
- predict the classes of the test set
- how many did it get right? (%)

Like so...

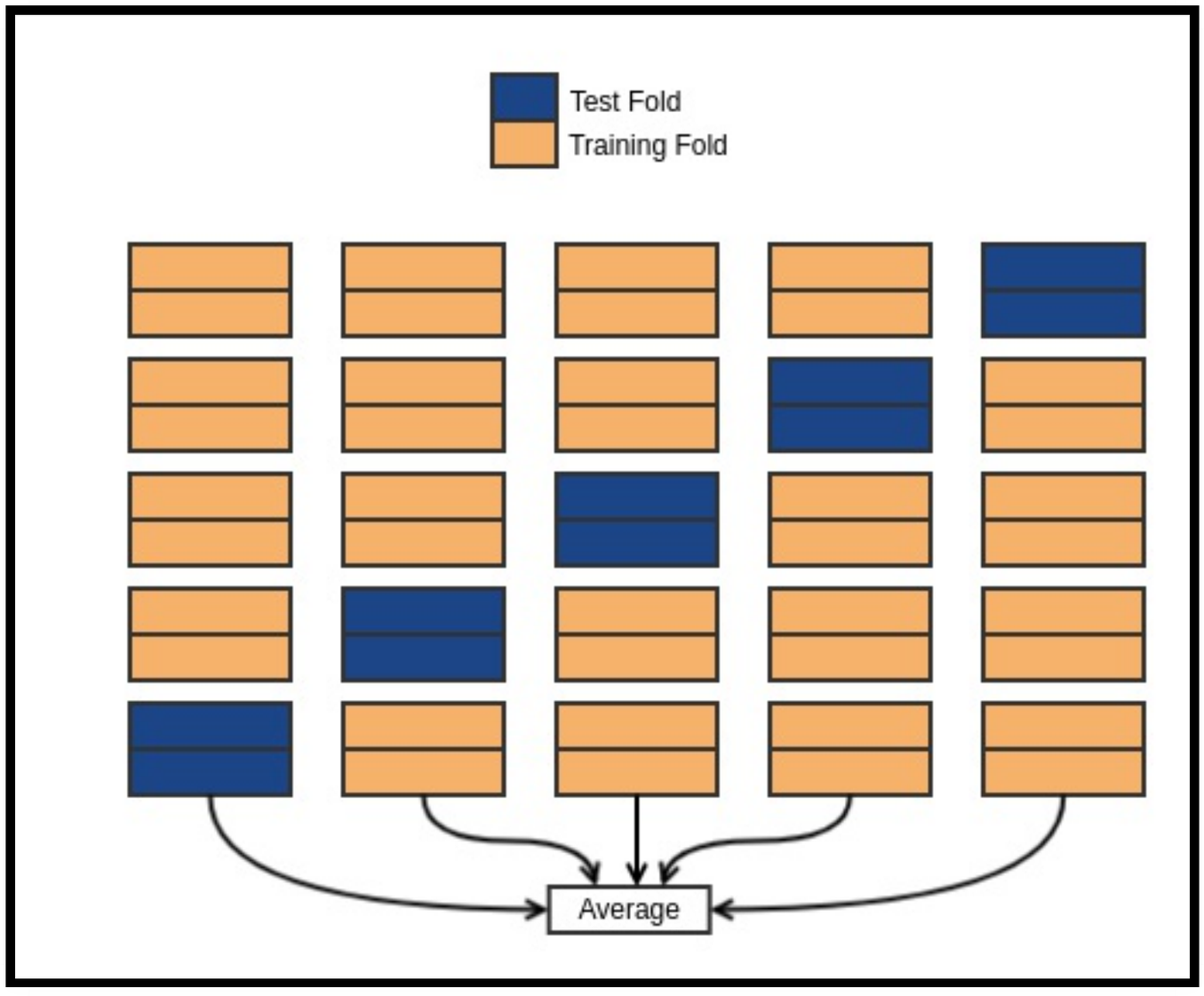


k-Fold Cross-Validation

Make better use of your data!

- choose say k = 5, then
- randomly assign examples to 5 equal sized "folds"
- train with 4/5 "folds", test with the other
- 5 times
- average the % correct

Like so...



That's it!

Instructions and the data set are at https://github.com/defshef/dojo-knn/README.md

- Any language you like
- Work alone or in groups
- If you're stuck shout up for a hint
- Last 20 mins will be a show-and-tell by... YOU LOT!
- Thank you and enjoy!