Machine Learning

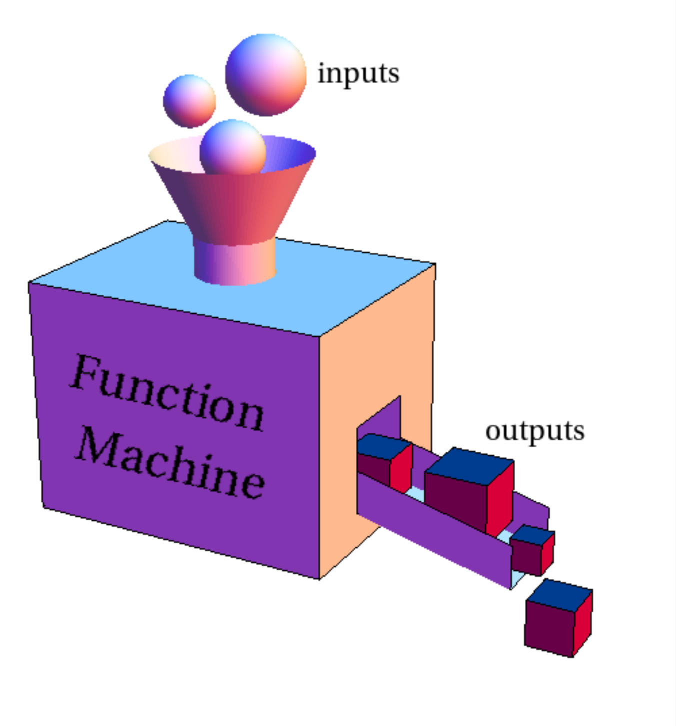
Dr. Michelle Lochner

LSST Dark Energy School 2017

What is machine learning?

What is machine learning?

- Essentially, automatically building a (usually highly nonlinear) model that maps a given input to output.
- Different algorithms use different prescriptions for building the model



When to use machine learning

For data exploration (unsupervised learning)

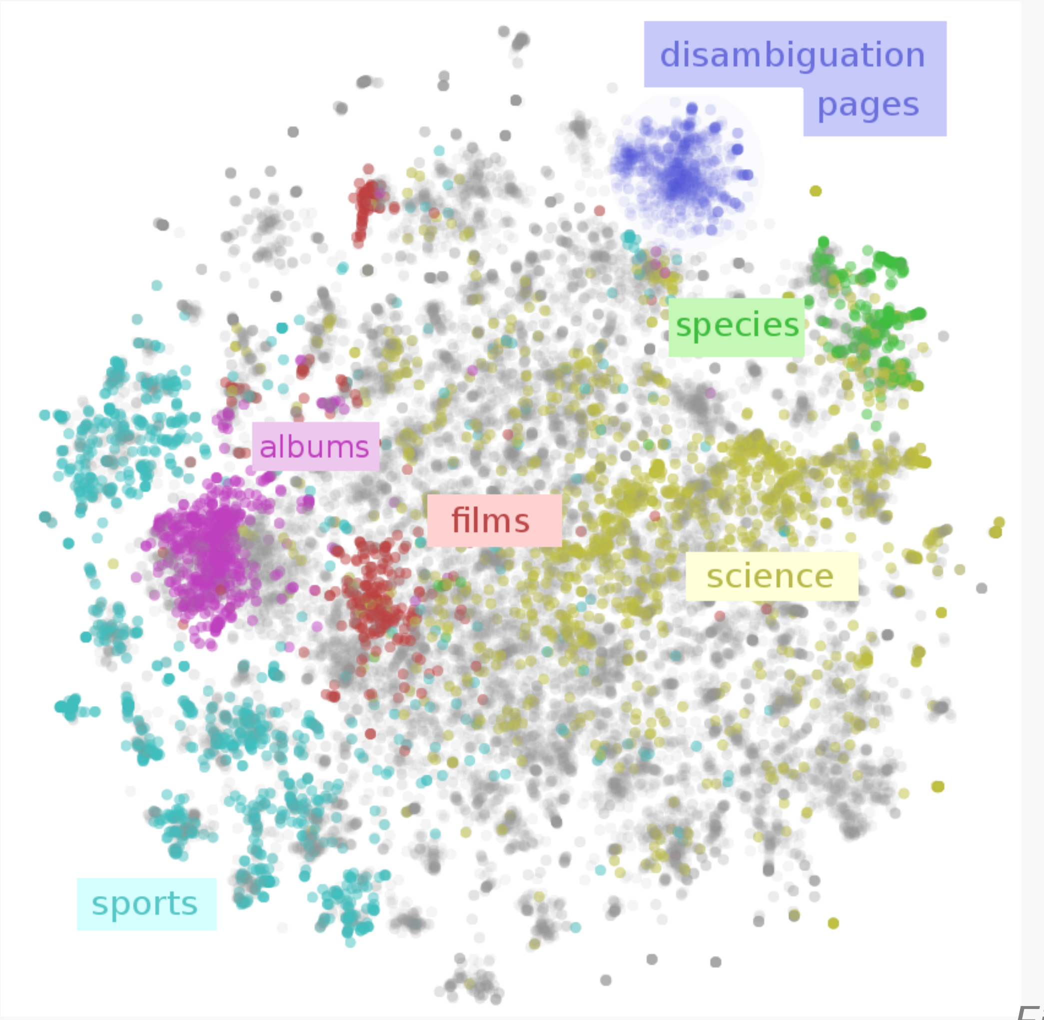
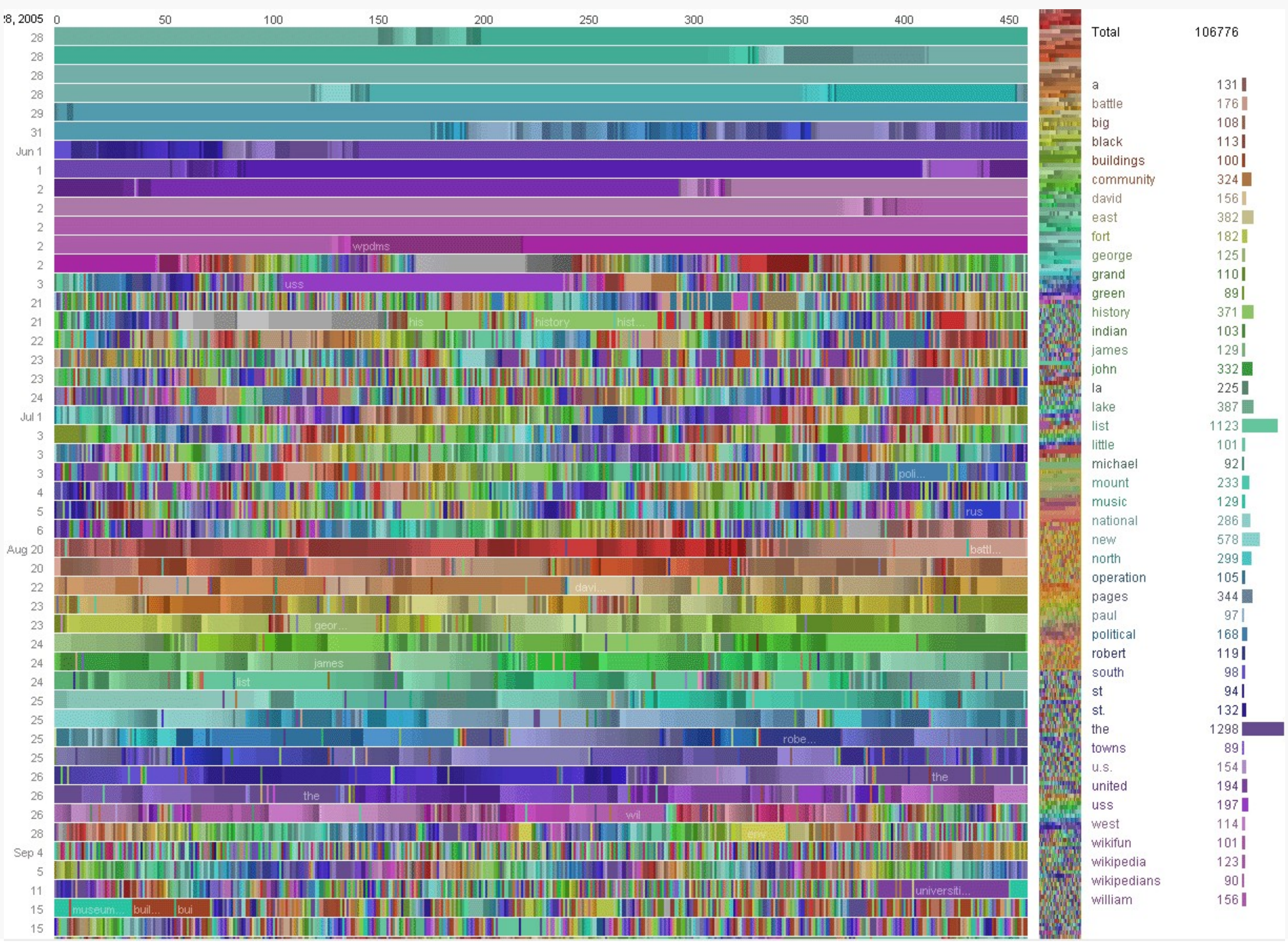


Figure: http://colah.github.io/

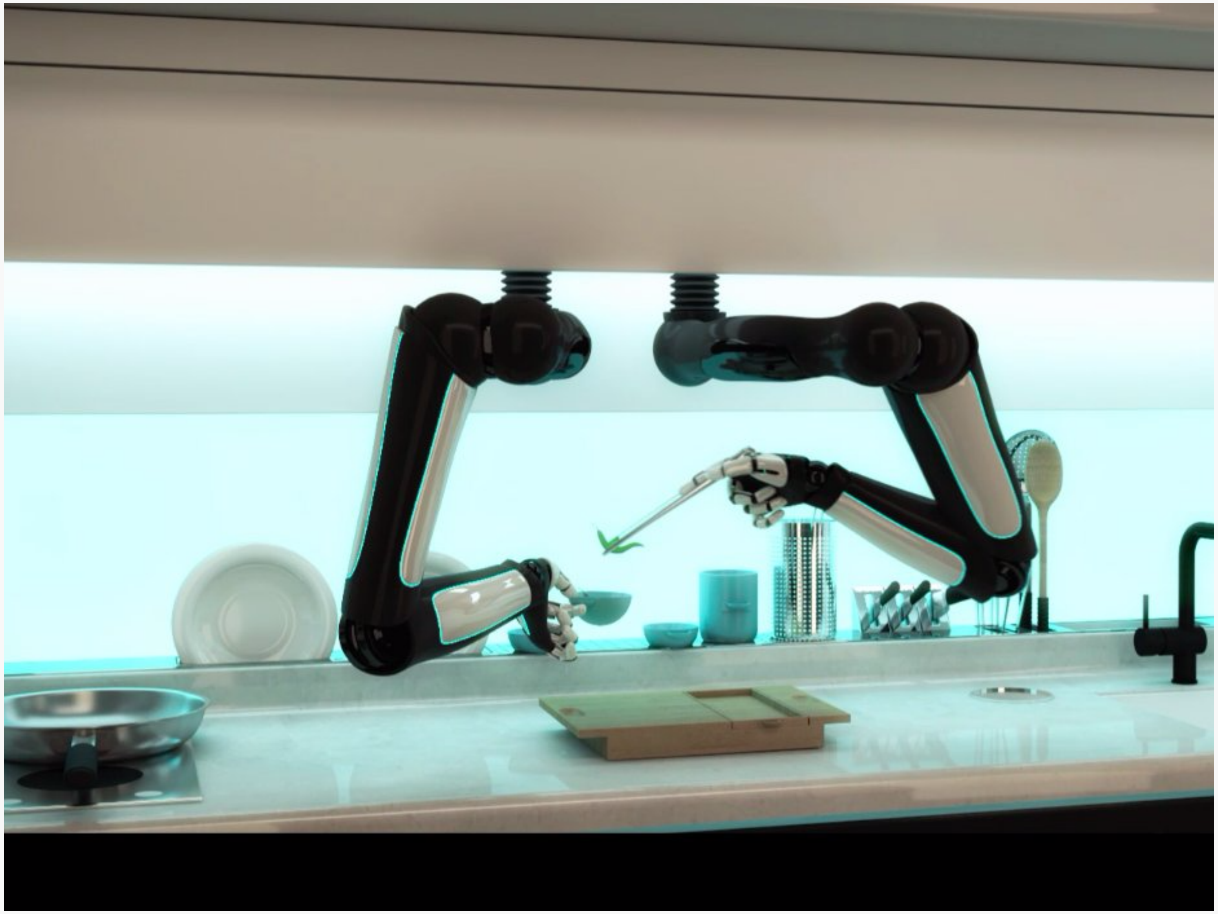
When to use machine learning

When your data are too complex for traditional model development and fitting with statistics



When to use machine learning

When you are too busy/ too lazy to perform a task repeatedly



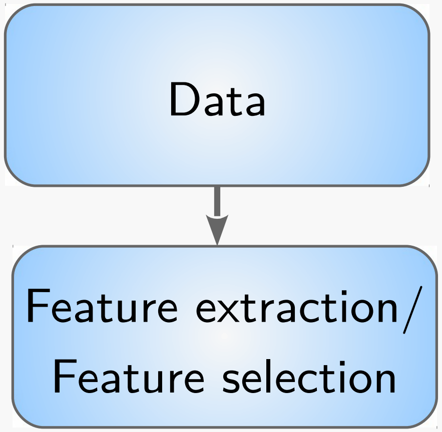
Scikit-learn

```
from python import solution
```

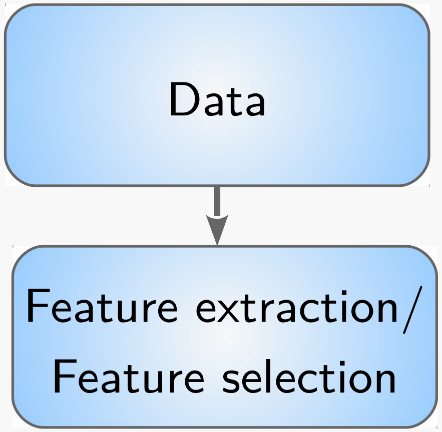
Some definitions

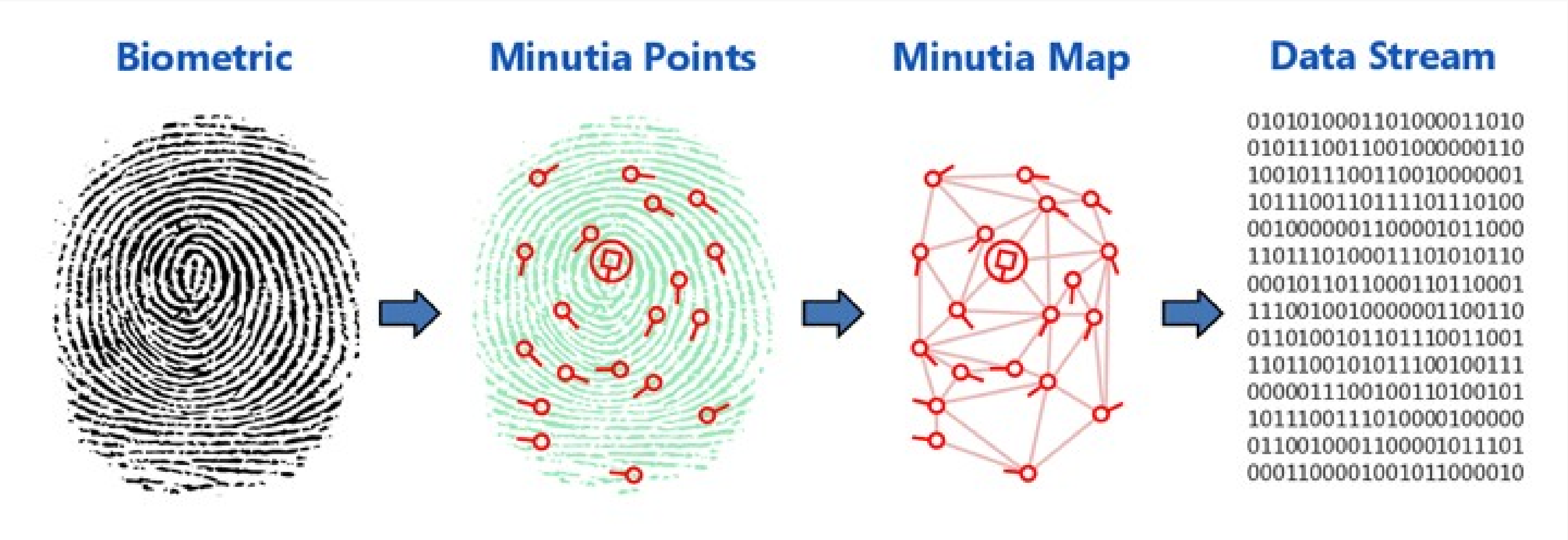
Data

Some definitions

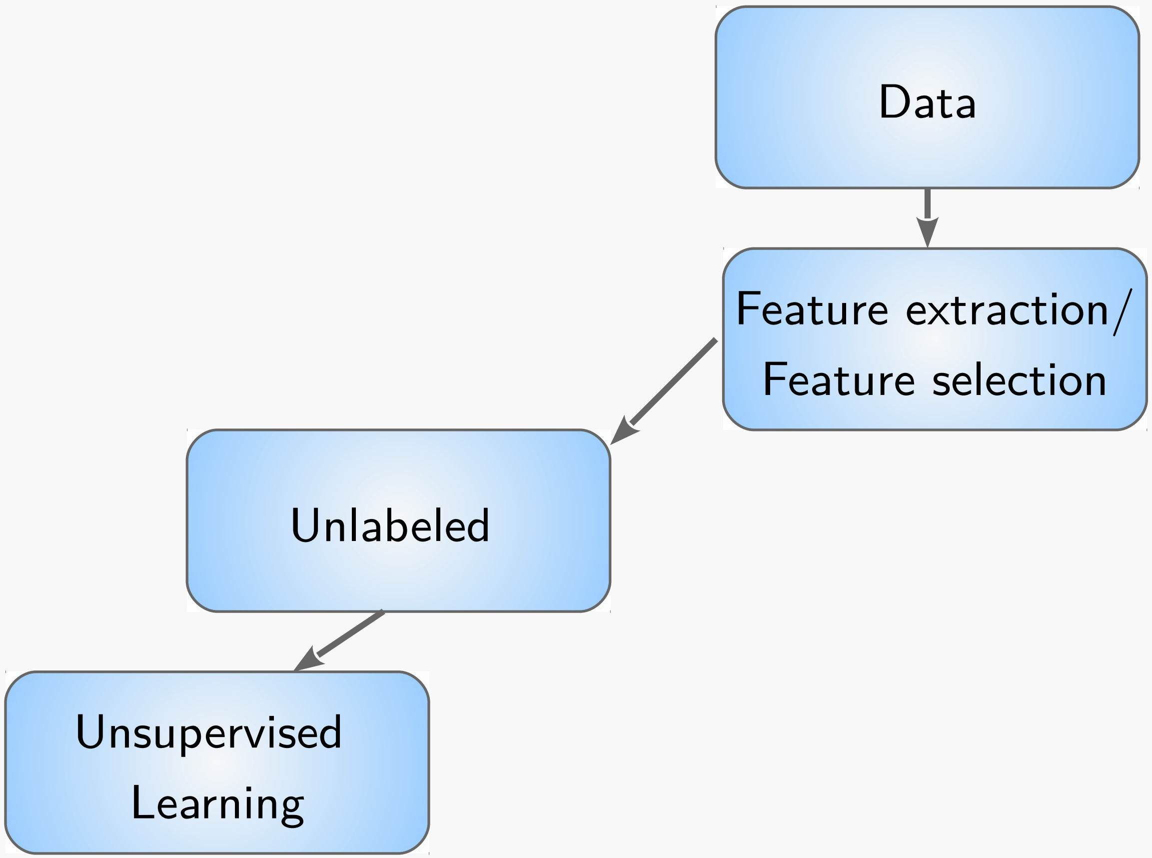


Some definitions





Some definitions

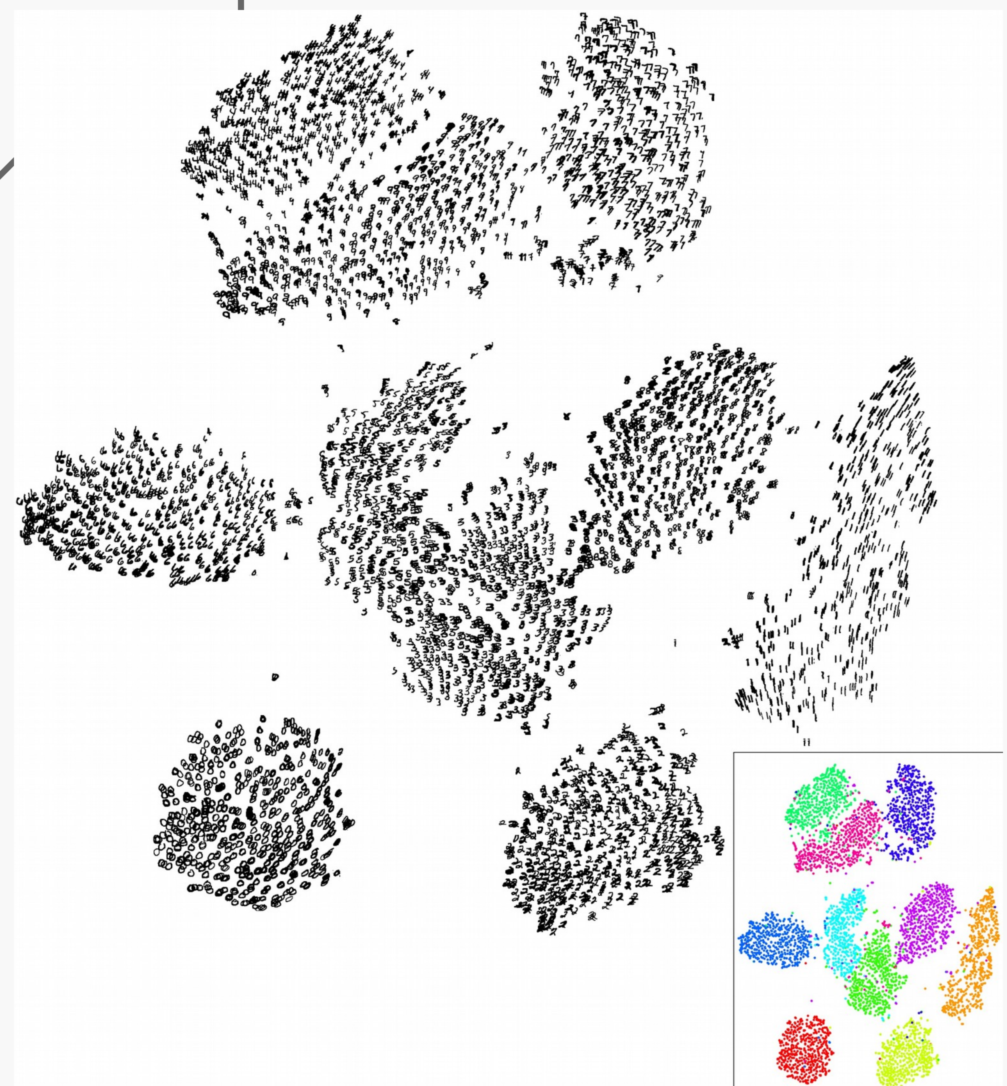


Some definitions

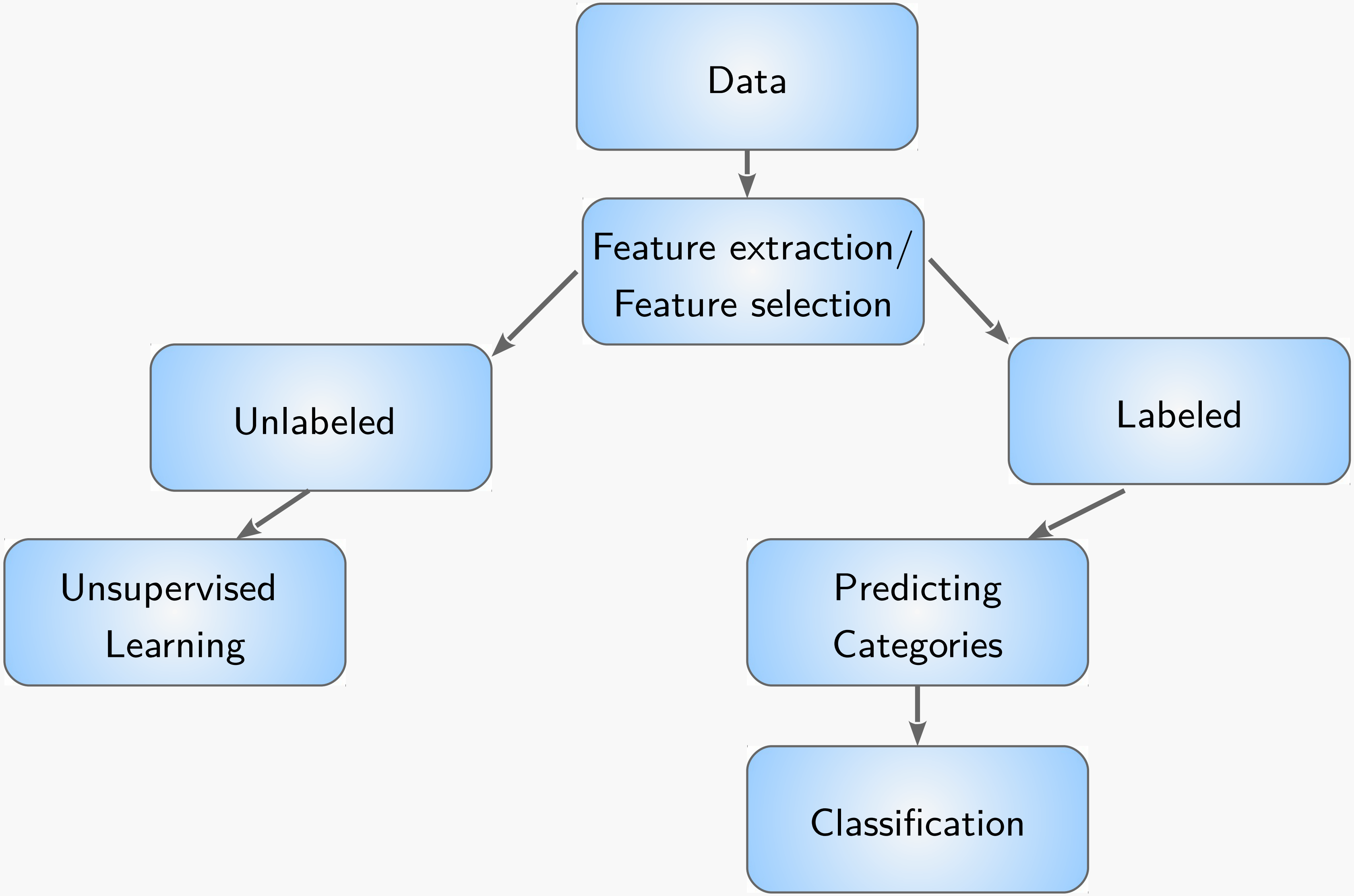
Data

Unlabeled

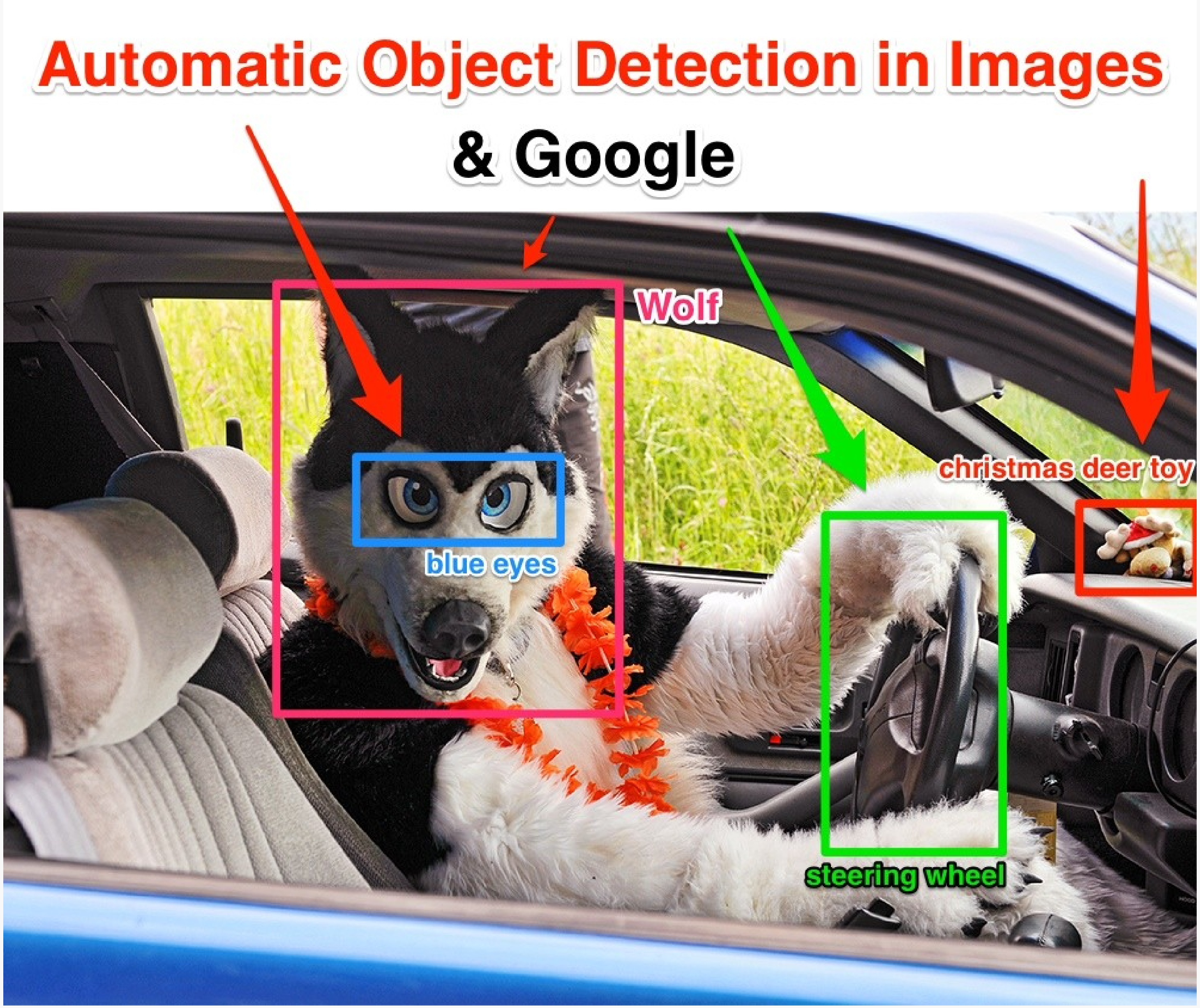
Unsupervised Learning



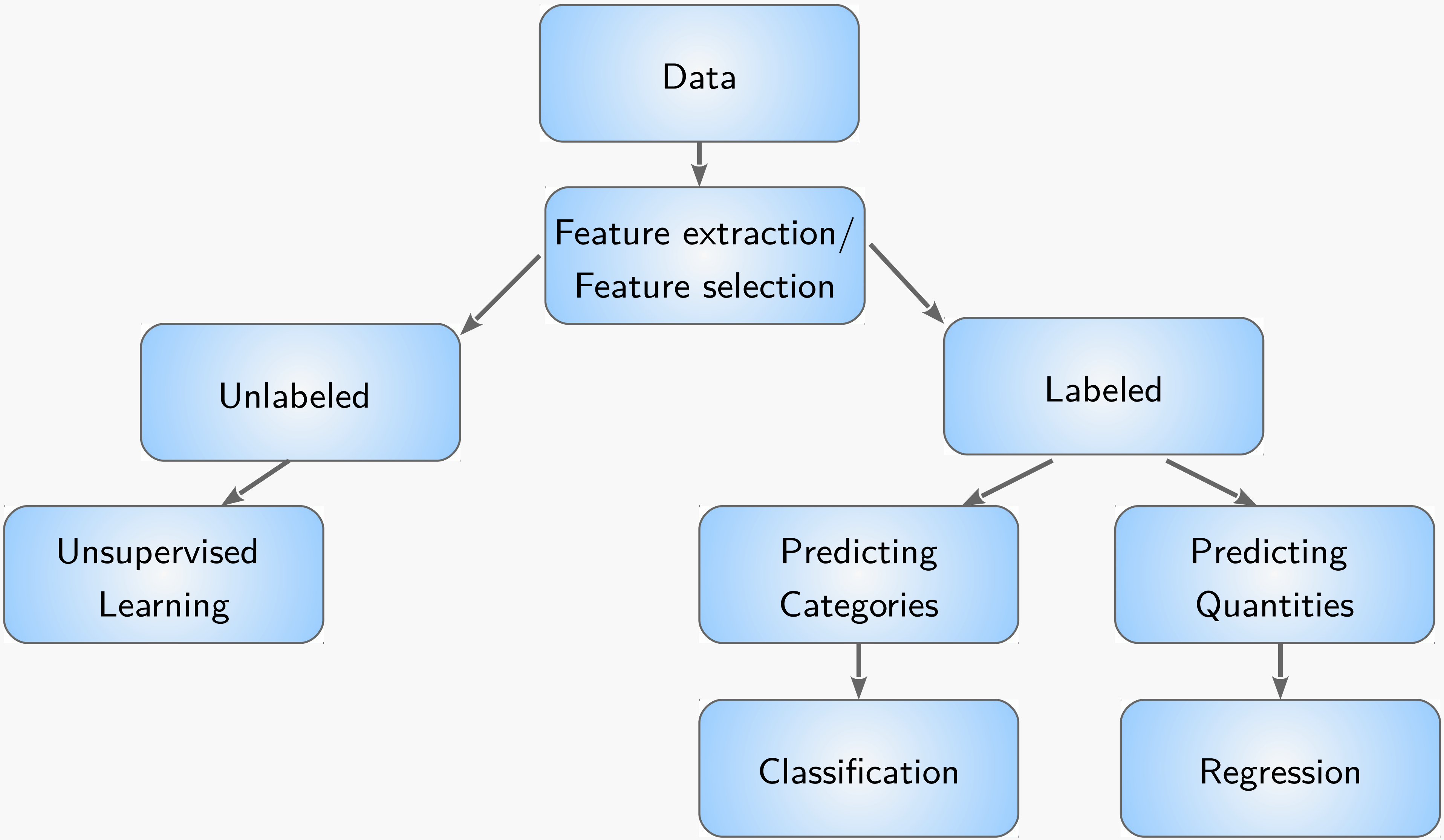
Some definitions



Some definitions

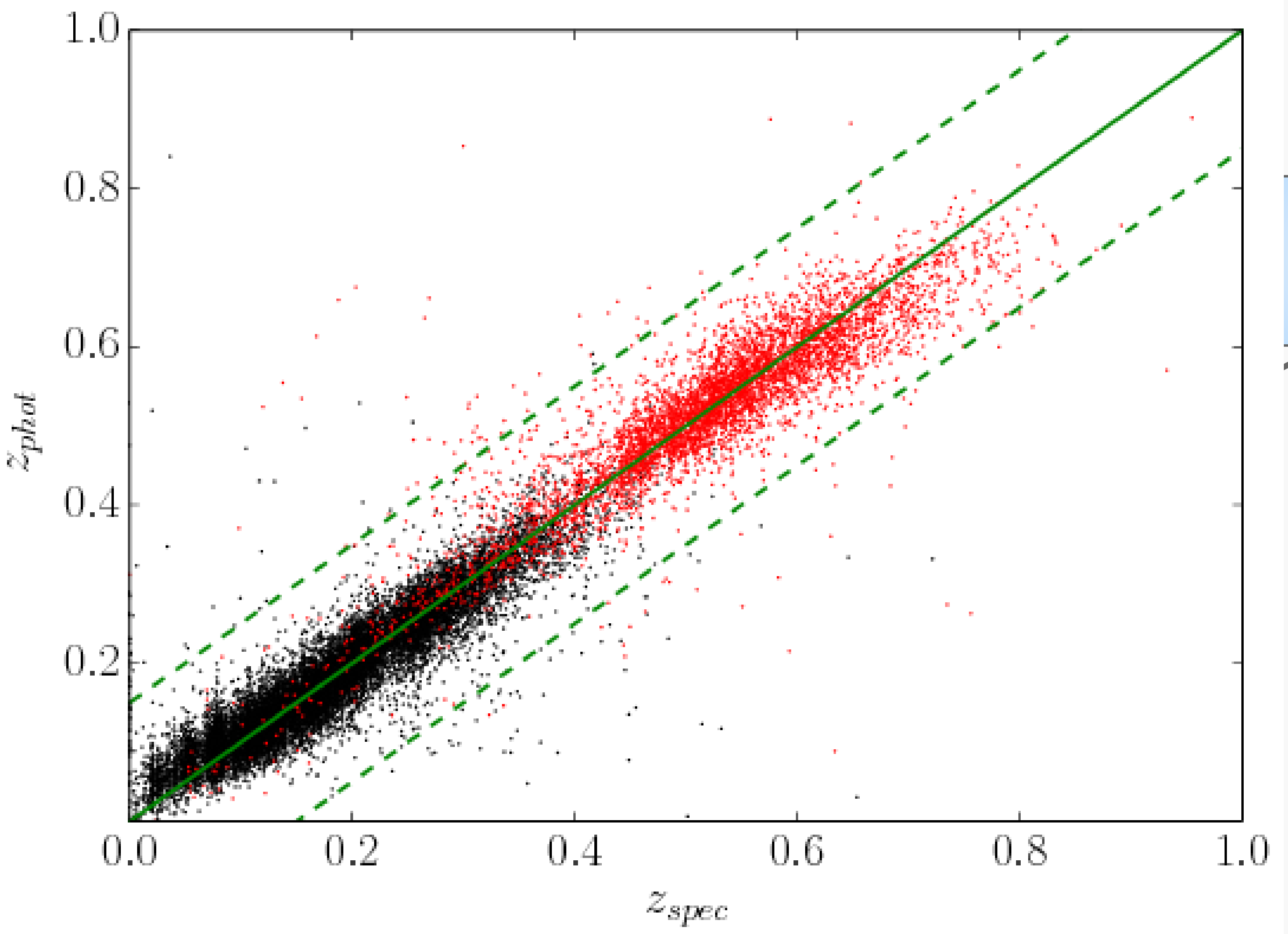


Some definitions



Some definitions

Data



Labeled

Predicting Quantities

Regression

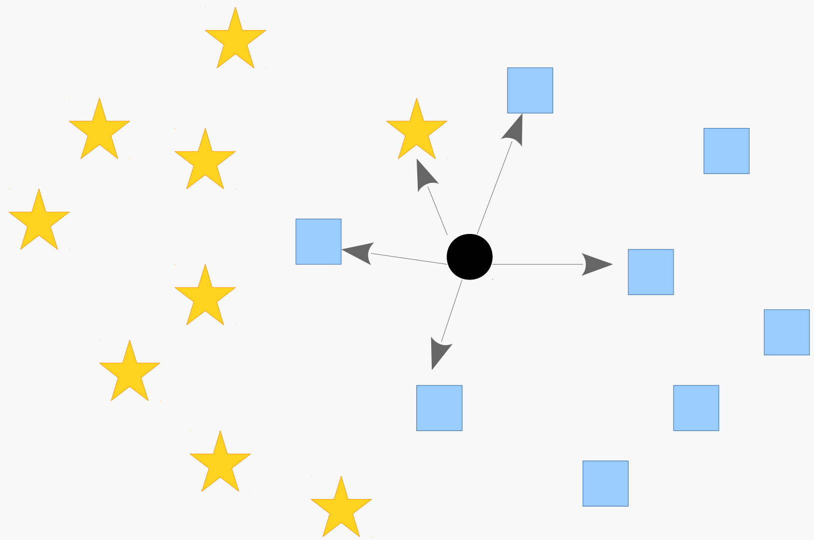
Figure: de Jong et al. (2015), ArXiv: 1507.00742v2

Machine Learning Algorithms

- K-nearest neighbours (KNN)
- Artificial neural networks (ANN)
- Random forests (RF)

K-nearest Neighbours

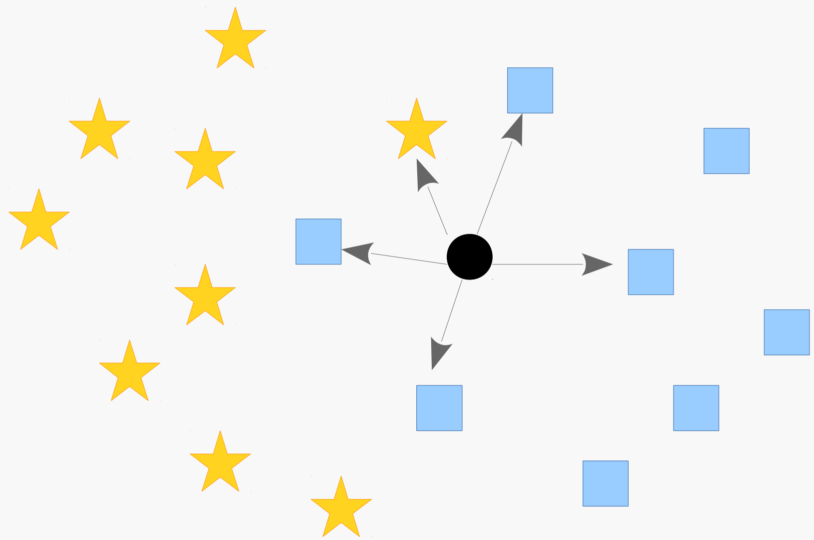
In it's simplest form, classify as the majority class of its k nearest neighbours.



`sklearn.neighbors.KNeighborsClassifier`

K-nearest Neighbours

More sophisticated (and useful) version weights the k nearest neighbours by inverse distance.



Probability of belonging to a particular class is simply the normalised number of votes for that class (inversely weighted by distance)

K-nearest Neighbours

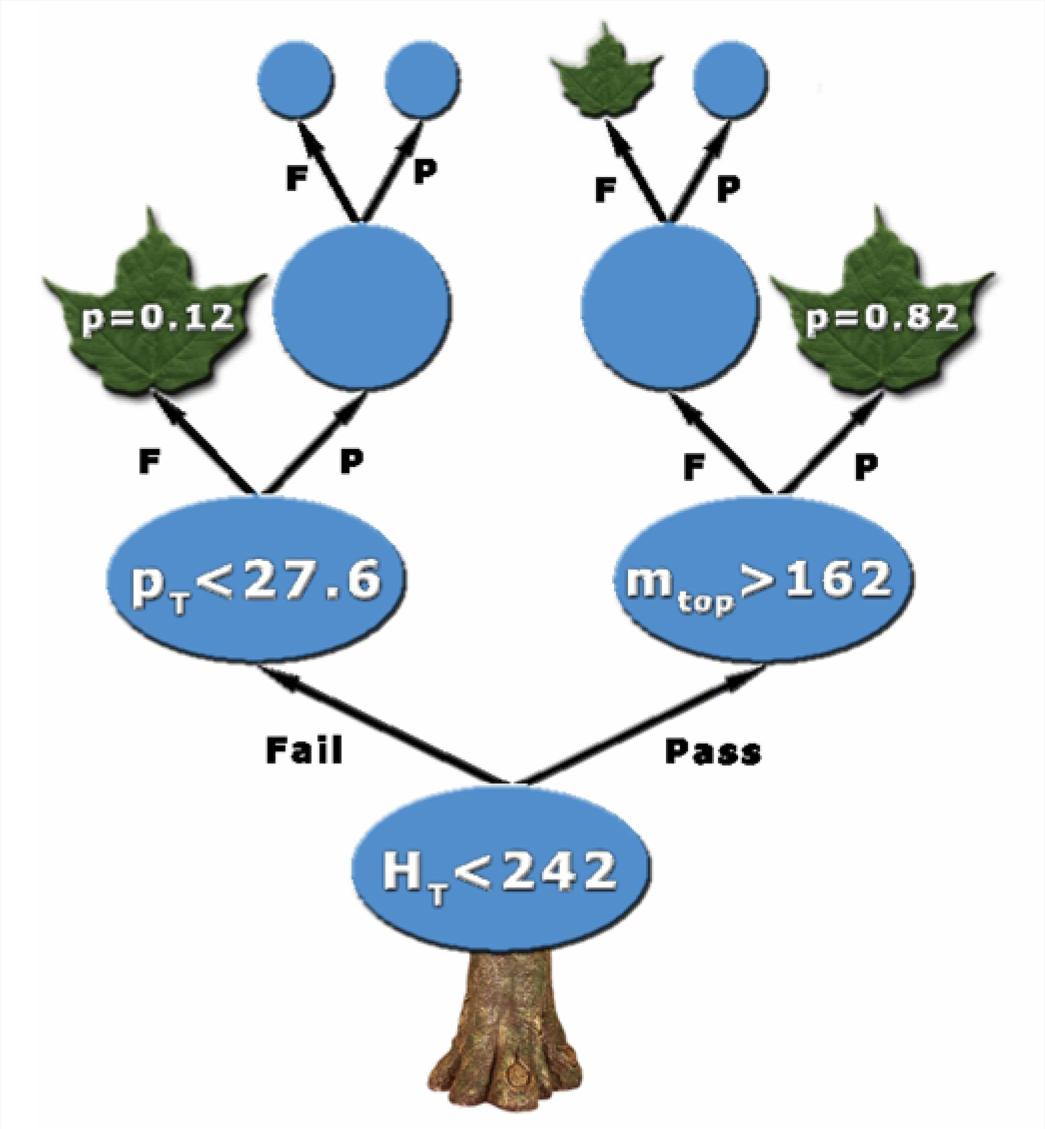
Advantages

- Conceptually very simple
- Easy to tune

Disadvantages

- Suffers very much from the curse of dimensionality
- Underperforms in most cases compared to more advanced algorithms

Decision Trees



Decision trees construct a series of nodes which make splits on a particular feature.

`sklearn.tree.DecisionTreeClassifier`

Constructing Decision Trees

- At each leaf node, decide which is the best feature to split the data on (such that it separates best between classes), and what's the best split value of that feature

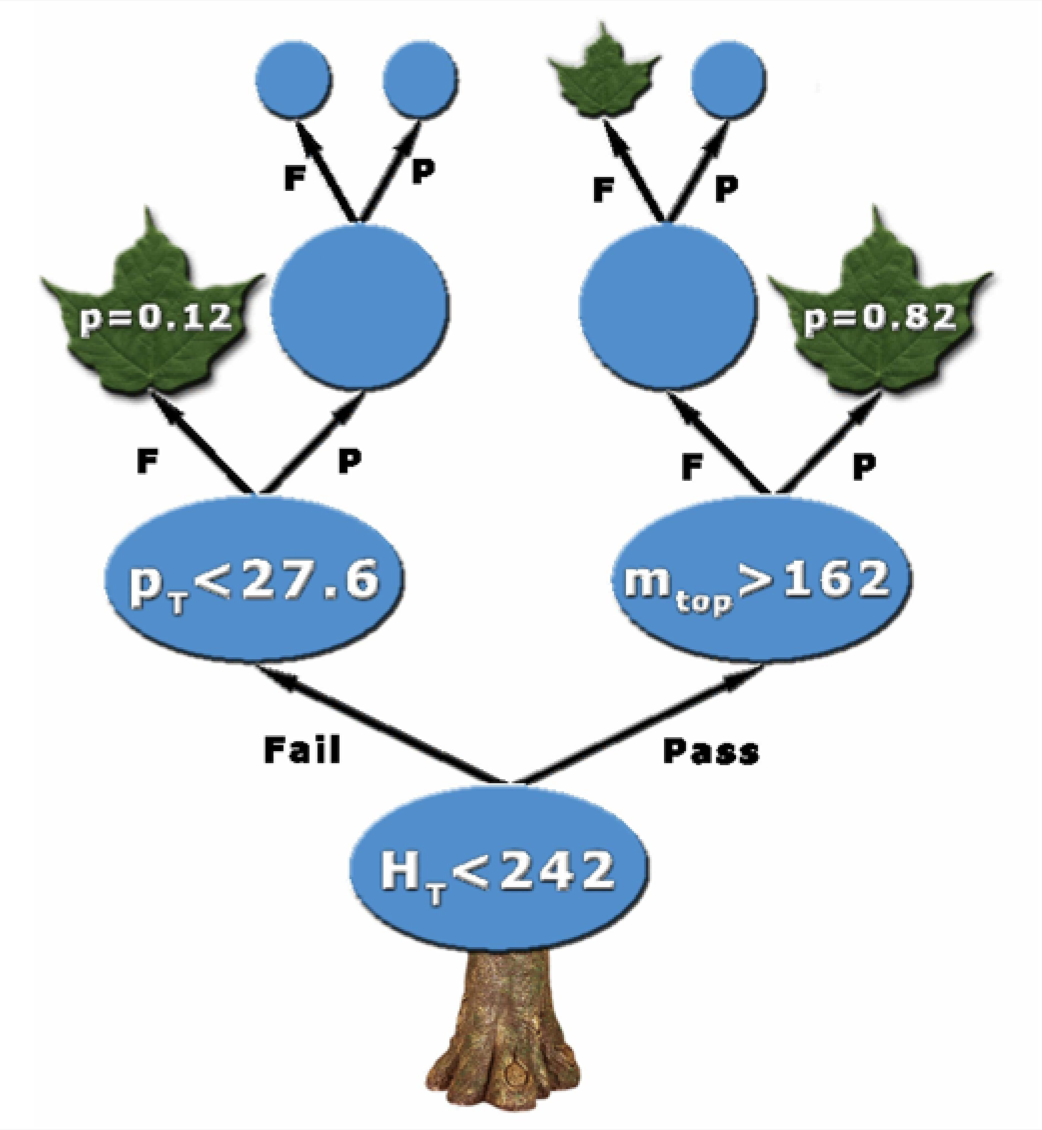
Constructing Decision Trees

- At each leaf node, decide which is the best feature to split the data on (such that it separates best between classes), and what's the best split value of that feature
- To make this choice, use either the entropy or the Gini impurity (see extra slides)

Constructing Decision Trees

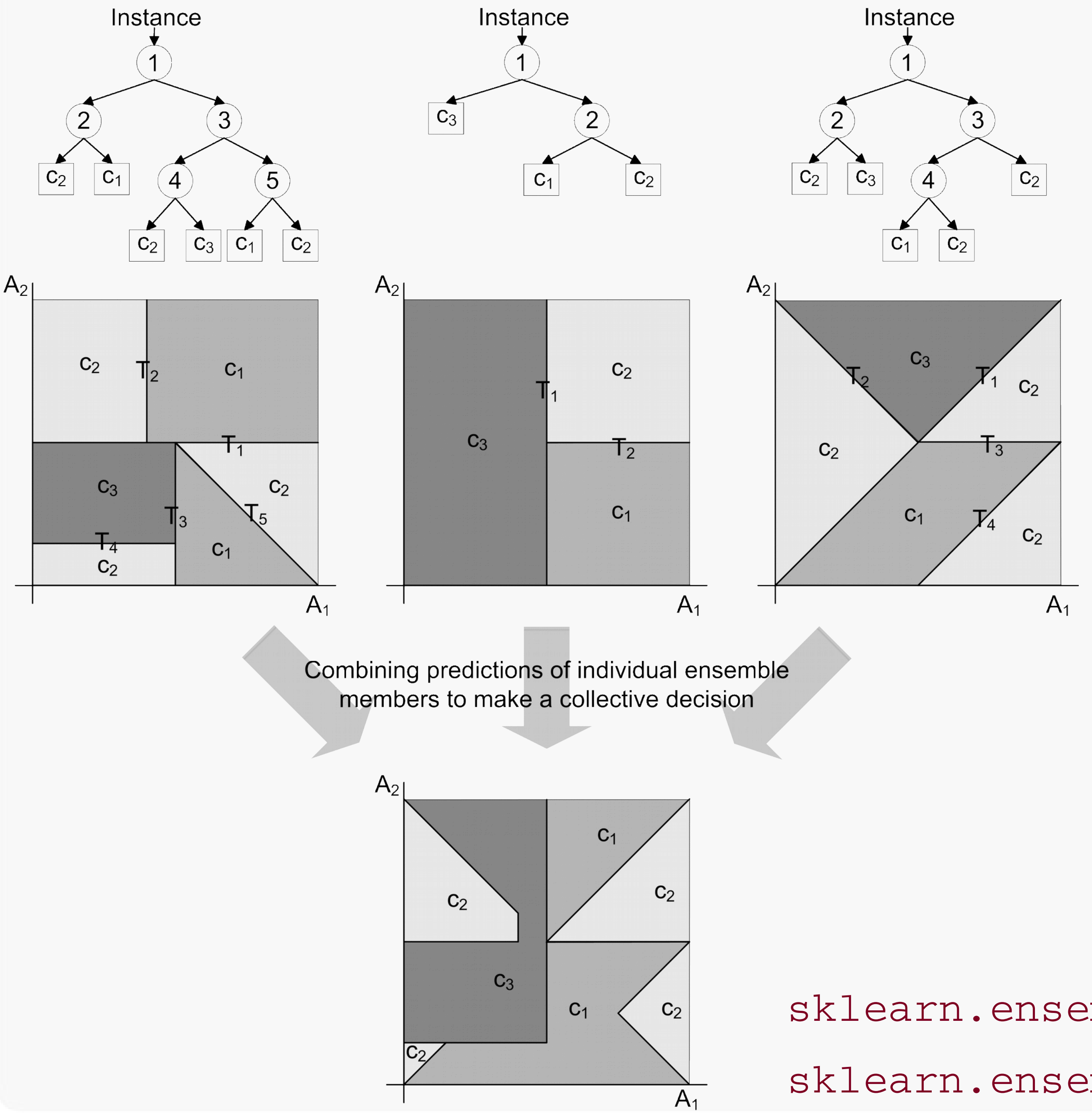
- At each leaf node, decide which is the best feature to split the data on (such that it separates best between classes), and what's the best split value of that feature
- To make this choice, use either the entropy or the Gini impurity (see extra slides)
- Making a prediction is simple, just propagate the features through the tree as a series of yes/no decisions

Decision Trees



Decision trees, while conceptually easy to understand and implement, tend to produce overcomplicated trees that overfit the data.

Ensemble Methods



Ensemble methods combine weak classifiers to create a robust classifier.

`sklearn.ensemble.RandomForestClassifier`

`sklearn.ensemble.AdaBoostClassifier`

Ensemble Methods with Decision Trees

Advantages

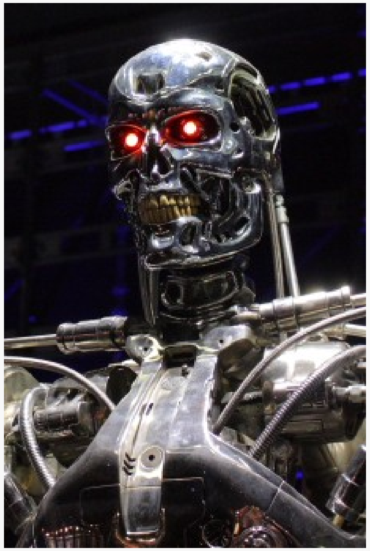
- Incredibly robust (low variance)
- Able to handle mixed feature types
- Robust to high dimensionality
- Able to naturally rank feature importance
- Random Forests is Michelle's favourite algorithm

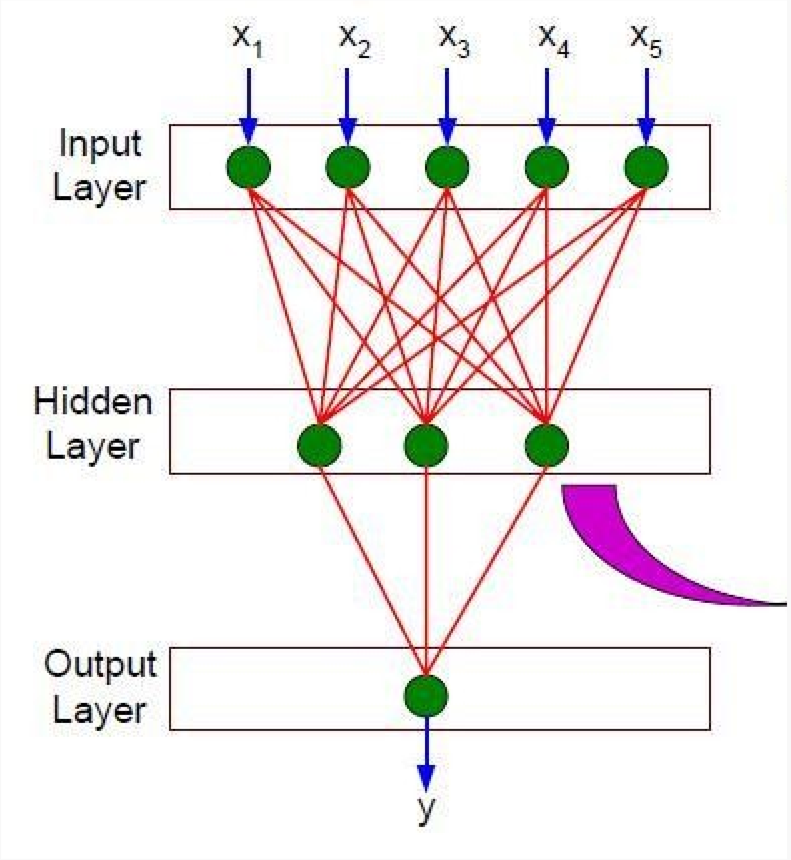
Disadvantages

- Computationally expensive
- Lots of hyperparameters to tweak

Artificial Neural Networks

Based on how the human brain learns (probably), ANNs are constructed from layers of connected neurons.



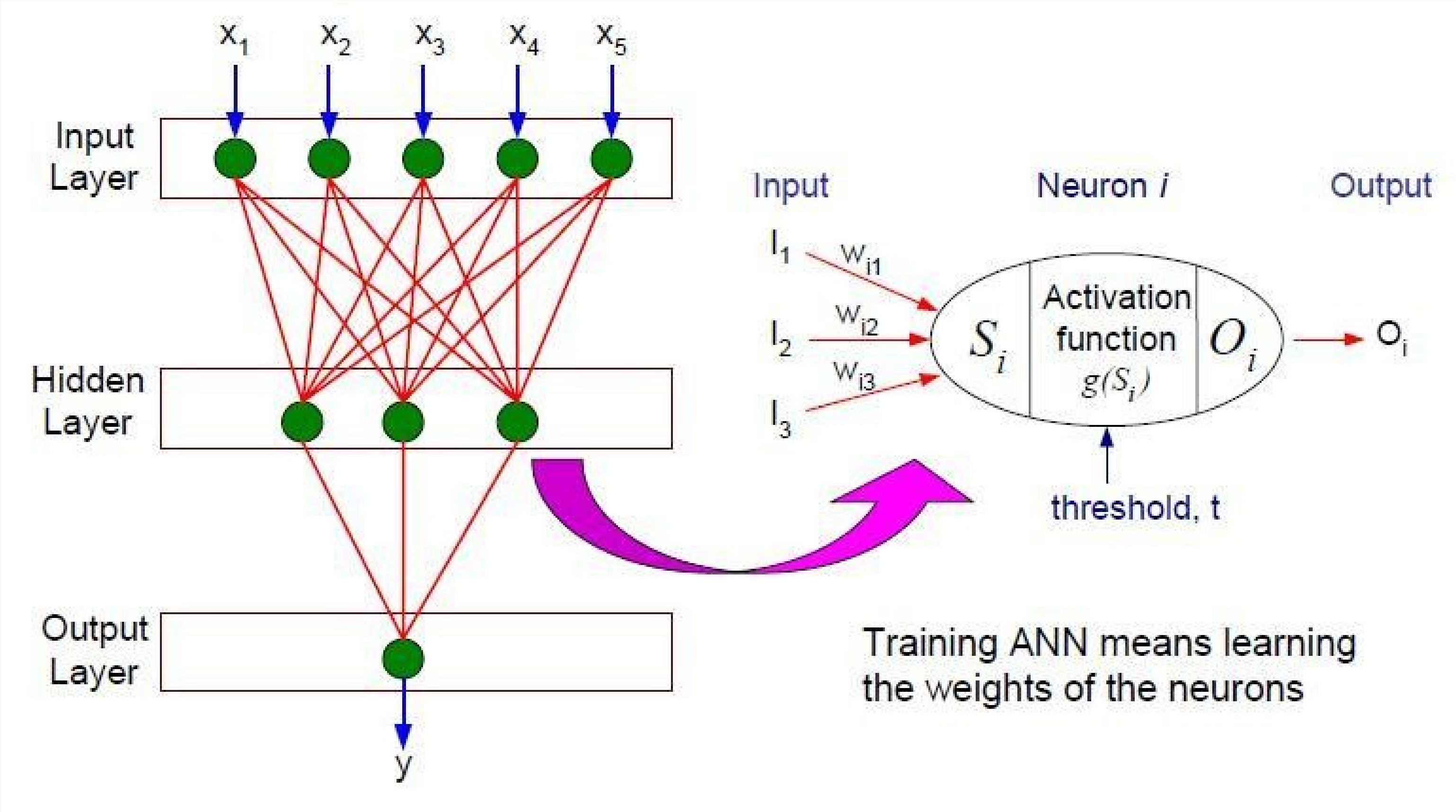


`sklearn.neural_network.multilayer_perceptron`

Figure: http://mines.humanoriented.com/

Artificial Neural Networks

Based on how the human brain learns (probably), ANNs are constructed from layers of connected neurons.



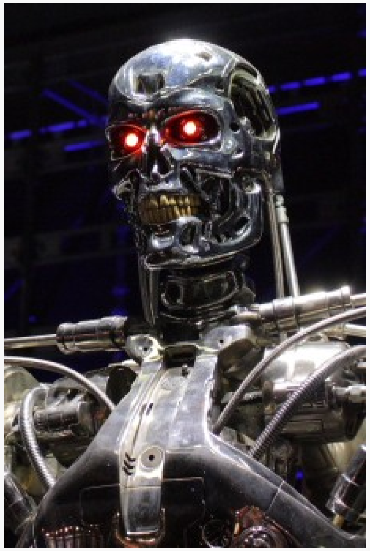


Figure: http://mines.humanoriented.com/

Artificial Neural Networks

Advantages

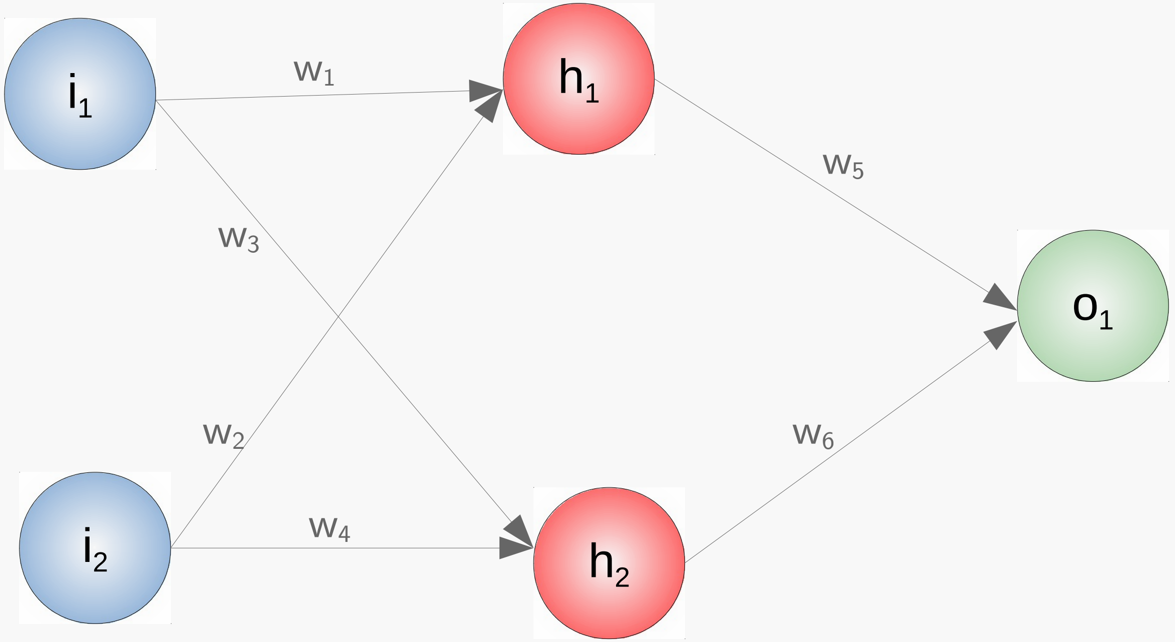
- Robust to high dimensionality
- One of few algorithms with a Bayesian interpretation (see Bishop or papers by MacKay)
- Highly non-linear, works well for many problems

Disadvantages

- Computationally expensive
- Complex networks require large amounts of training data
- Can be difficult to interpret (black boxy)

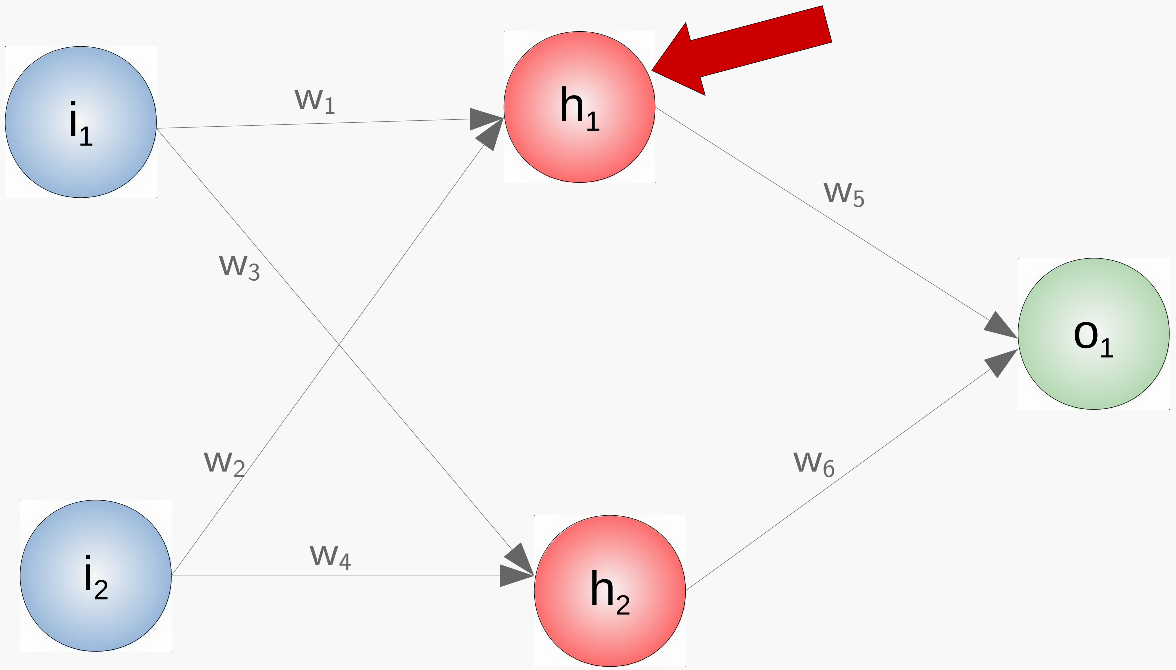
Question

- Write down the output in terms of the inputs and weights of this network, assuming a tanh activation function



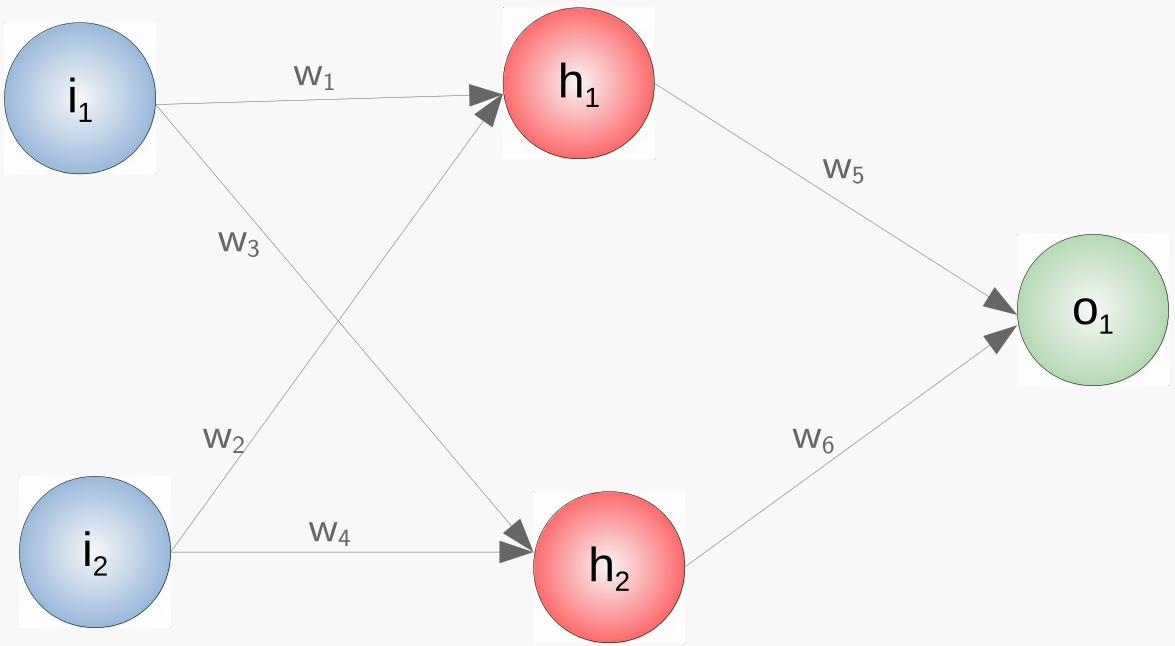
Question

- Write down the output in terms of the inputs and weights of this network, assuming a tanh activation function



Question

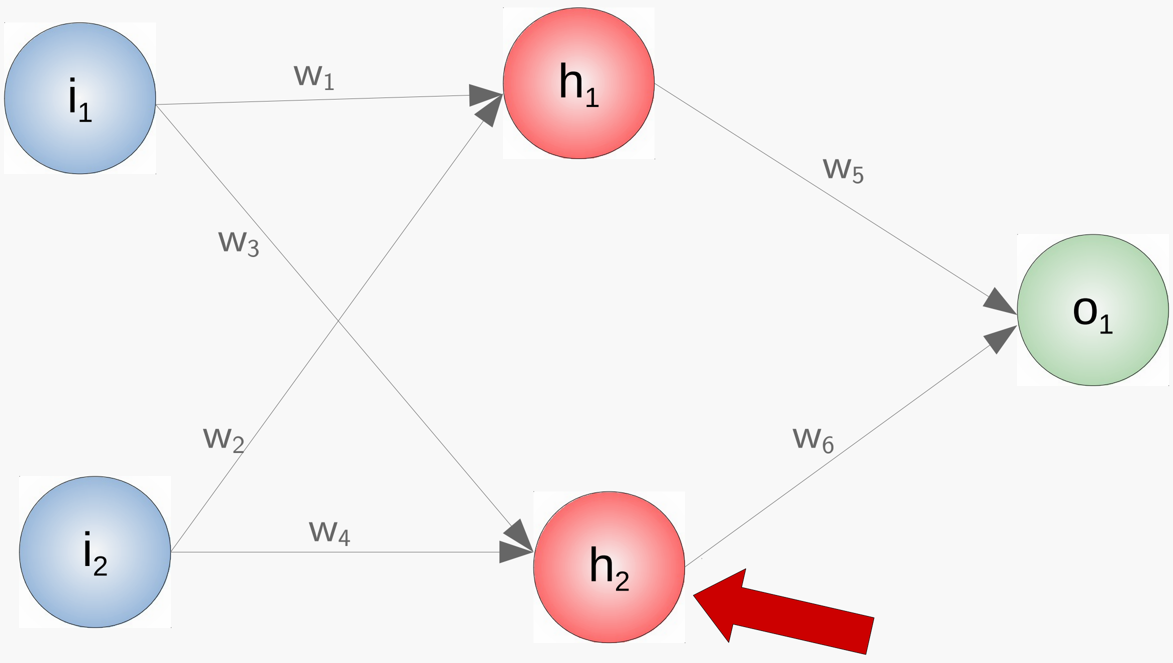
- Write down the output in terms of the inputs and weights of this network, assuming a tanh activation function



$$\tanh(w_1 i_1 + w_2 i_2)$$

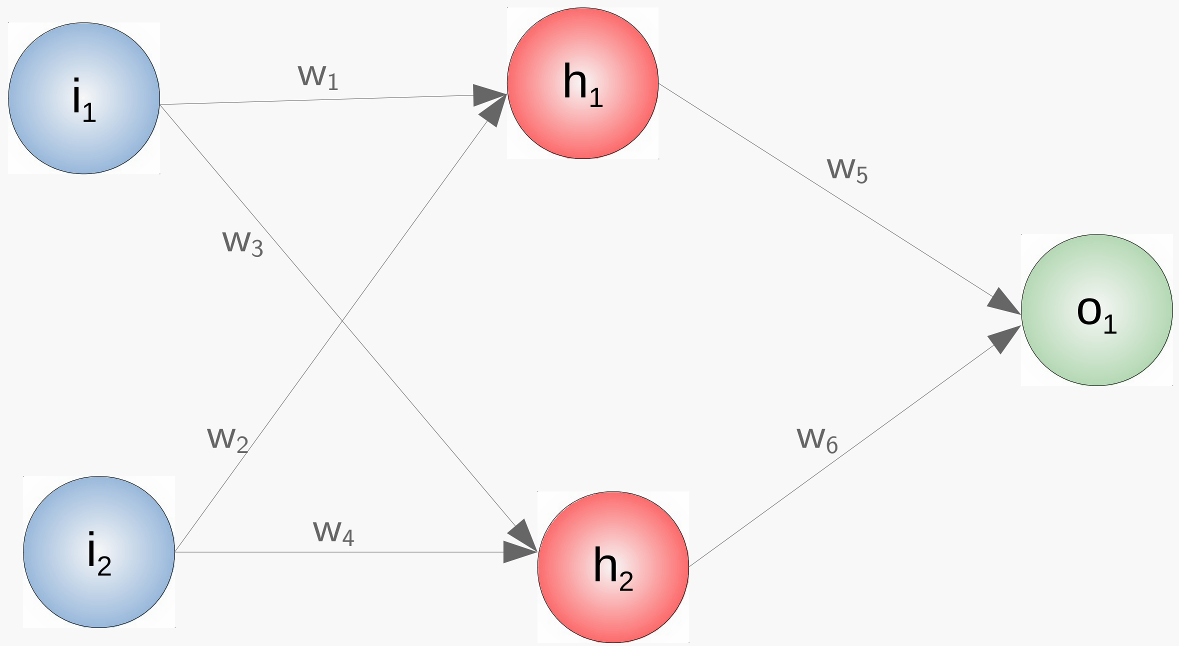
Question

- Write down the output in terms of the inputs and weights of this network, assuming a tanh activation function



Question

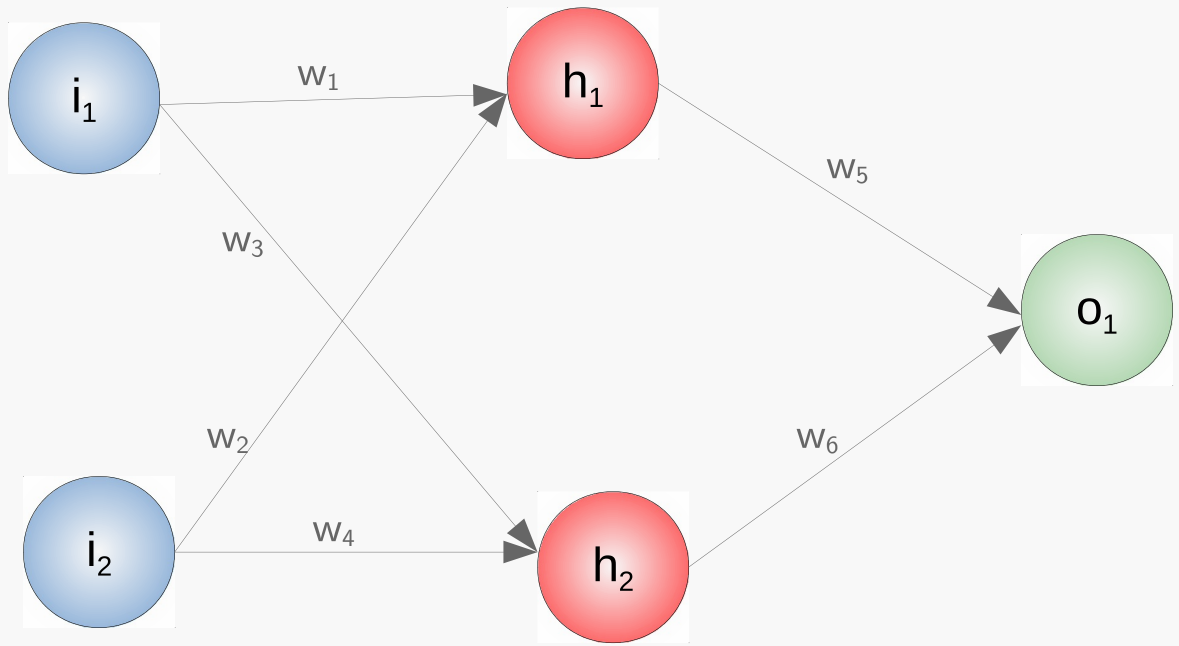
- Write down the output in terms of the inputs and weights of this network, assuming a tanh activation function



$$\tanh(w_3 i_1 + w_4 i_2)$$

Question

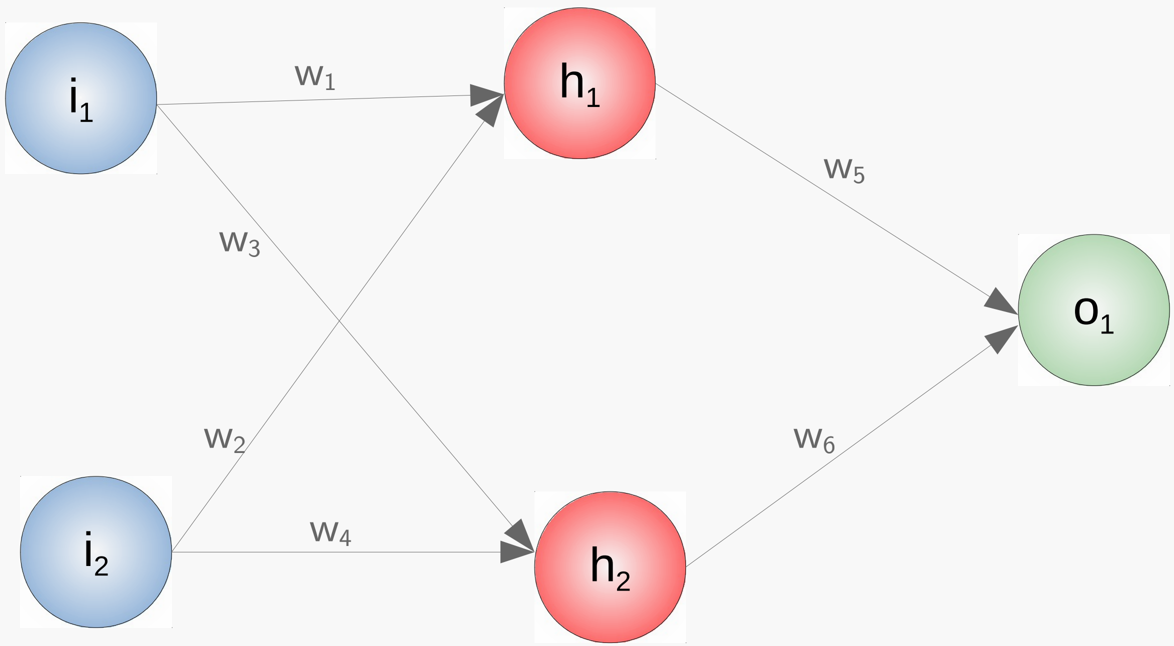
- Write down the output in terms of the inputs and weights of this network, assuming a tanh activation function



$$[w_5 \tanh(w_1 i_1 + w_2 i_2) + w_6 \tanh(w_3 i_1 + w_4 i_2))]$$

Question

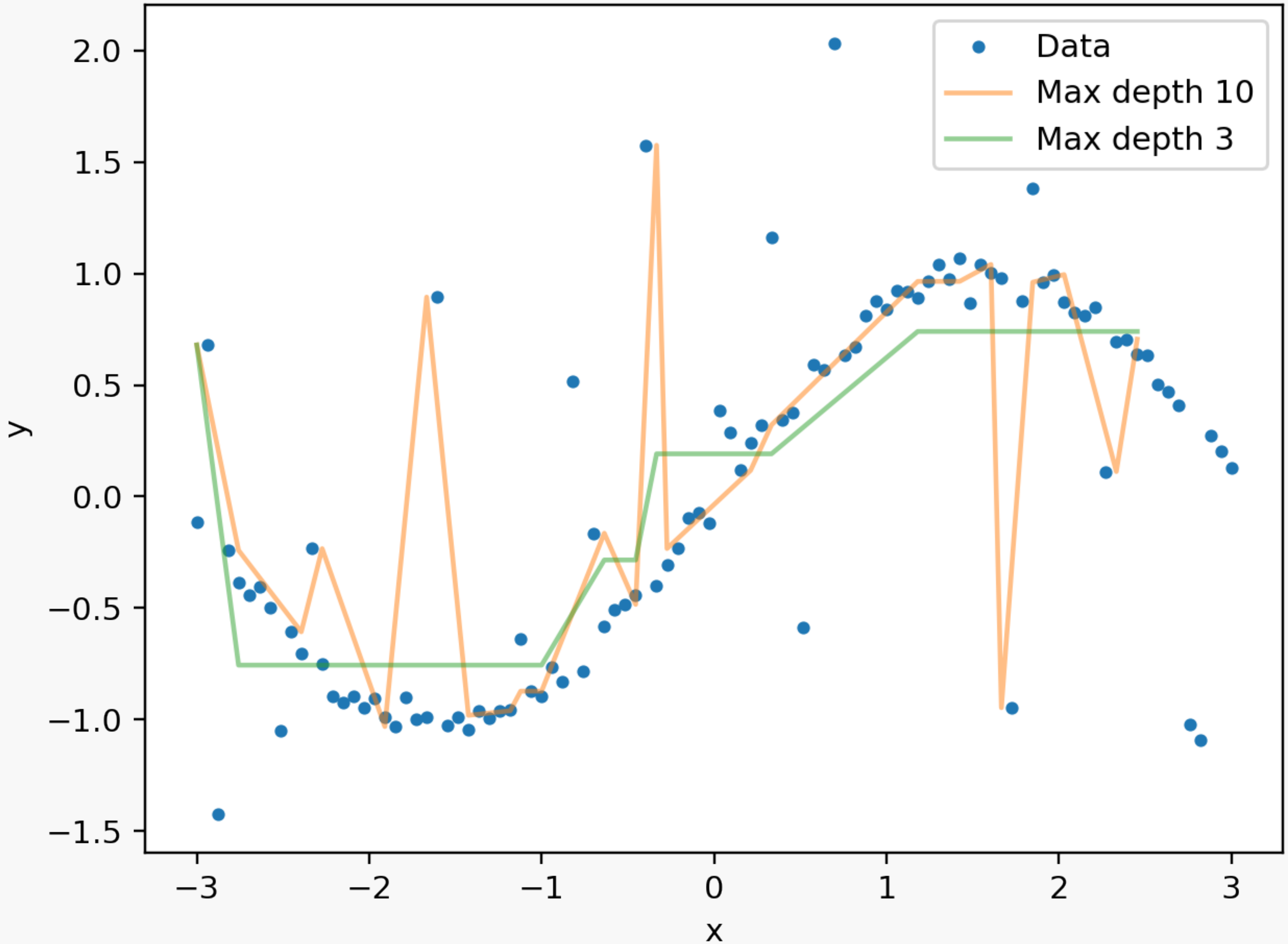
- Write down the output in terms of the inputs and weights of this network, assuming a tanh activation function



$$\tanh\left[w_5 \tanh(w_1 i_1 + w_2 i_2) + w_6 \tanh(w_3 i_1 + w_4 i_2))\right]$$

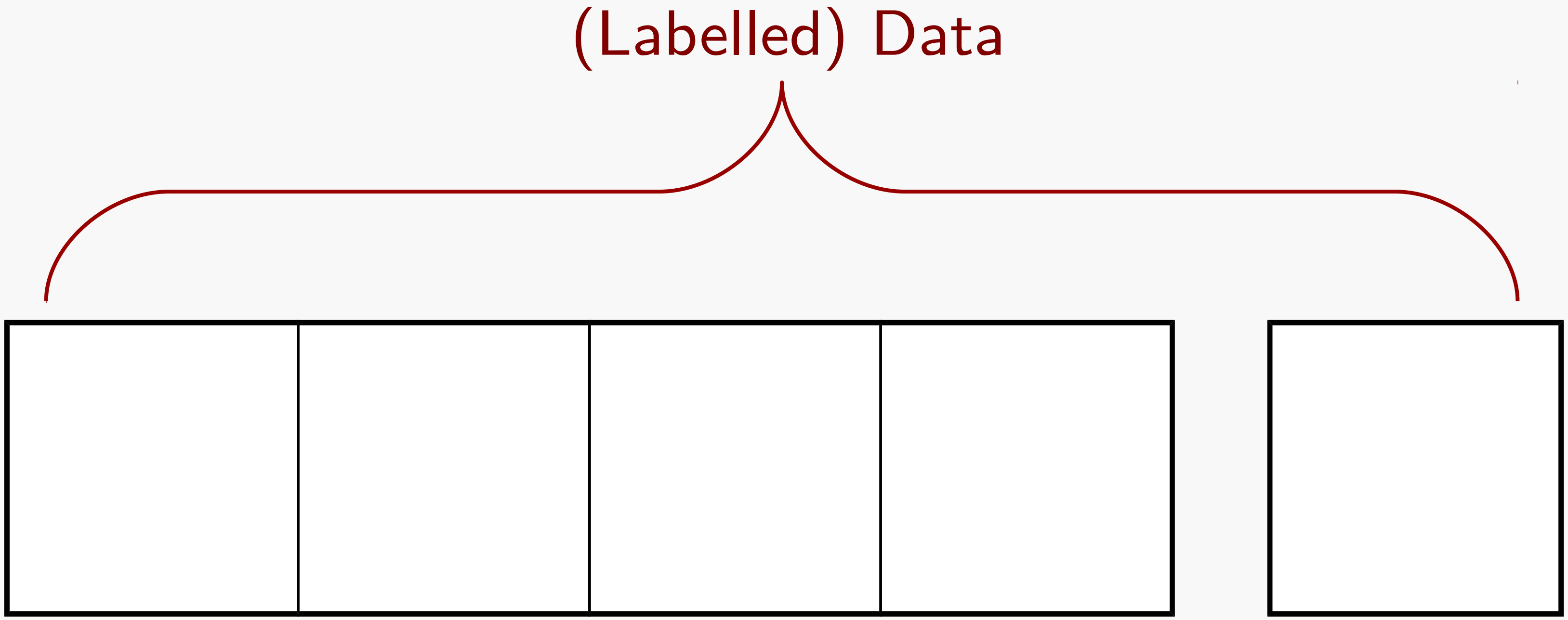
Overfitting

Overfitting

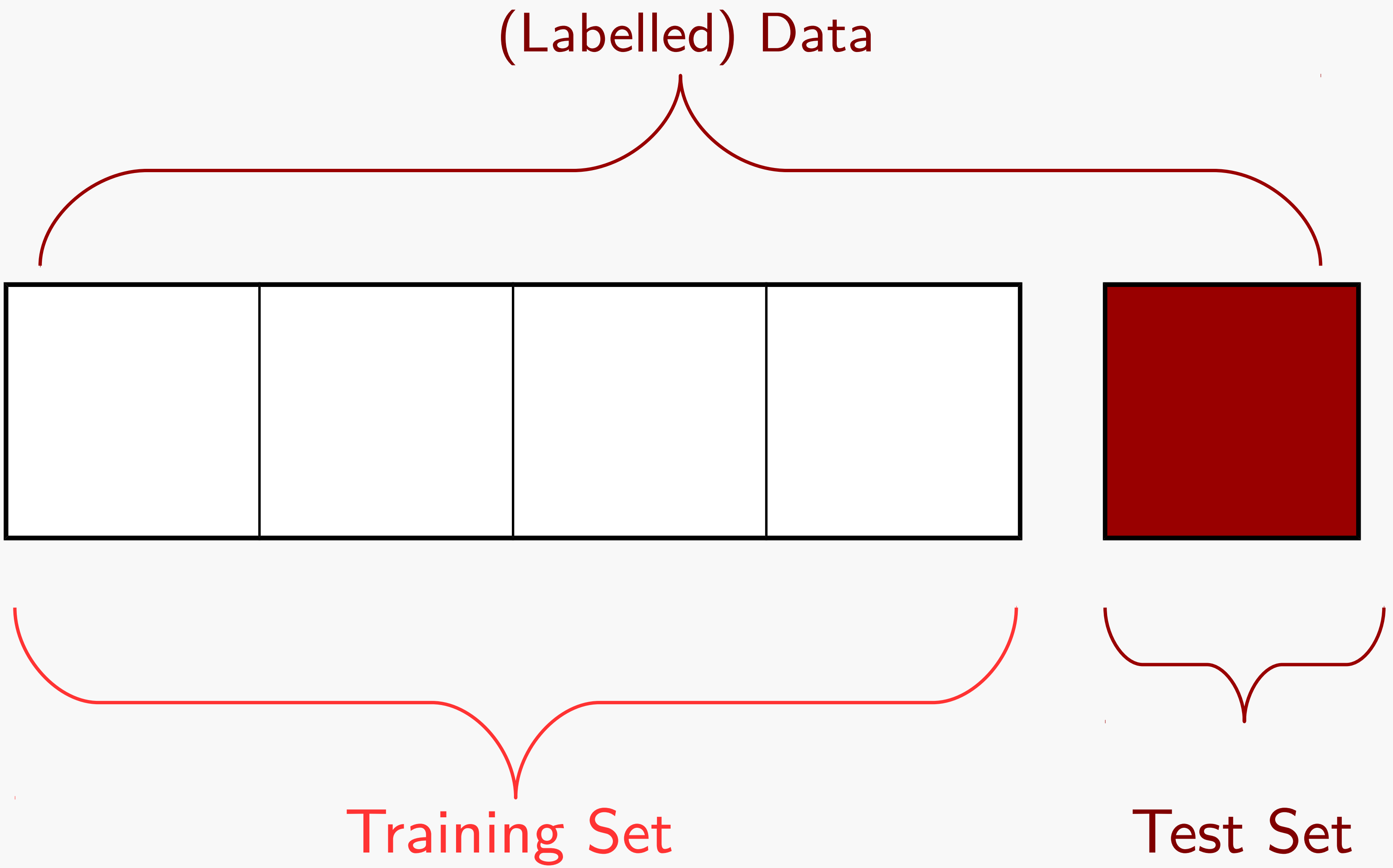


Overfitting example using Decision Trees

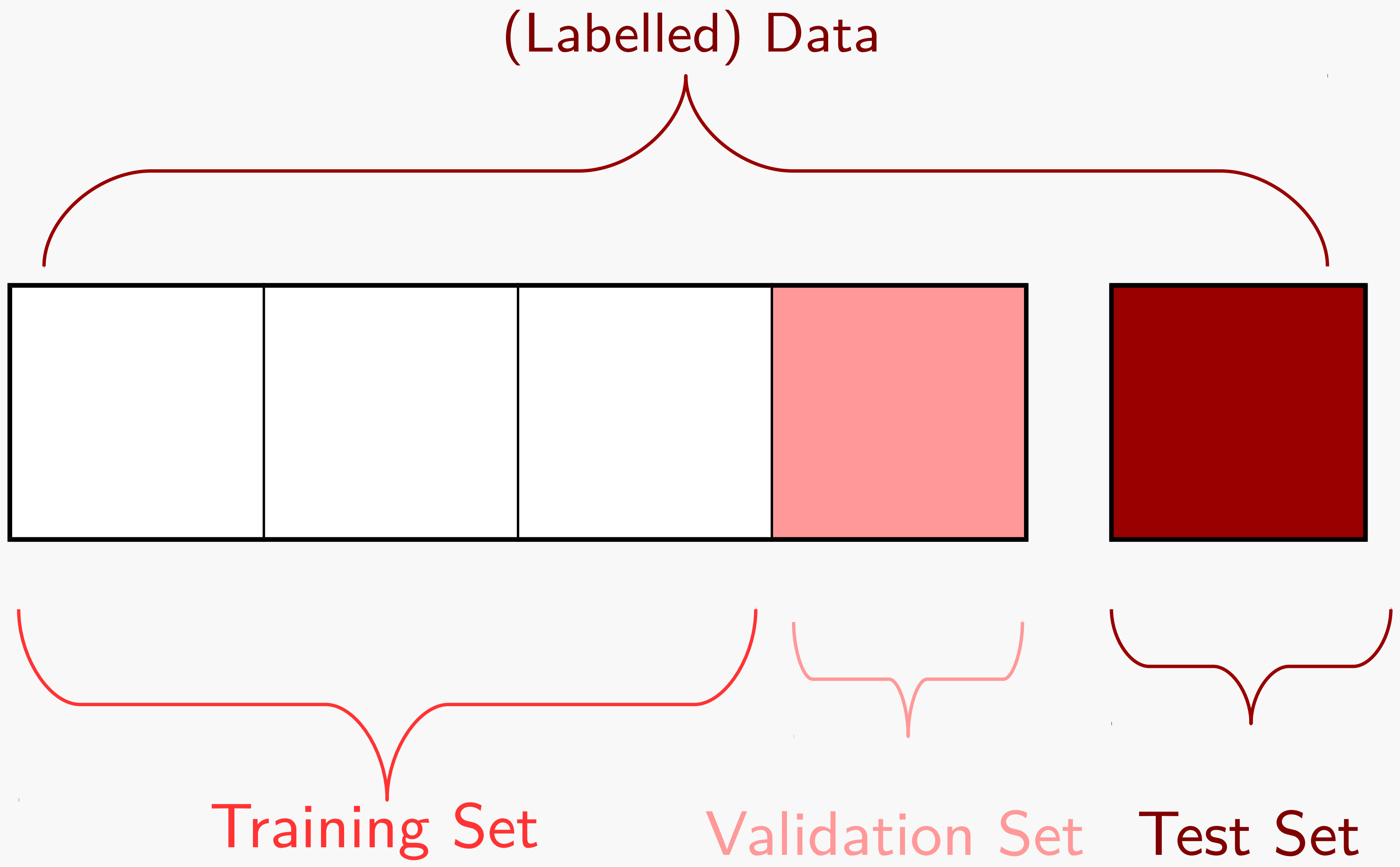
Training, Validation and Testing



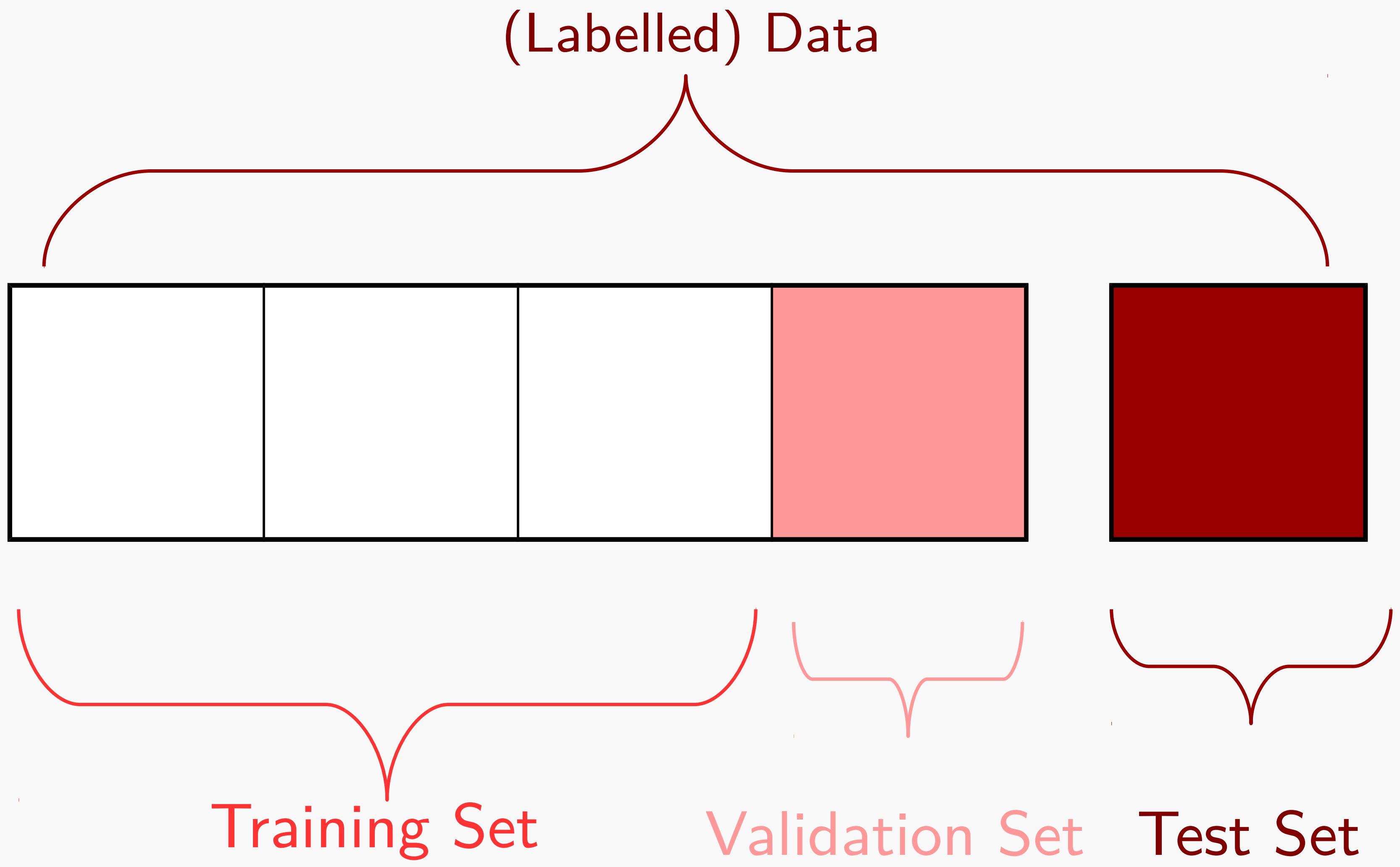
Training, Validation and Testing



Training, Validation and Testing



Training, Validation and Testing



`sklearn.model_selection.train_test_split`

Cross validation

- Overfitting is very bad
- Split data into three: training, validation and test
- Use cross validation to select hyperparameters without overfitting

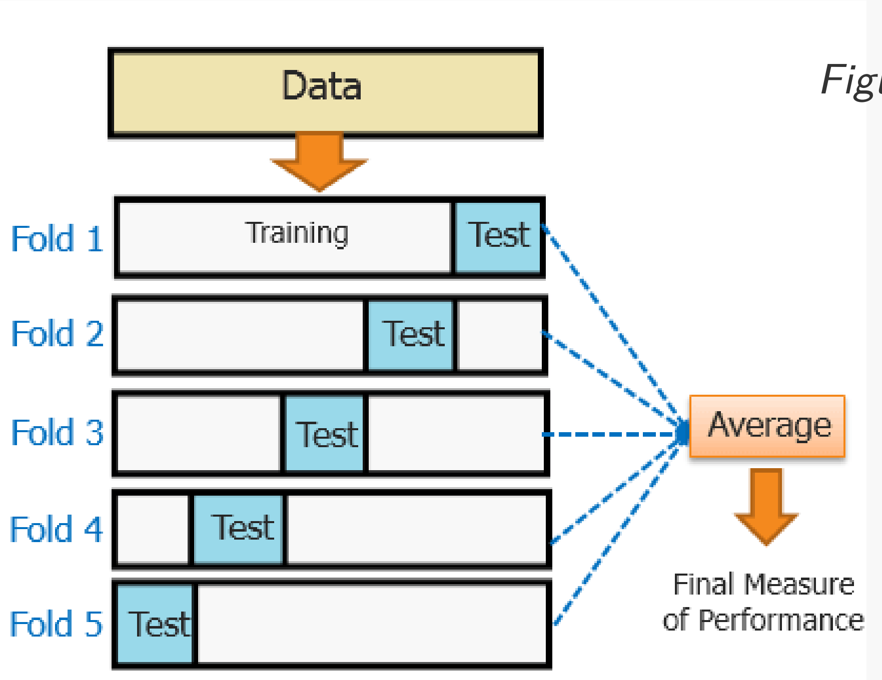
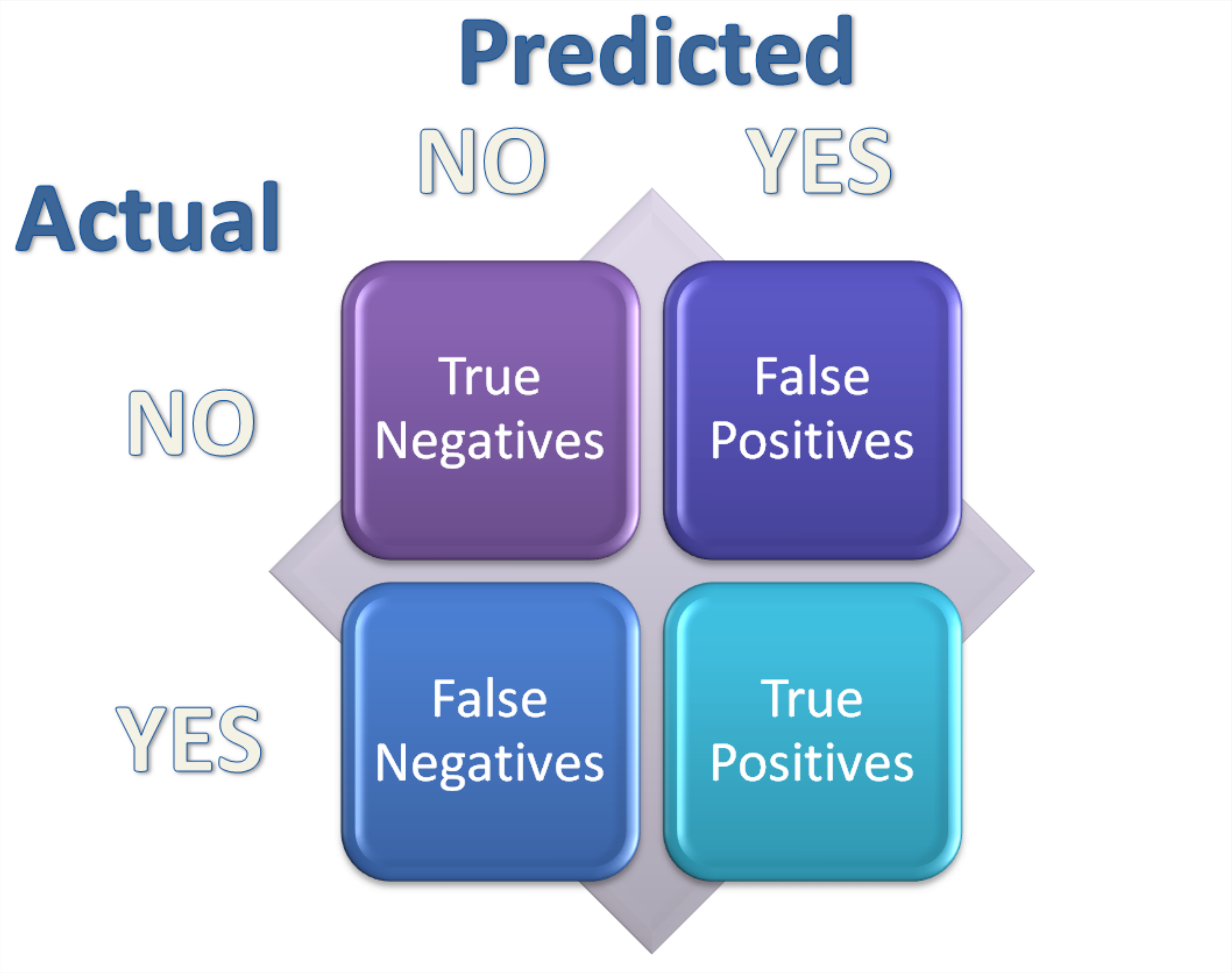


Figure: www.edureka.in/data-science

`sklearn.model_selection.GridSearchCV`

Evaluating algorithms

Evaluating algorithms



`sklearn.metrics.confusion_matrix`

Questions

- What's the best way to evaluate how well your algorithm has classified a set of objects?

Questions

- What's the best way to evaluate how well your algorithm has classified a set of objects?
- A common choice is precision:

$$precision = TP/P$$

Questions

- What's the best way to evaluate how well your algorithm has classified a set of objects?
- A common choice is precision:

$$precision = TP/P$$

- What's wrong with this?

Questions

- What's the best way to evaluate how well your algorithm has classified a set of objects?
- A common choice is precision:

$$\mathit{precision} = TP/P$$

- What's wrong with this?
- What about accuracy:

$$\mathit{accuracy} = (TP + TN)/\ (P{+}N)$$

Questions

- What's the best way to evaluate how well your algorithm has classified a set of objects?
- A common choice is precision:

$$precision = TP/P$$

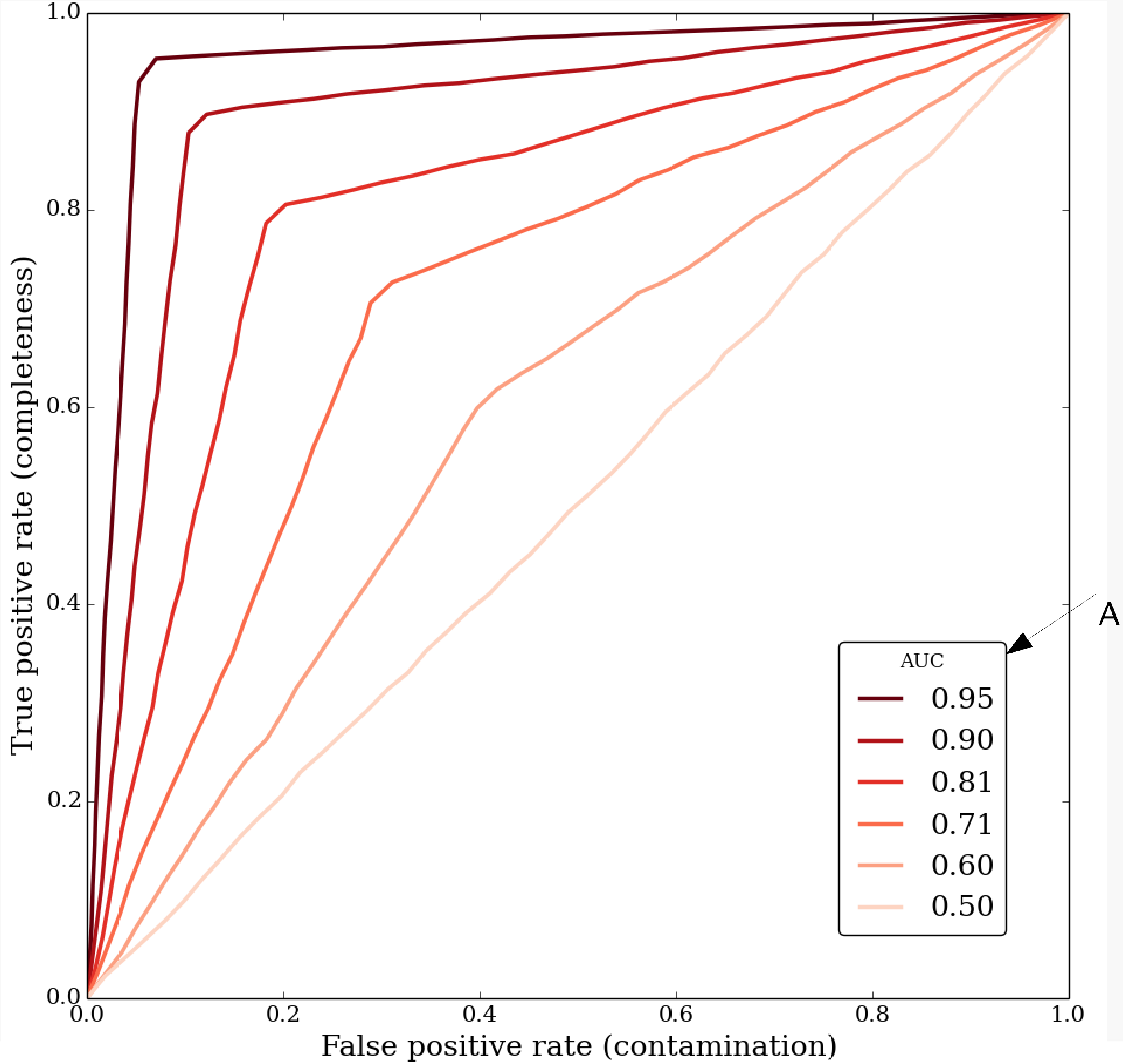
- What's wrong with this?
- What about accuracy:

$$accuracy = (TP + TN)/\ (P{+}N)$$

- What's wrong with this?

Receiver Operator Characteristic (ROC) curves

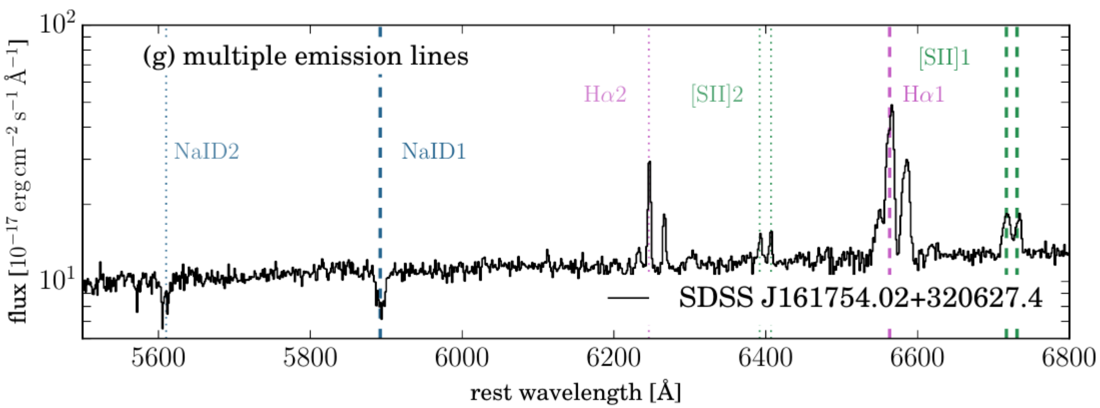
`sklearn.metrics.roc_curve`

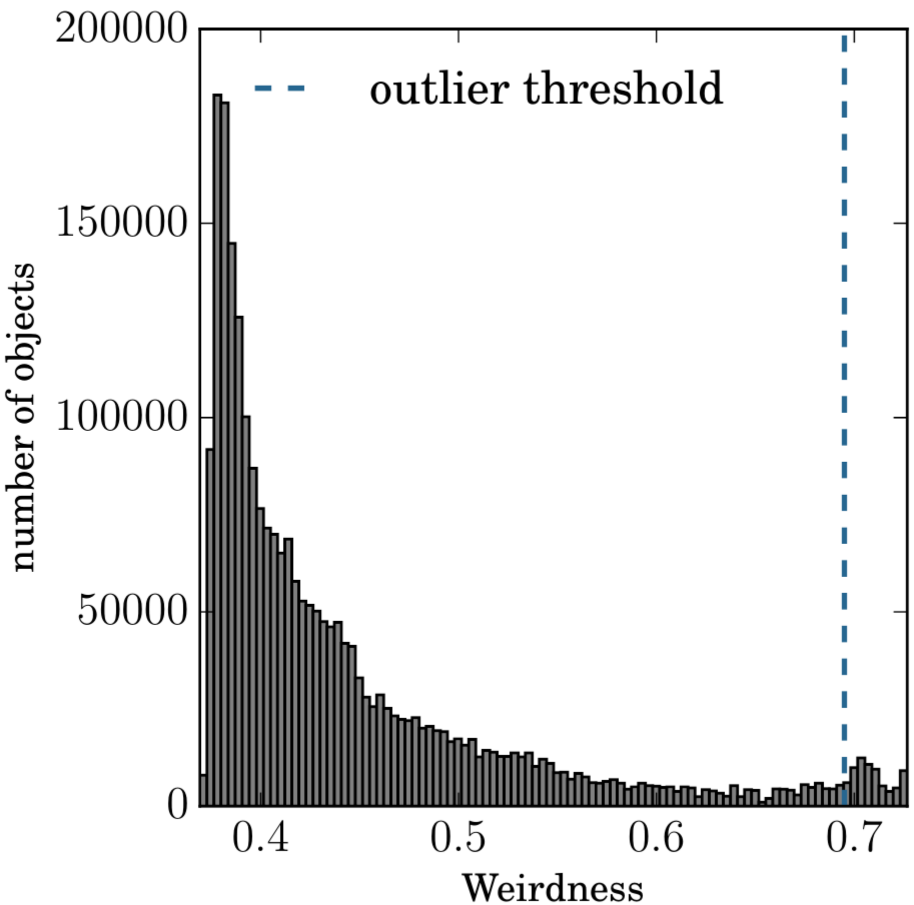


AUC = Area Under Curve

Examples of Machine Learning

The weirdest SDSS galaxies: results from an outlier detection algorithm

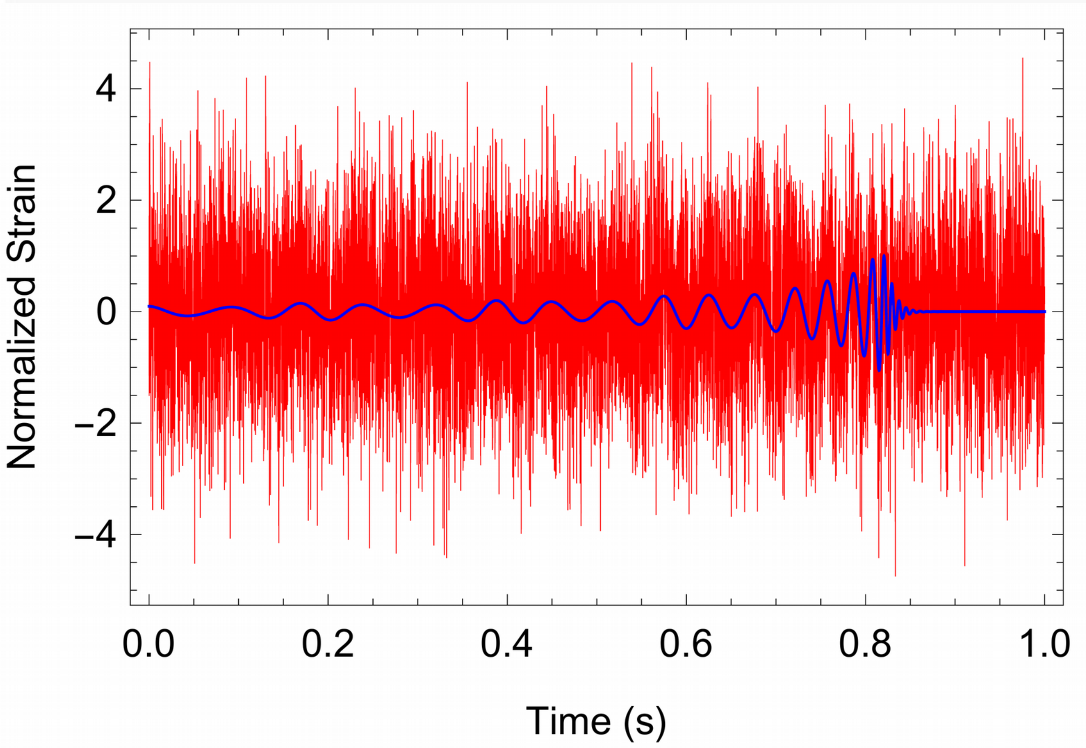


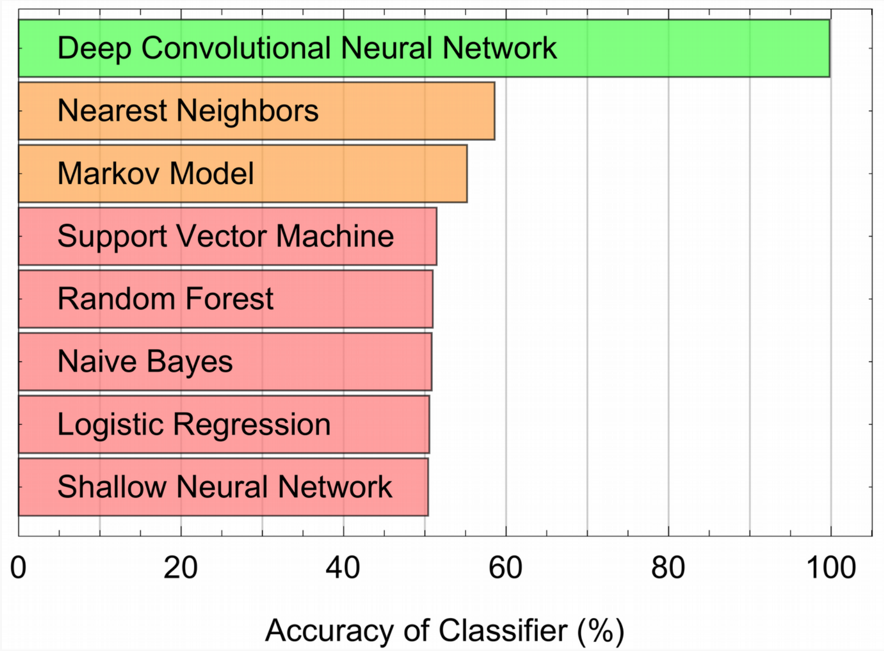


Baron & Poznanski (2016) http://arxiv.org/abs/1611.07526v1

Examples of Machine Learning

Deep Neural Networks to Enable Real-time Multimessenger Astrophysics

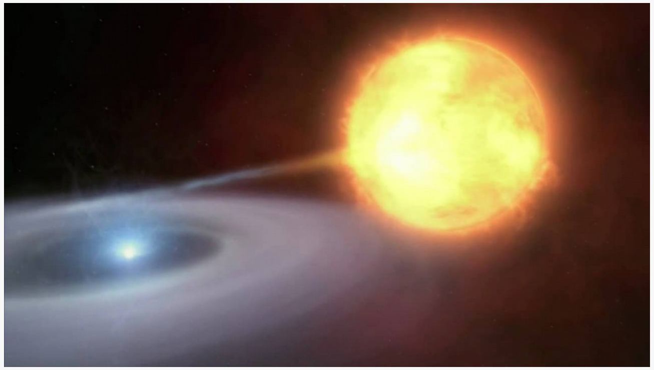




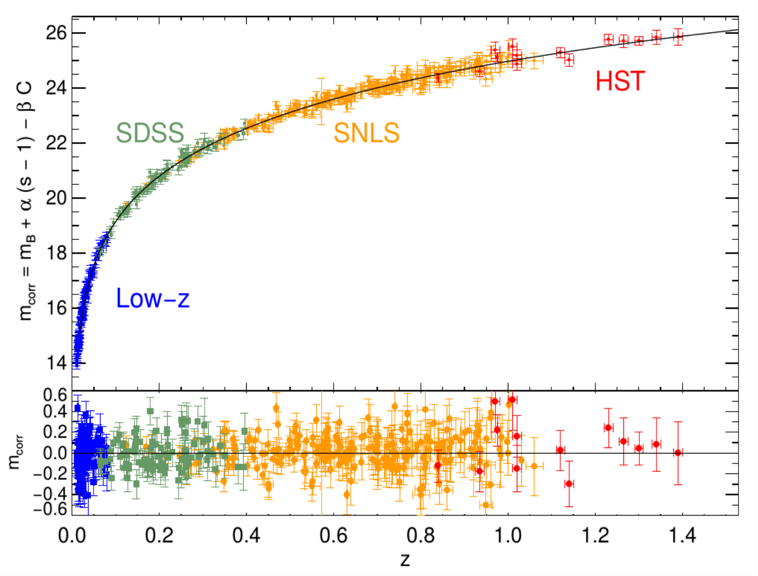
George et al. (2017) http://arxiv.org/abs/1701.00008v1

Photometric Classification of Supernovae

Type Ia supernovae

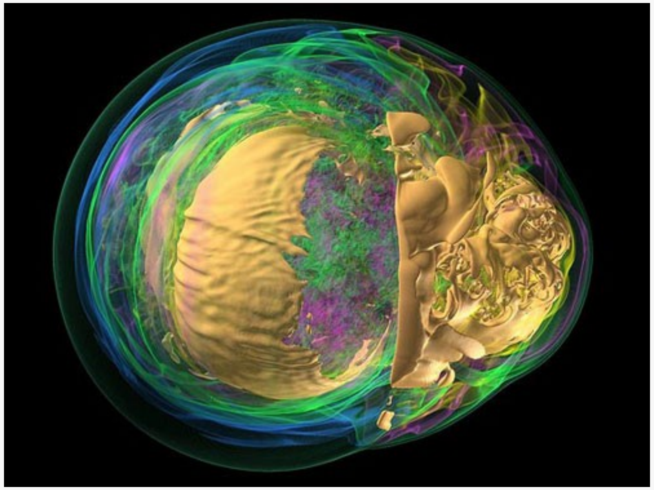


Cosmology



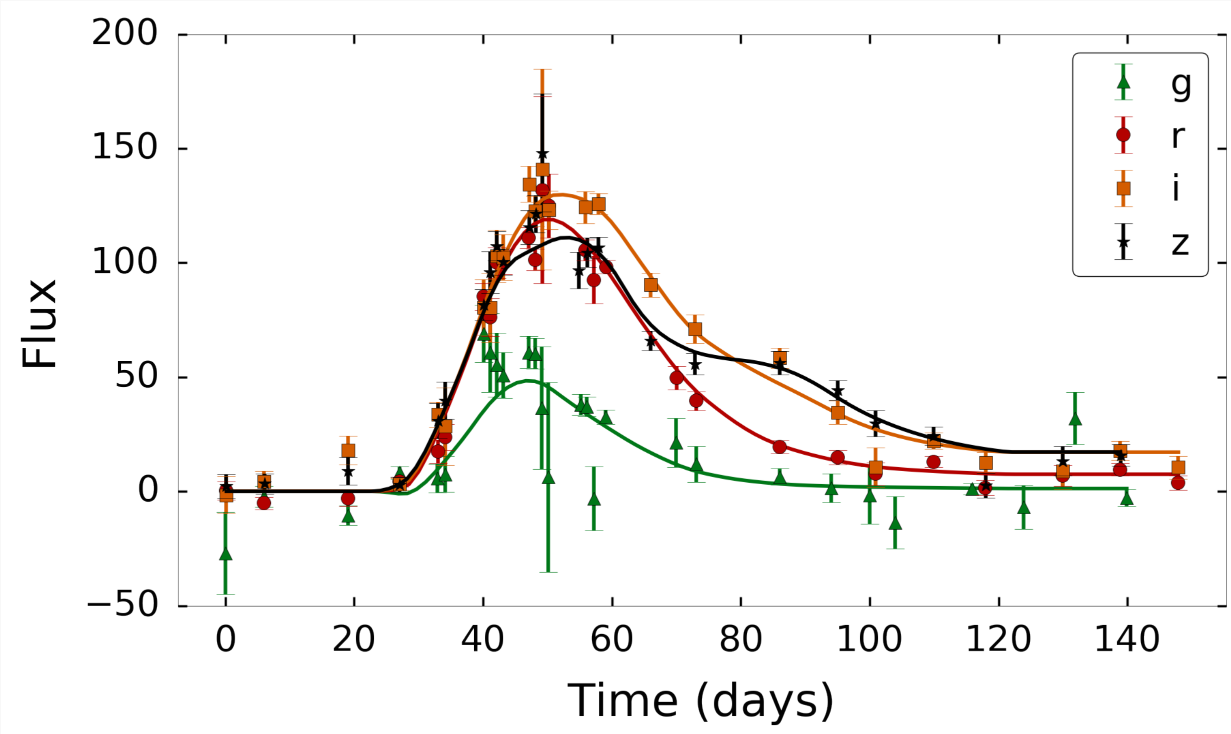
Core collapse supernovae

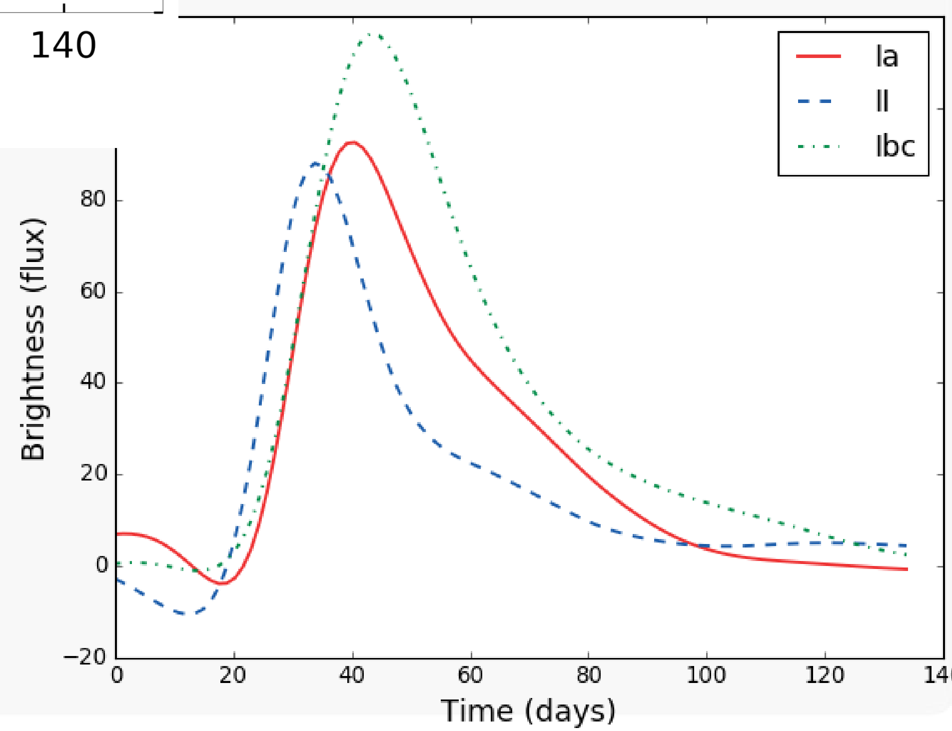
Supernova astrophysics



Conley et al. 2011

Photometric Classification of Supernovae

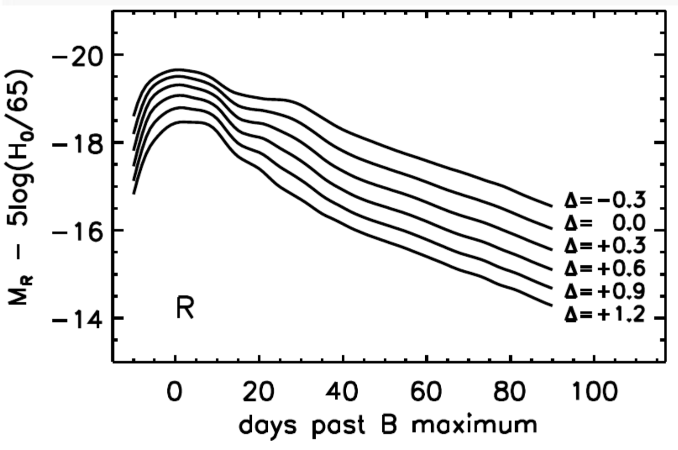


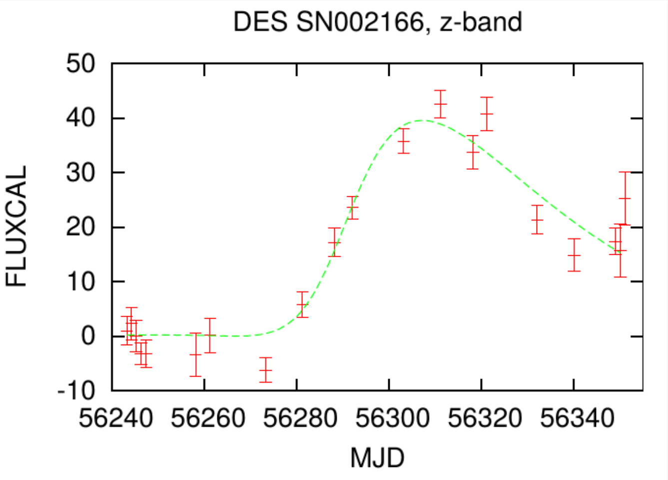


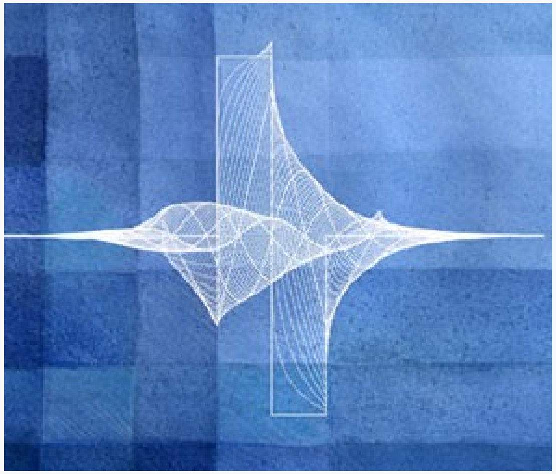
Lochner et al. (2016) 1603.00882

Feature selection

We've identified three promising approaches:







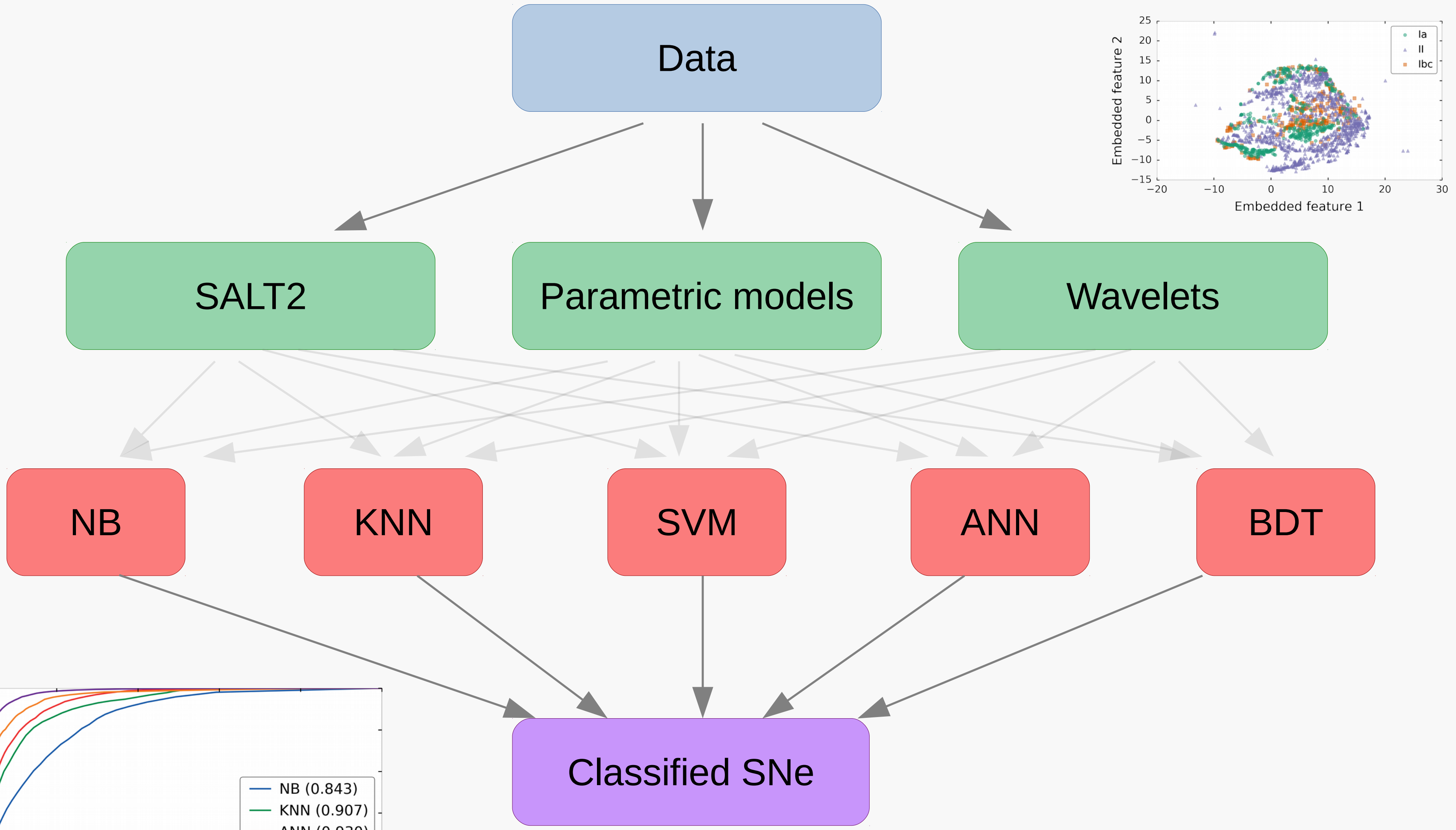
1) Template fitting

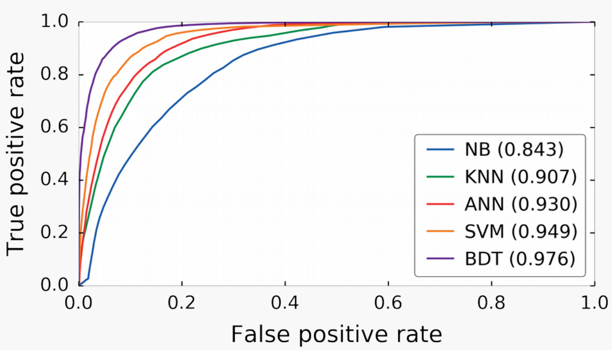
2) General light curve parameterisations

3) Wavelets

Model independence

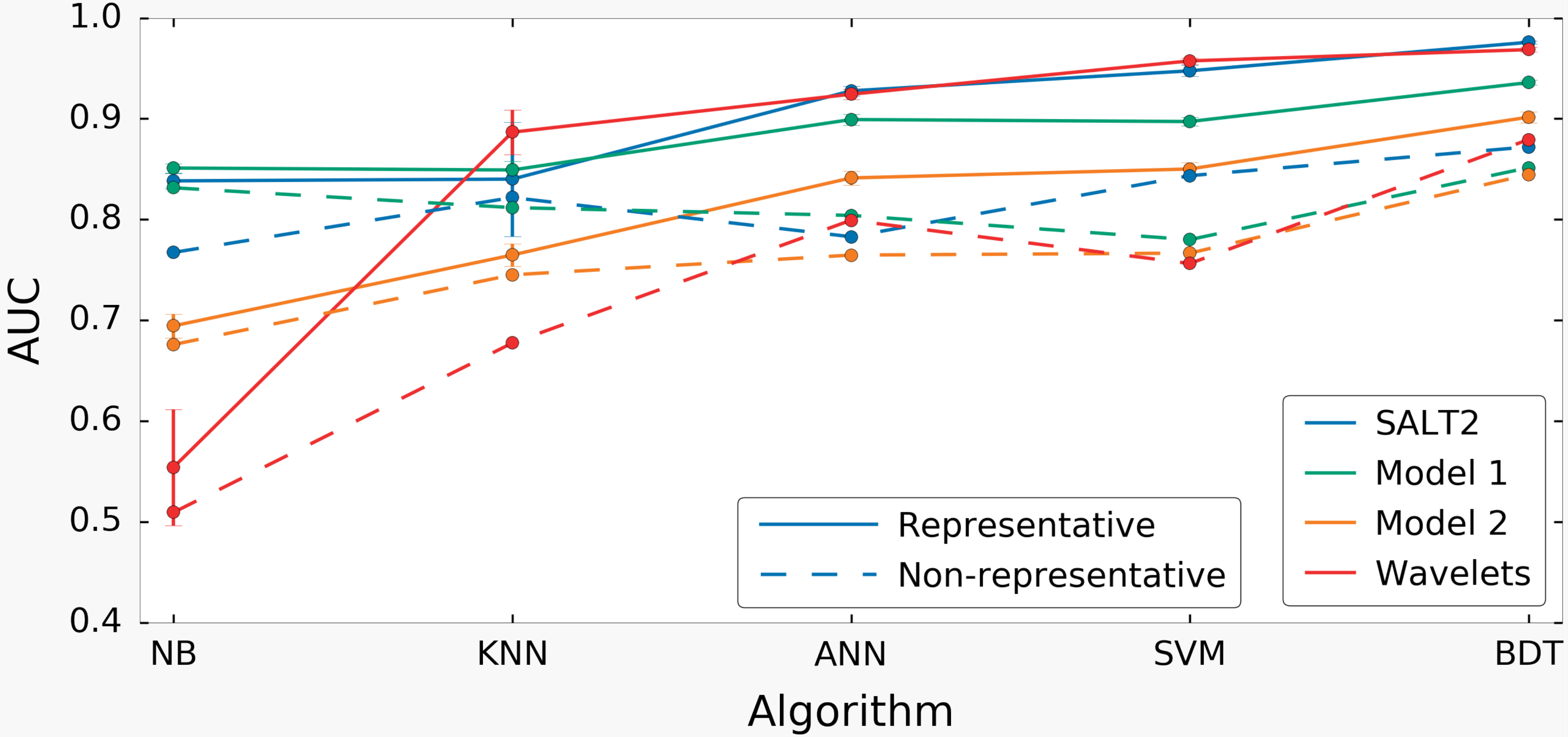
Pipeline





https://github.com/LSSTDESC/snmachine

Results



M. Lochner, J.D. McEwen, H.V. Peiris, O. Lahav & M.K. Winter (2016) 1603.00882

Question

What is the difference between machine learning and statistics?

Machine Learning vs. Statistics

Decide whether machine learning or statistics is the appropriate approach to the following problems:

- Predicting the weather
- Removing foregrounds (e.g. in CMB, 21cm etc.)
- Testing for rare diseases (e.g. HIV test)
- Removing artifacts in astronomical images (e.g. CCD errors, aeroplanes etc.)
- Photometric redshift estimation

Read up on these BEFORE doing ML

- Feature rescaling
- Overfitting
- Hyperparameters and cross validation
- Dimensionality reduction
- Representativeness (N.B!)
- Performance metrics

Now you're ready to code!

https://github.com/MichelleLochner/ml-tutorial/

An Introduction to Machine Learning with Scikit-learn

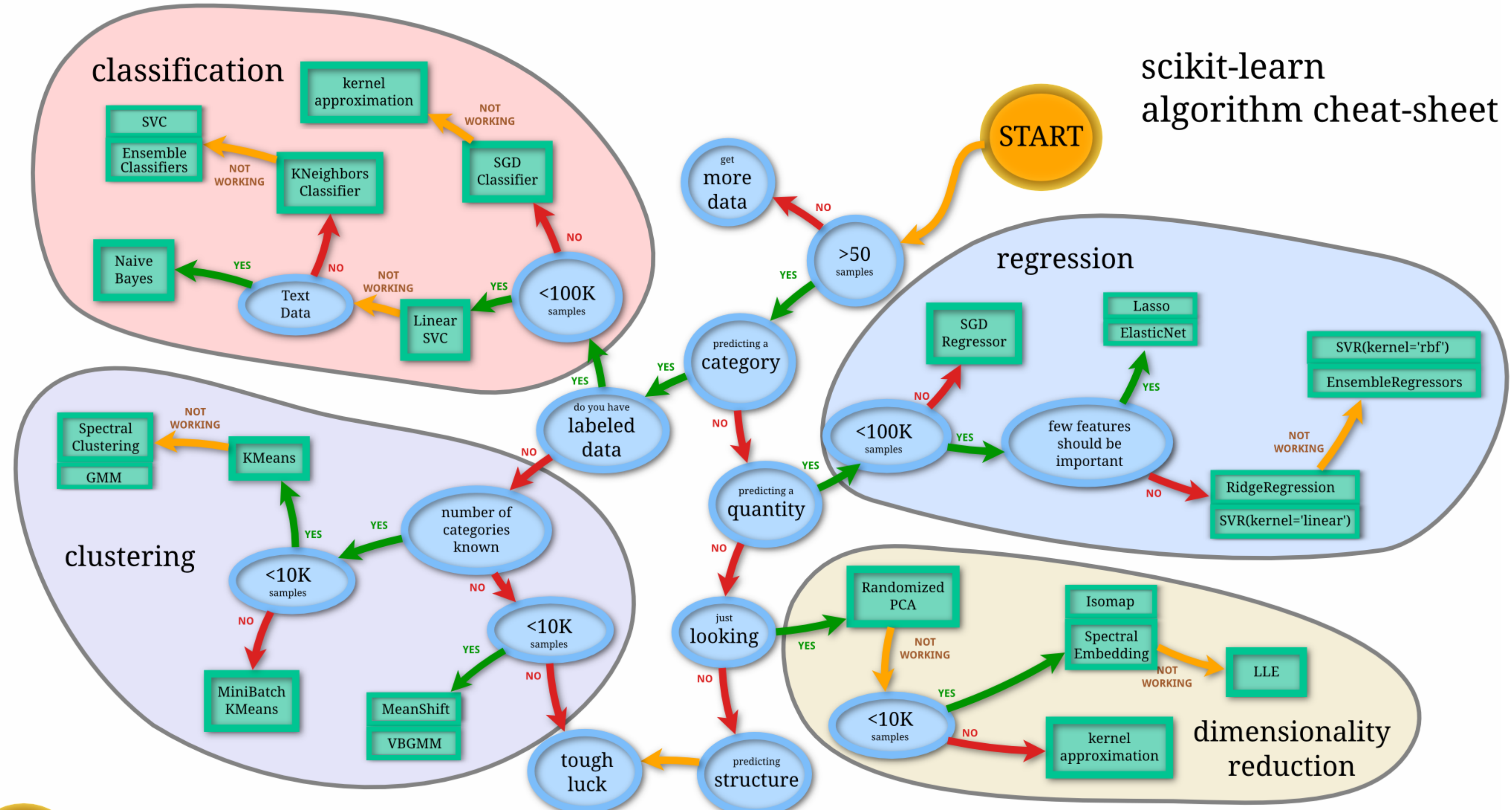
By Michelle Lochner

First we need to import some tools from scikit learn and the classifiers we're going to use

In [60]:
```
from __future__ import division, print_function
from sklearn.datasets import make_circles

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import accuracy_score
from sklearn.metrics import roc_curve, roc_auc_score

from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeRegressor

import numpy as np
import time
import matplotlib.pyplot as plt
%matplotlib nbagg
```

In [61]:
```
def plot_roc(fpr, tpr):
    """
    Simple ROC curve plotting function.

    Parameters
    ----------
    fpr : array
```

Tips and Tricks



References

https://www.coursera.org/learn/machine-learning

https://github.com/rasbt/python-machine-learning-book

https://github.com/jakevdp/sklearn_tutorial

http://ipython-books.github.io/featured-04/

Bishop, Pattern Recognition and Machine Learning, 2006

Lochner et al. (2016) http://arxiv.org/abs/1603.00882

Email me at:
dr.michelle.lochner@gmail.com

Extra slides

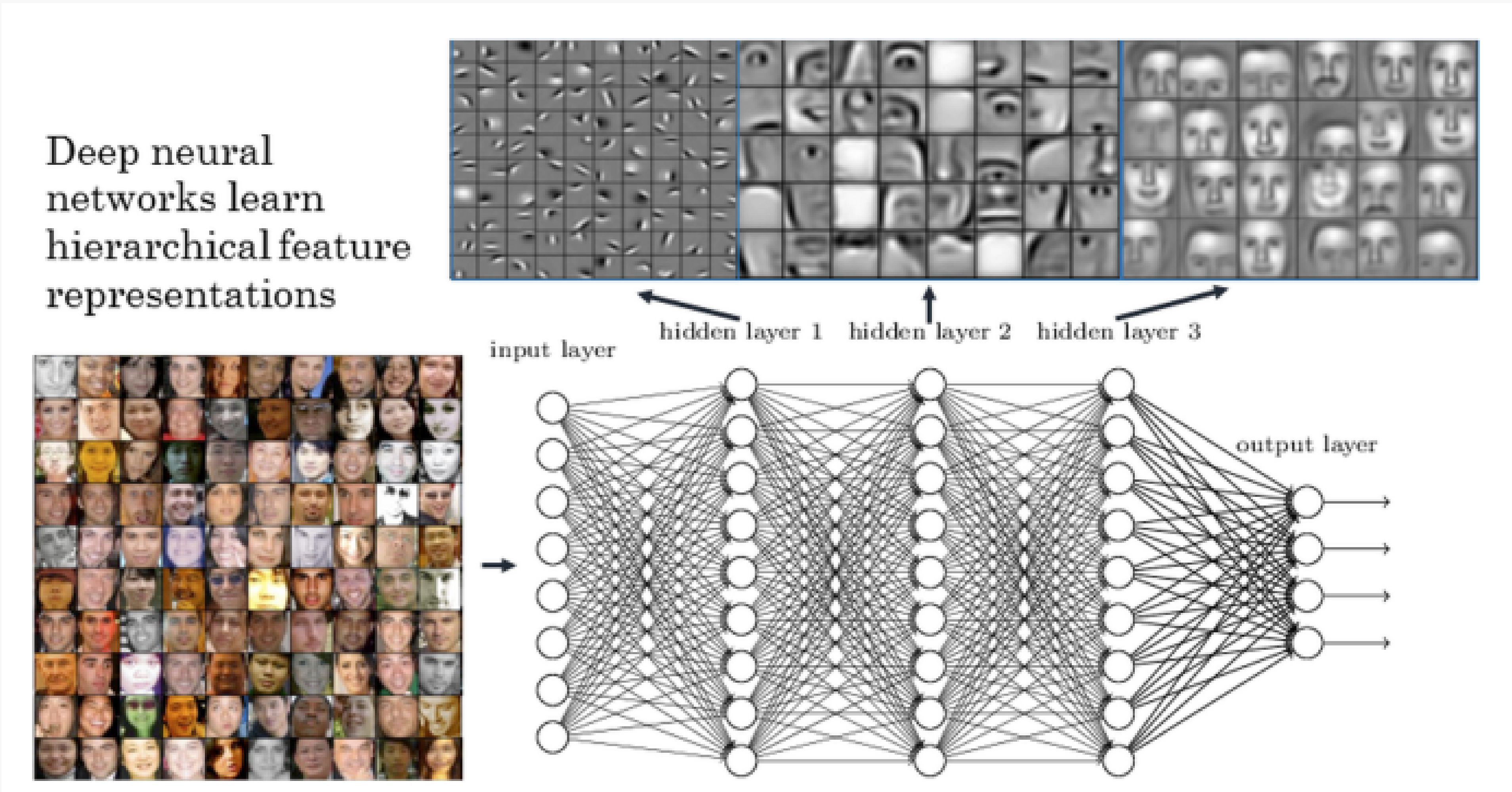
Going Deep

Deep learning has revolutionised machine learning in recent years, solving problems that have baffled computers for decades

https://github.com/AstroHackWeek/AstroHackWeek2015/blob/master/hacks/deep-learning/Deep%20Learning%20Example.ipynb

Going Deep

Convolutional neural networks for image classification



Going Deep

Convolutional neural networks for image classification, recurrent neural network for image descriptions

construction worker in orange safety vest is working on road.

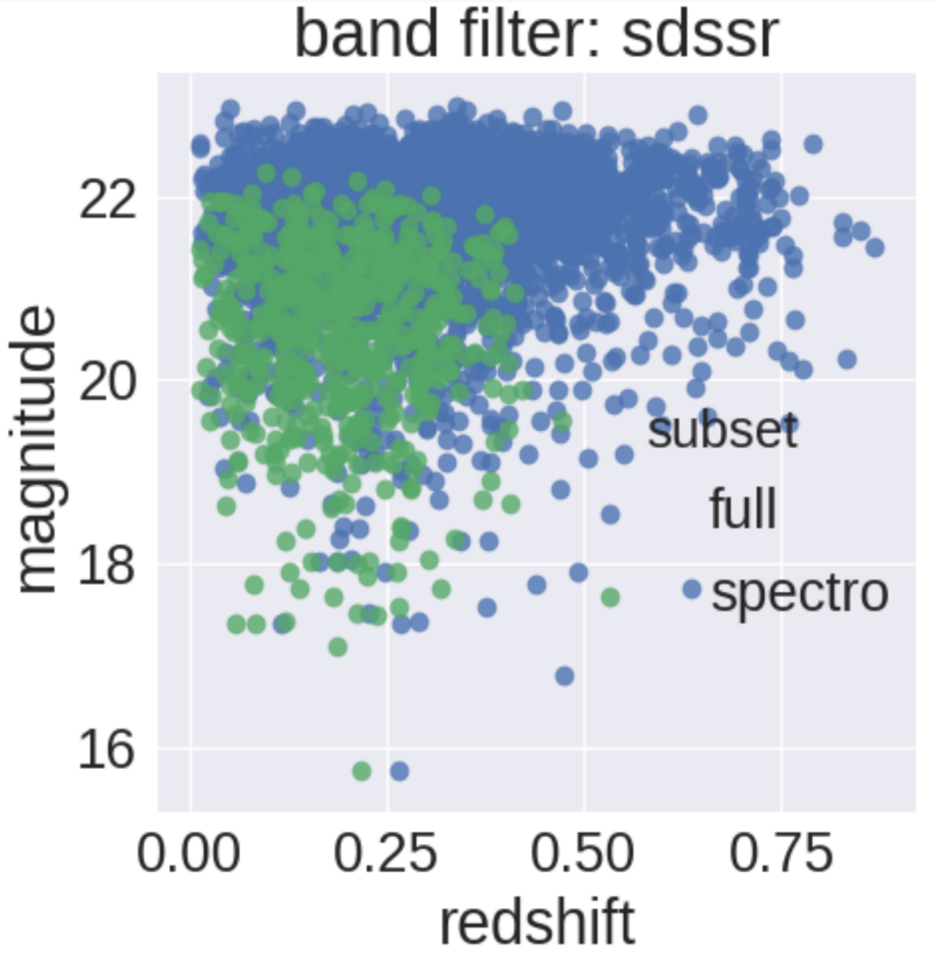
two young girls are playing with lego toy.

Karpathy & Fei-Fei (2015)

Going Deep

Alpha Go

Non-representative Data



type	# in spectro	fraction	# in photo	fraction
Ia	500	85.9%	1625	40.4%
II	62	10.7%	2311	57.5%
Ibc	20	3.4%	86	2.1%
total	582	100%	4022	100%

Most training sets are non-representative in some way. Spectroscopic follow-up is always biased!

Example: SDSS supernova survey

Schuhmann et al. in prep

Constructing Decision Trees

Constructing Decision Trees

- How do you decide which feature to split on?
- What you want is the feature (and value of that feature) which best separates the data between classes to create the smallest possible tree
- How do you determine how well a feature separates the data?

Constructing Decision Trees

Information gain

Entropy is defined as: $$H(T) = -\sum_{i}^{n} p_i \log_2 p_i$$

where p_i is the proportion of the subset belonging to the i'th class.

The best feature split occurs when information gain (entropy of parent – entropy of all children) is maximised

Constructing Decision Trees

Gini Impurity

Idea is to minimise misclassification. The Gini impurity is defined as:

$$I_G(T) = -\sum_{i}^{n} p_i(1 - p_i)$$

Constructing Decision Trees

Gini Impurity

Idea is to minimise misclassification. The Gini impurity is defined as:

$$I_G(T) = -\sum_{i}^{n} p_i(1 - p_i)$$

In practice, the choice of criterion has less impact on results than choices on the construction of your tree (such as the maximum size of the tree).

Types of Ensemble Methods

Bagging

Randomly selects subsets of data (with replacement) to train an ensemble of trees and then averages the result

Boosting

Boosting iteratively runs decision trees upweighting hard-to-classify data in each iteration.

Random Forests

Uses bagging with the "random subspace method", randomly selecting which features to give to the decision tree classifier to further reduce possible overfitting.

Neural Networks

Question

The total error is given by:

$$E_{\text{total}} = \sum \frac{1}{2}(\text{target} - \text{predicted output})^2$$

- How would you use this to help you learn the weights of a neural network?

Backpropagation

One commonly used, simple method is backpropagation. Essentially, you want to quantify how much a given weight affects the error. This implies a derivative, for example $\frac{\partial \mathrm{E}}{\partial w_5}$

You can use the chain rule to determine what this quantity is in terms of inputs and outputs to nodes.

Once you compute the value of this derivative, subtract it from the weight for the next iteration.

Backpropagation

You'll find the weights at the *beginning* of the network depend on the weights at the *end*, which is why backpropagation starts at the end of the network and works backwards.

These are also often called *feed-forward* networks, because you first iterate through the network forwards to compute the error, then compute the derivatives *backwards* with backpropagation to adjust the weights.