Machine Learning

Dr. Michelle Lochner
AIMS-DISCNET
Data Science School 2018

AIMS African Institute for Mathematical Sciences SOUTH AFRICA

SKA AFRICA SQUARE KILOMETRE ARRAY

Materials

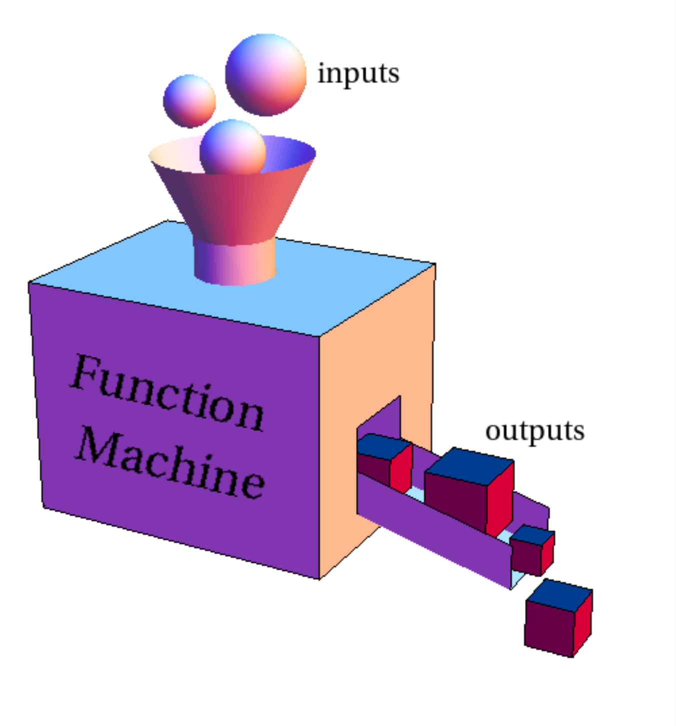
https://github.com/MichelleLochner/ml-tutorials/

```
from python import solution
```

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)

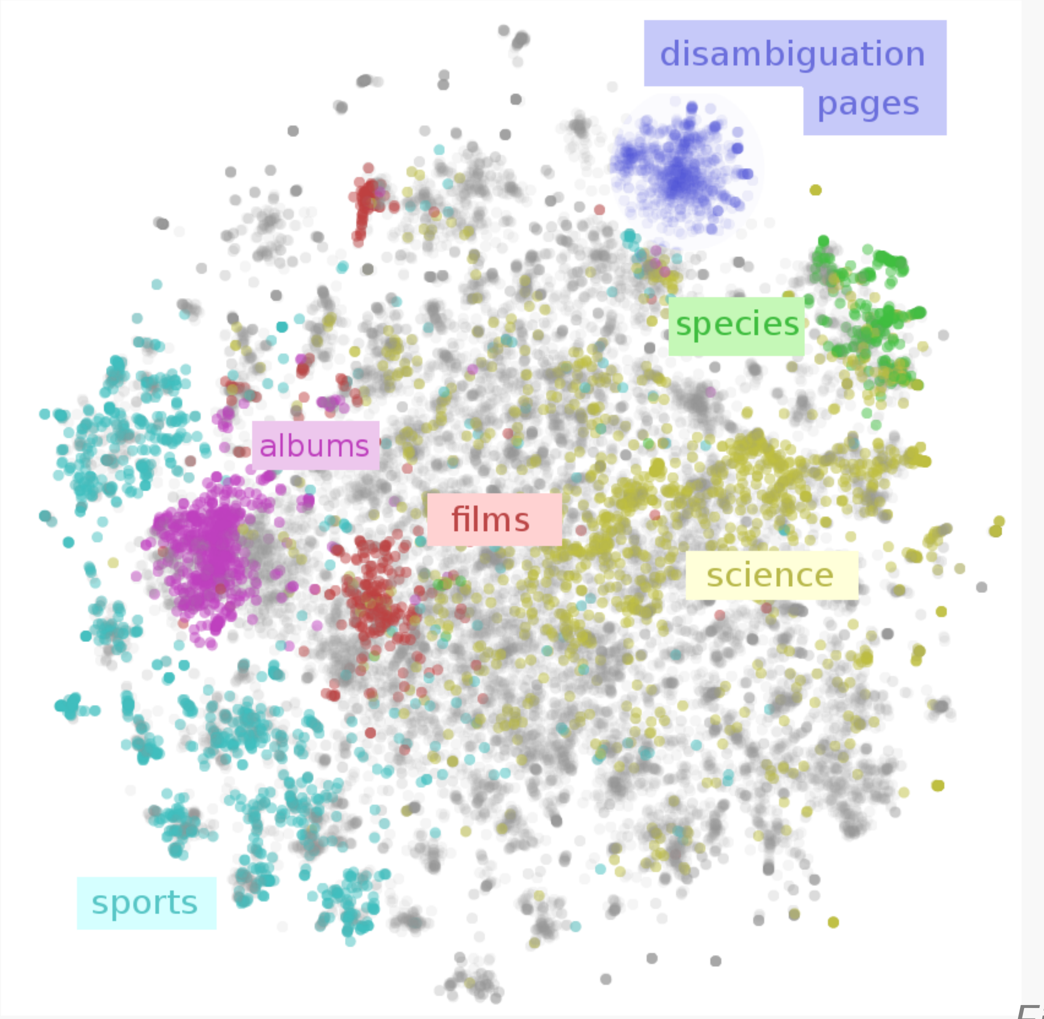
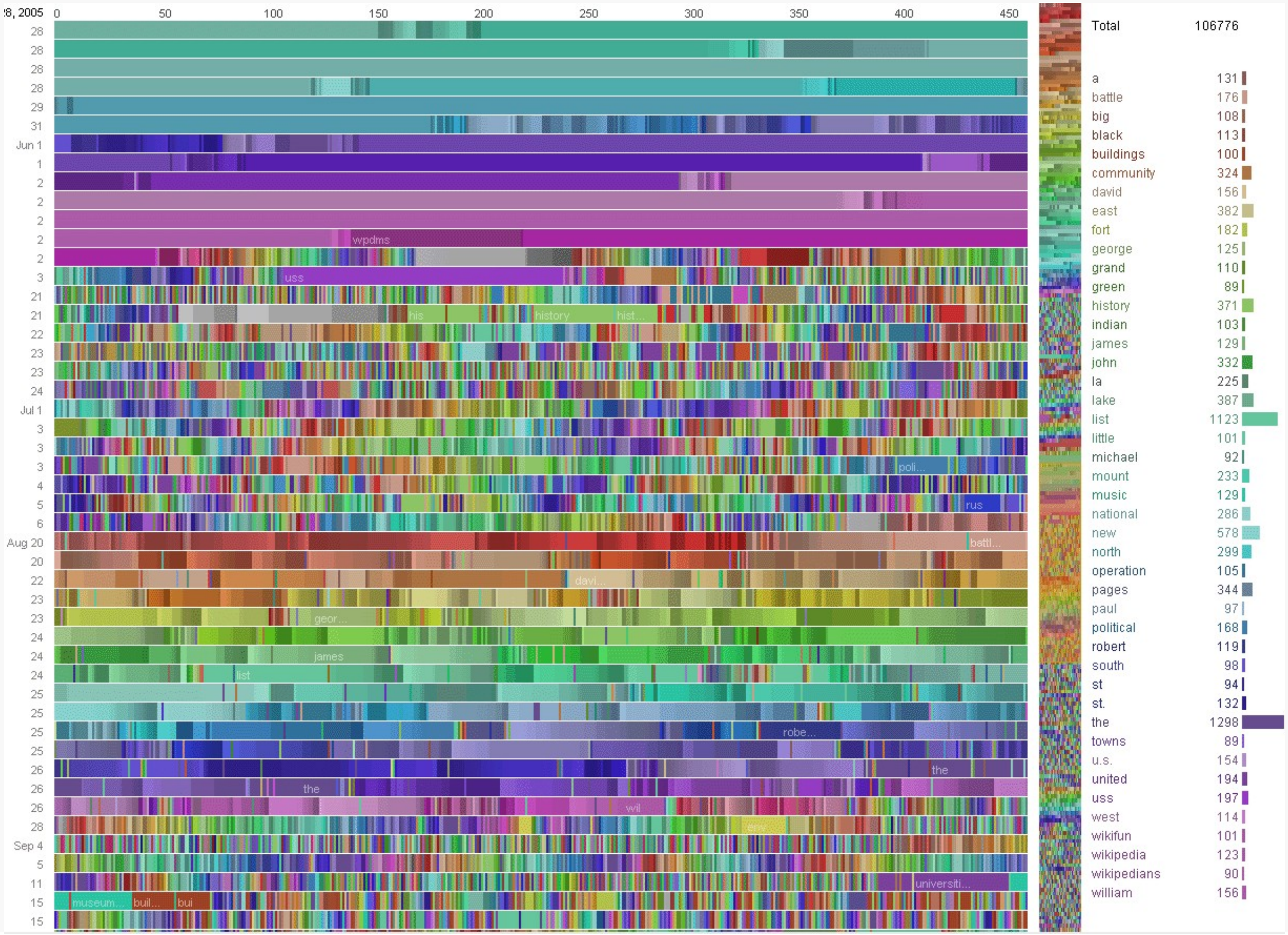


Figure: http://colah.github.io/

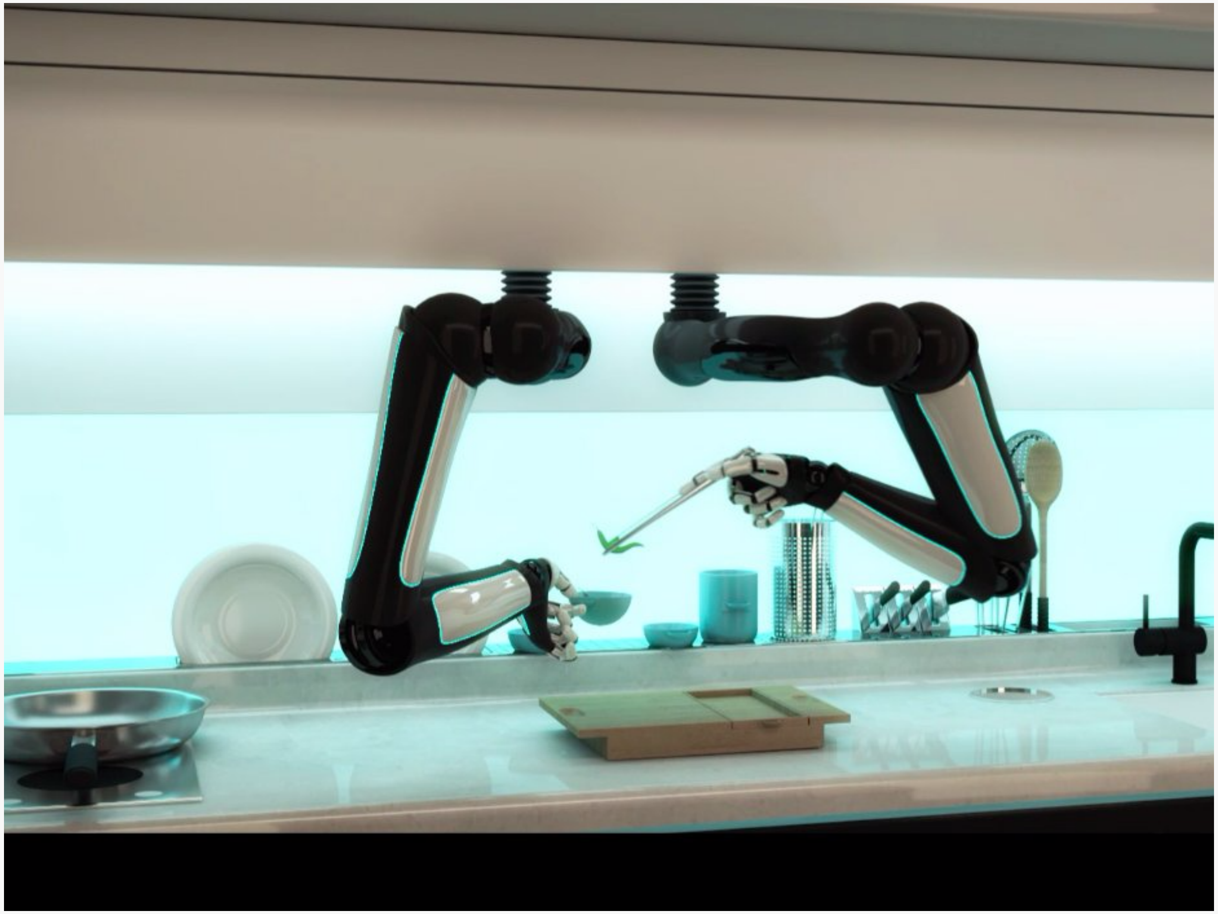
When to use machine learning

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



Questions

- Why has machine learning, which has been around for over 50 years, only recently become so popular?

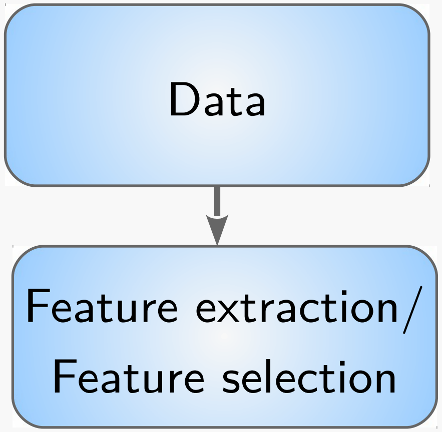
Questions

- Why has machine learning, which has been around for over 50 years, only recently become so popular?
- What's an example of where machine learning is used in your every day life?

Some definitions

Some definitions

Some definitions

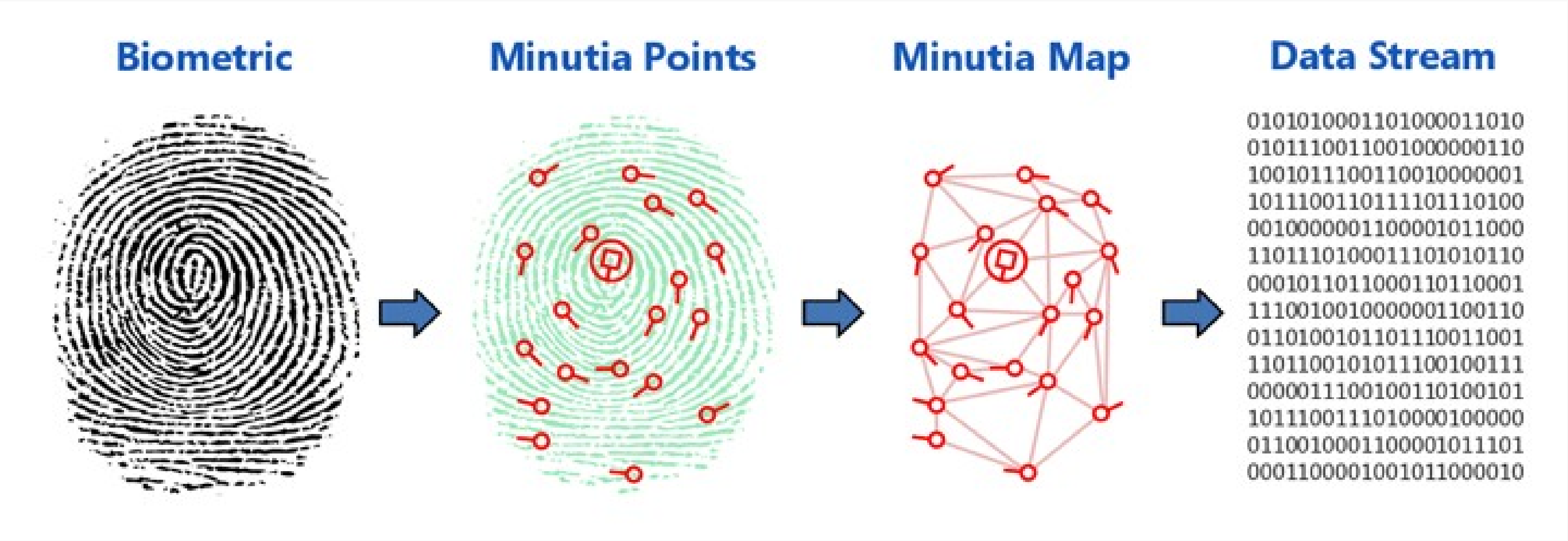


Some definitions

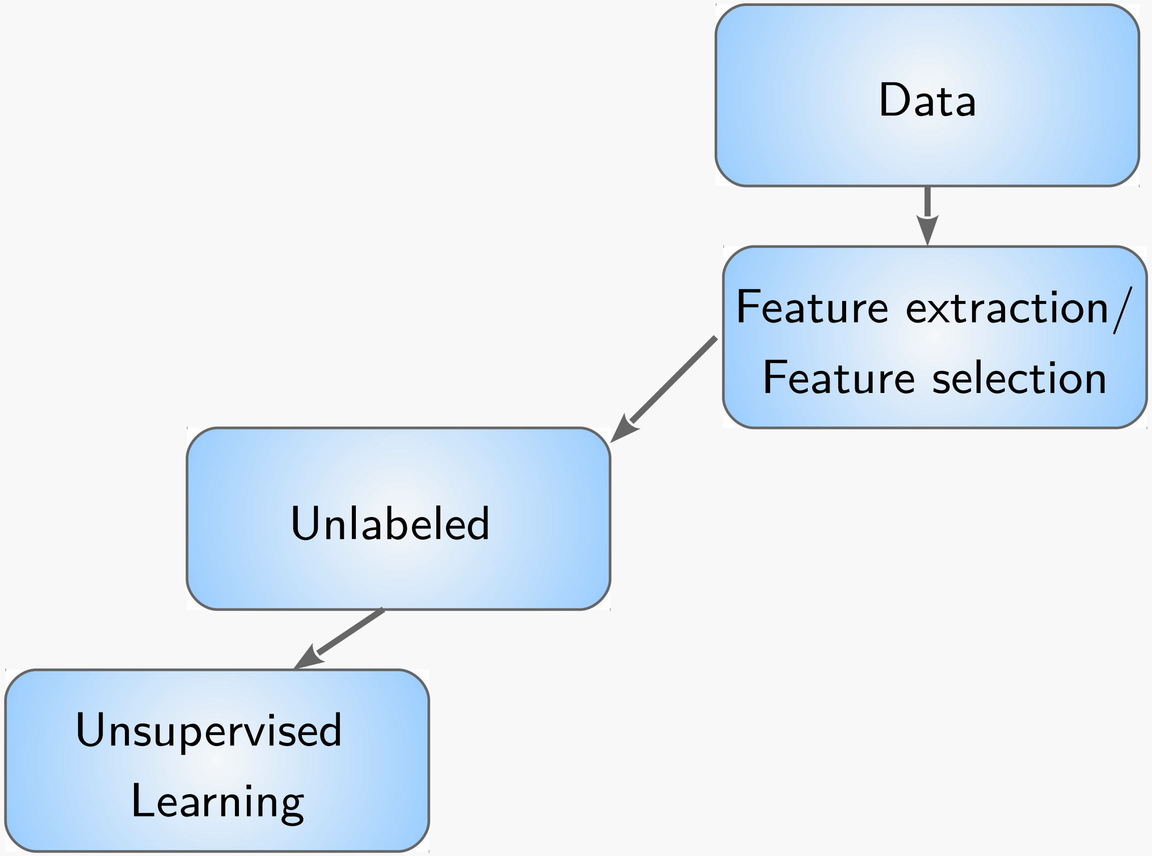
Data

↓

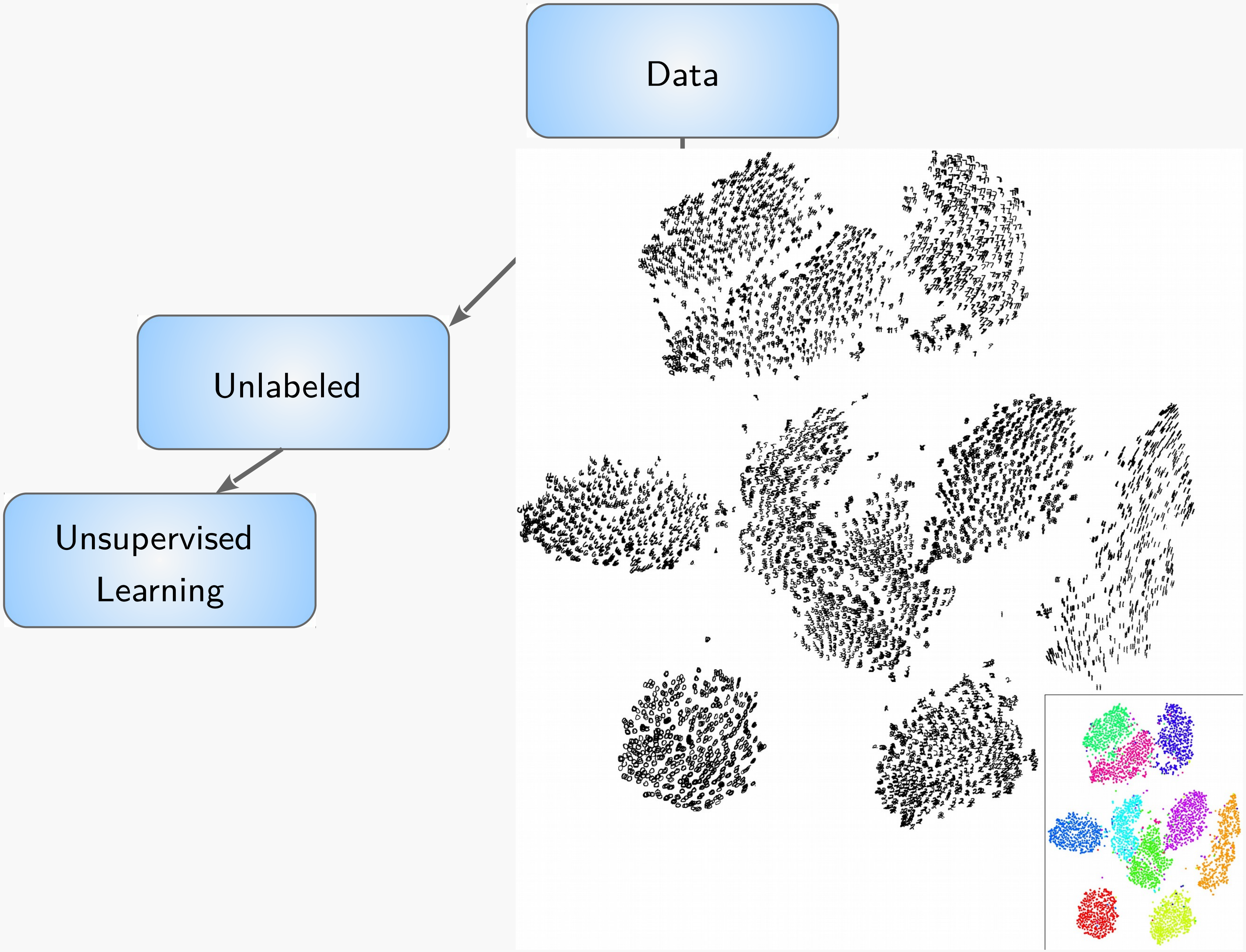
Feature extraction/
Feature selection



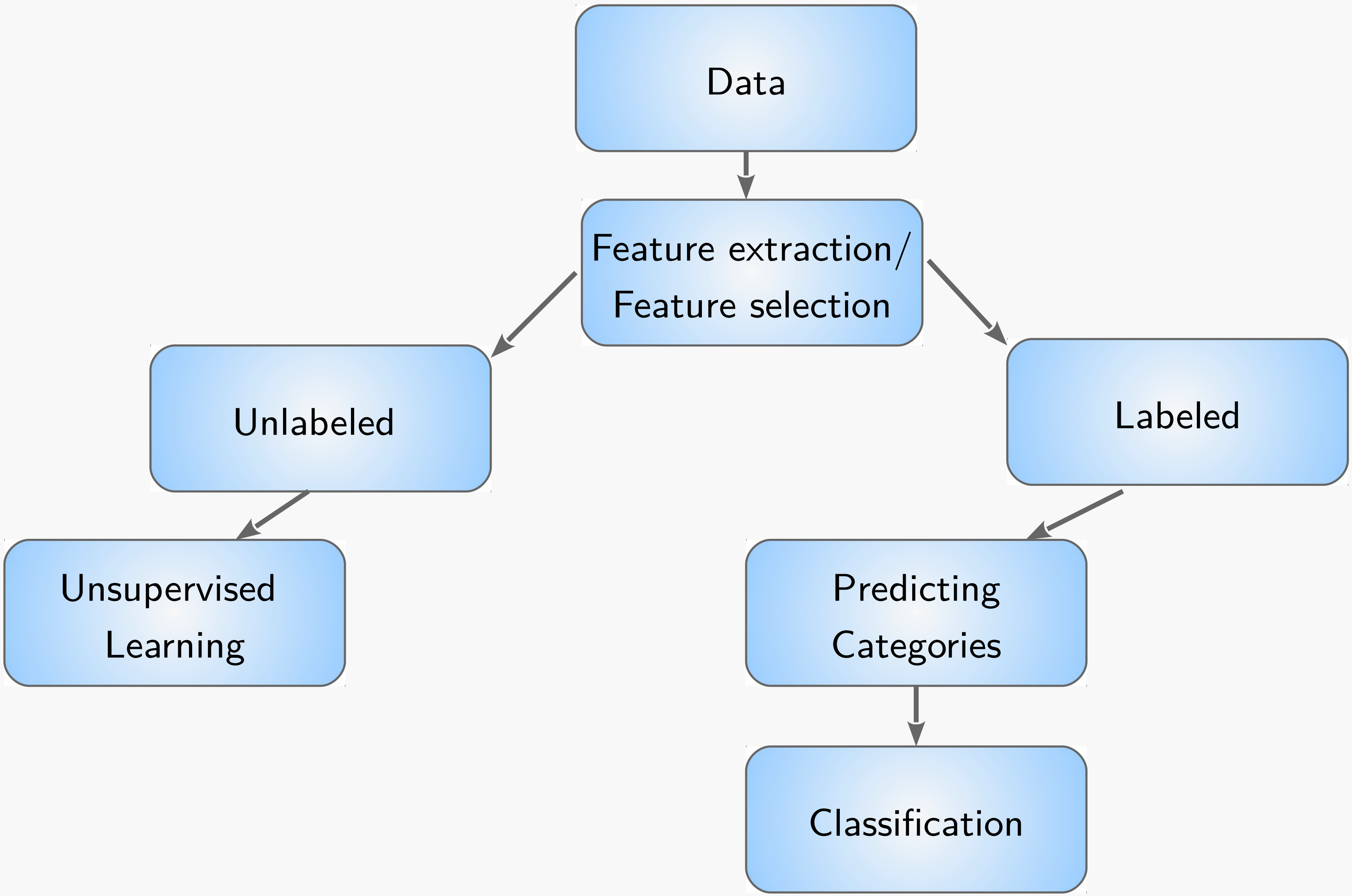
Some definitions



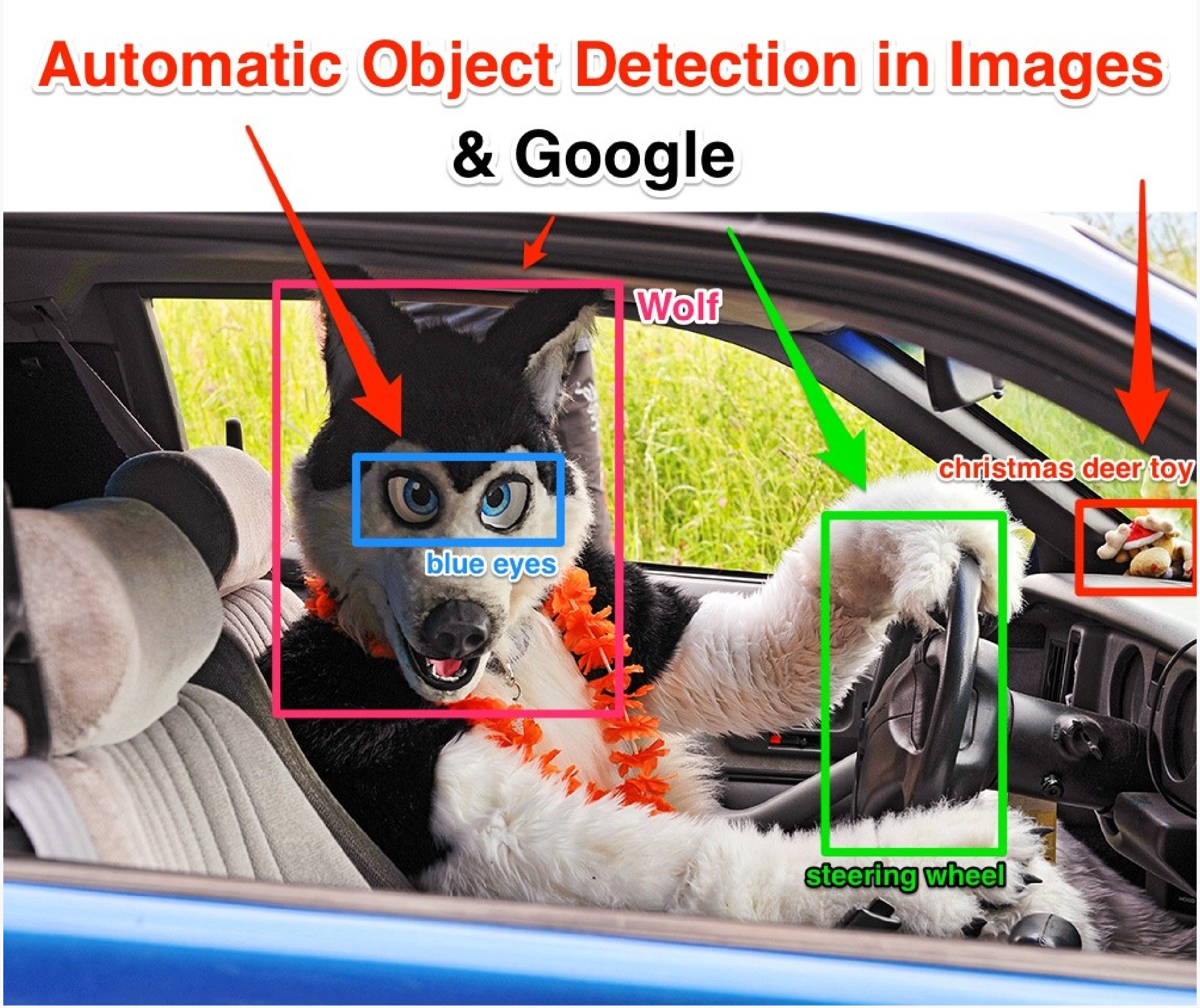
Some definitions



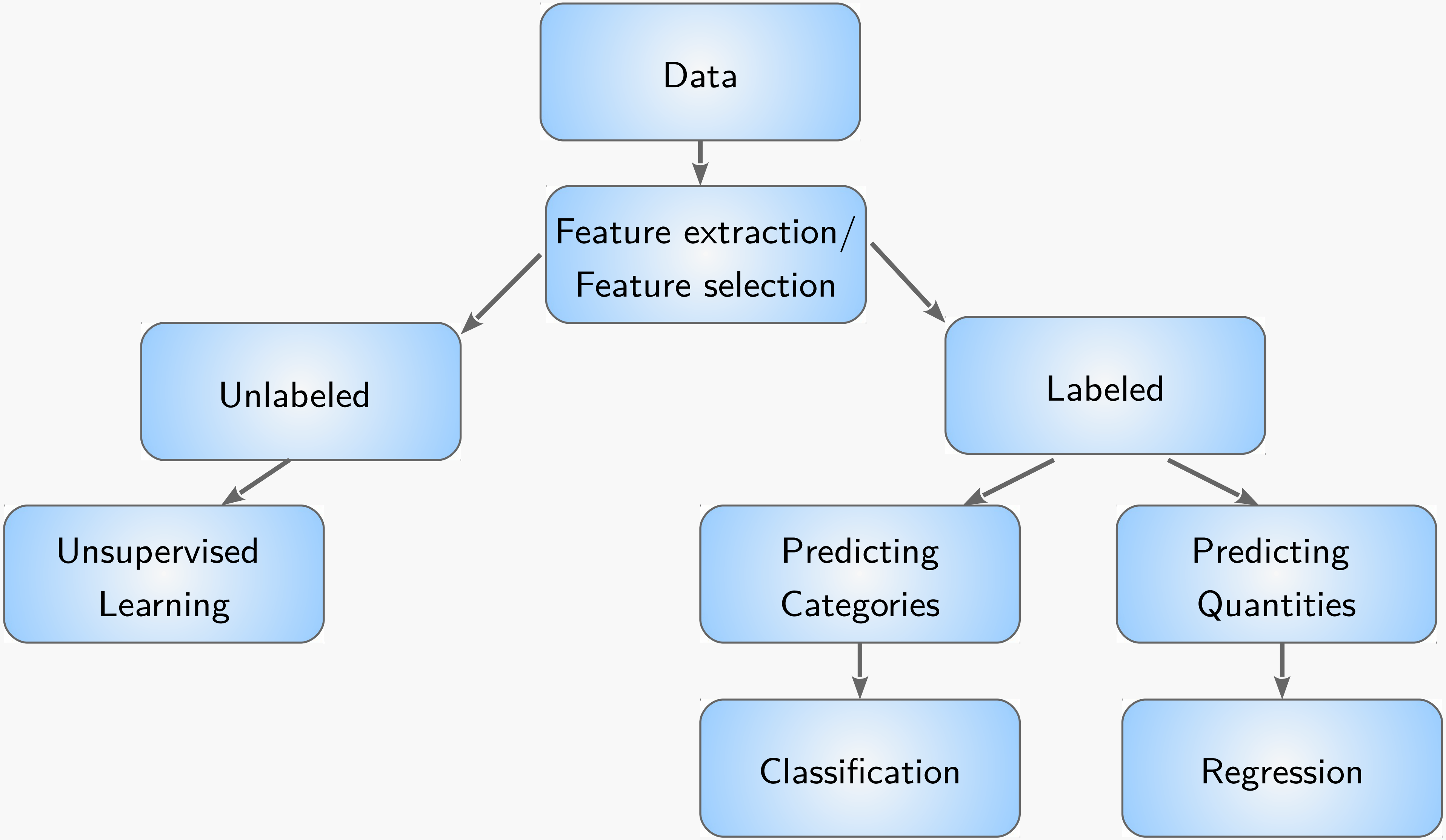
Some definitions



Some definitions

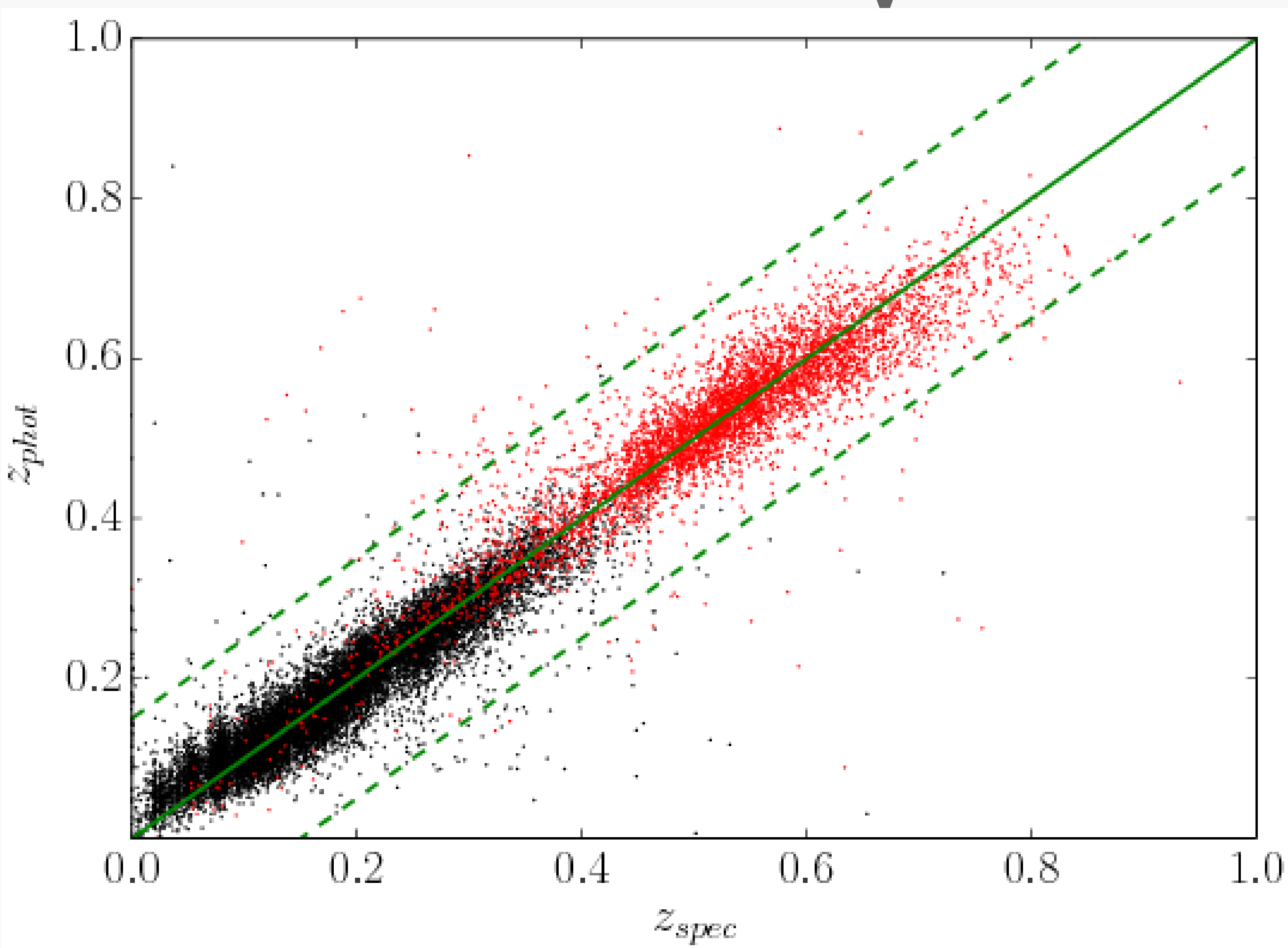


Some definitions



Some definitions

Data



Labeled

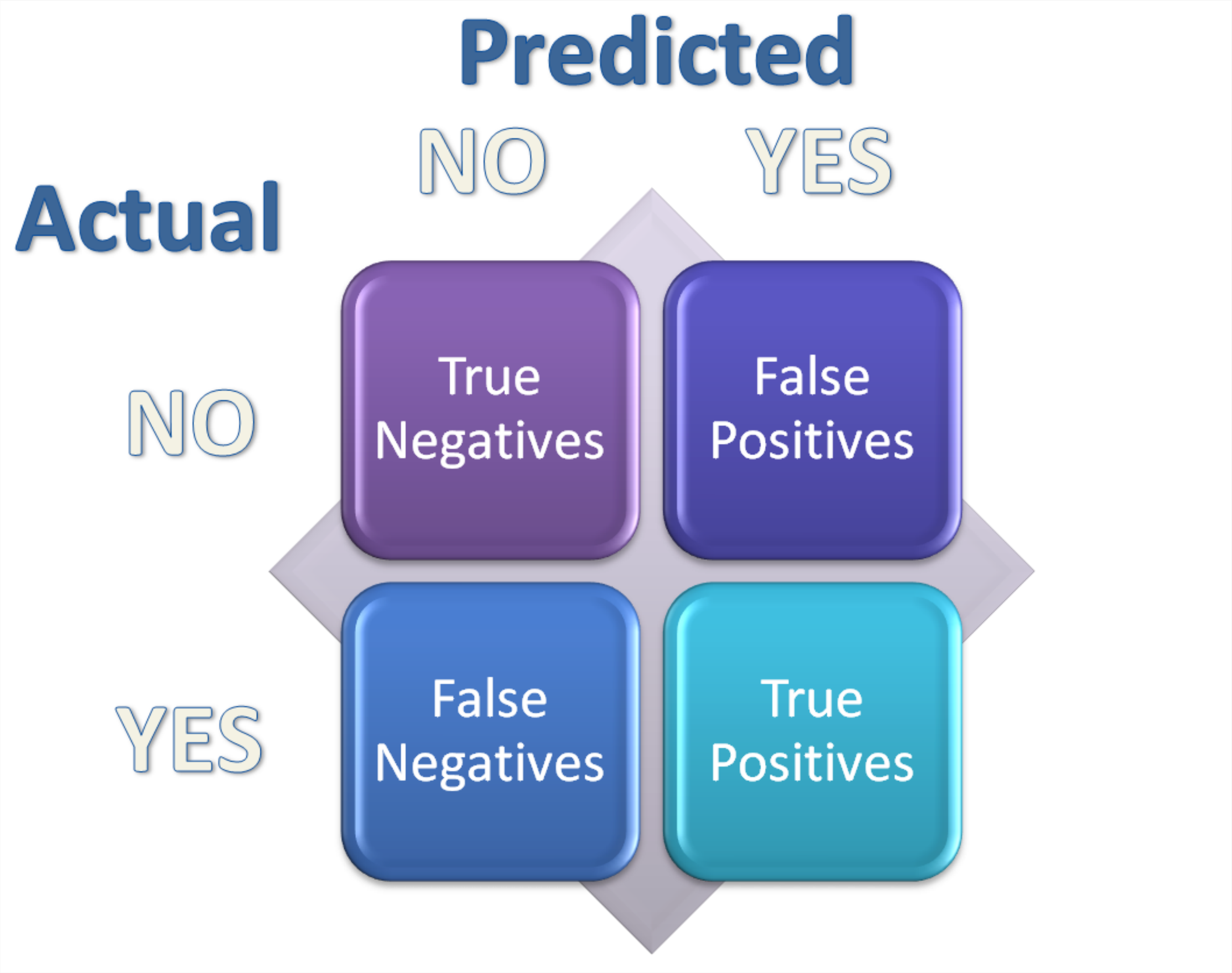
Predicting Quantities

Regression

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

Evaluating classification algorithms

Evaluating classification algorithms



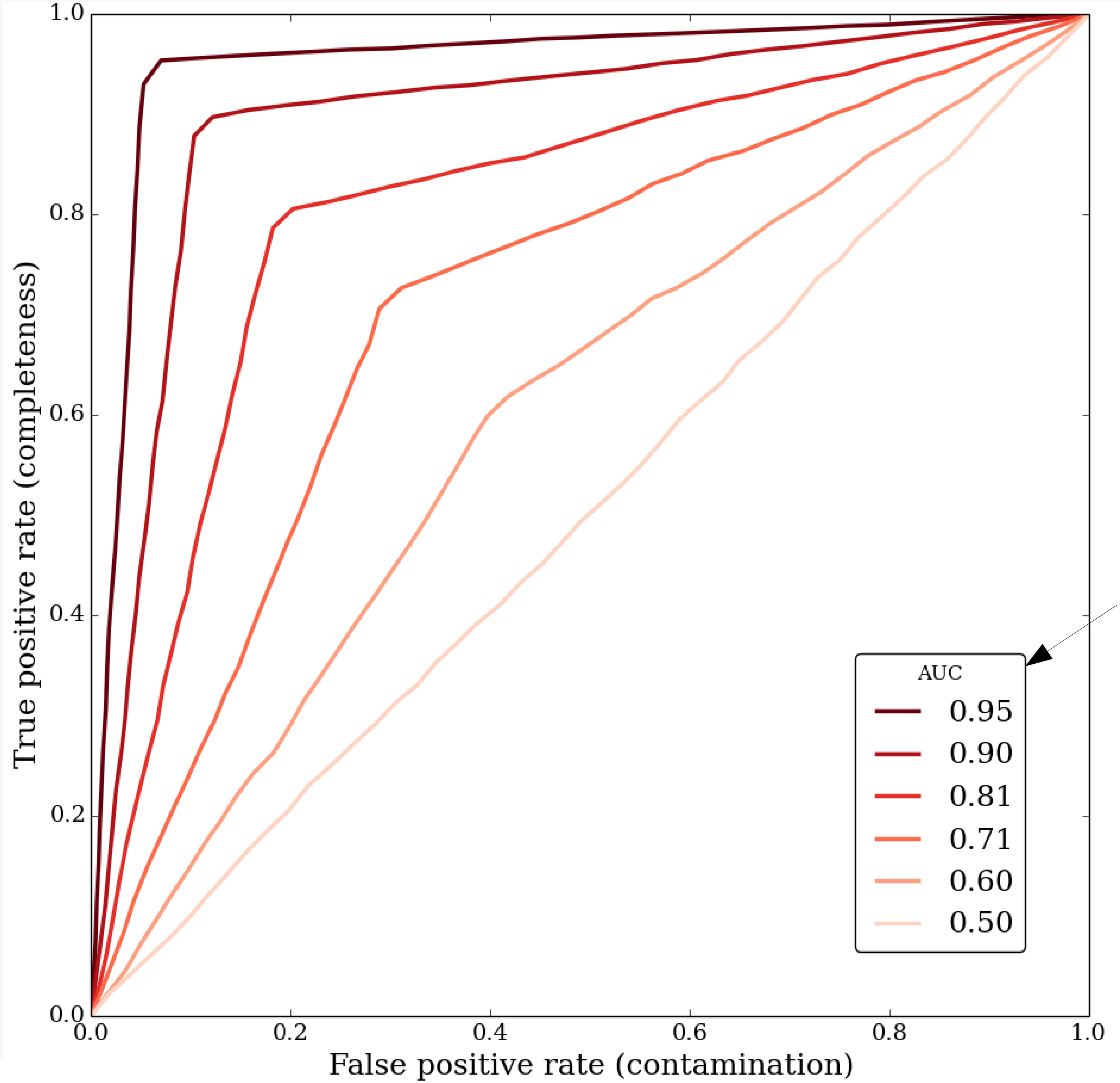
`sklearn.metrics.confusion_matrix`

Questions

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

Receiver Operator Characteristic (ROC) curves

`sklearn.metrics.roc_curve`



AUC =

Area Under Curve

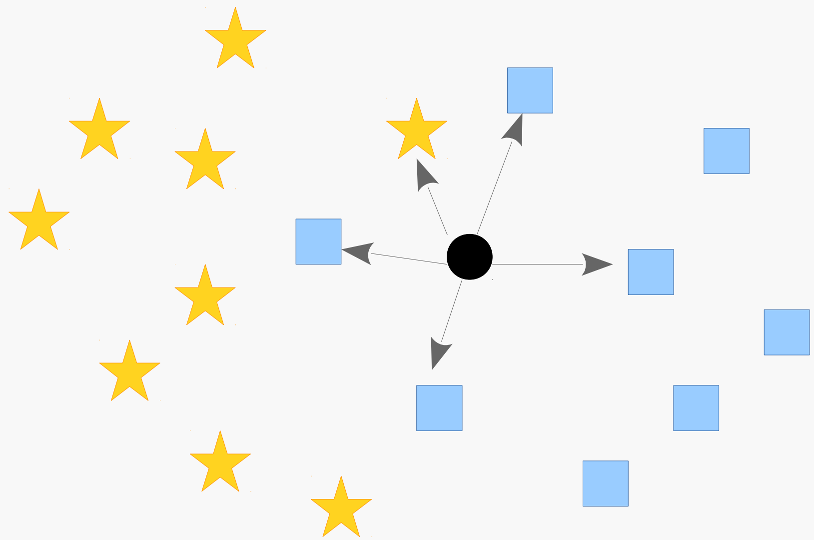
Machine Learning Algorithms

Machine Learning Algorithms

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

K-nearest Neighbours

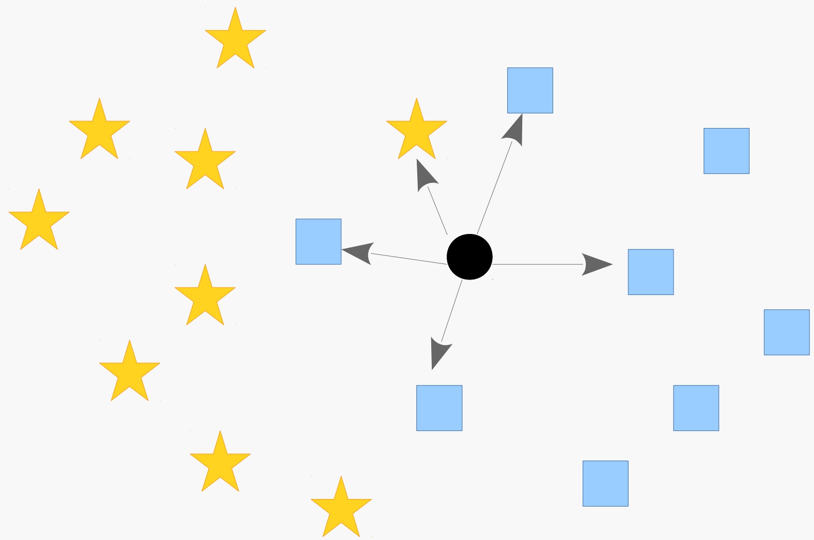
In it's simplest form, classify as the majority class of its k nearest neighbours.



`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

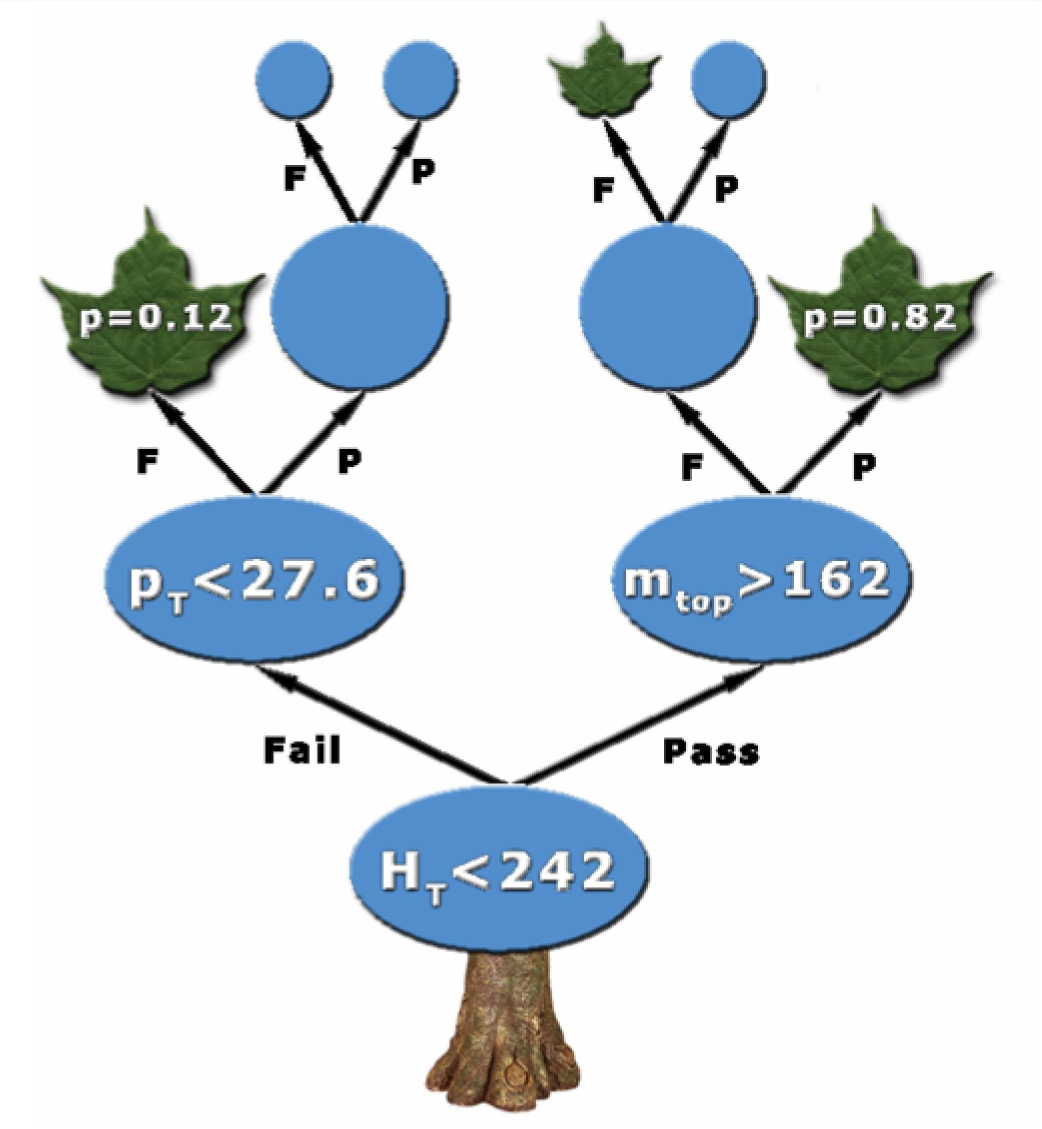
Advantages

- 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

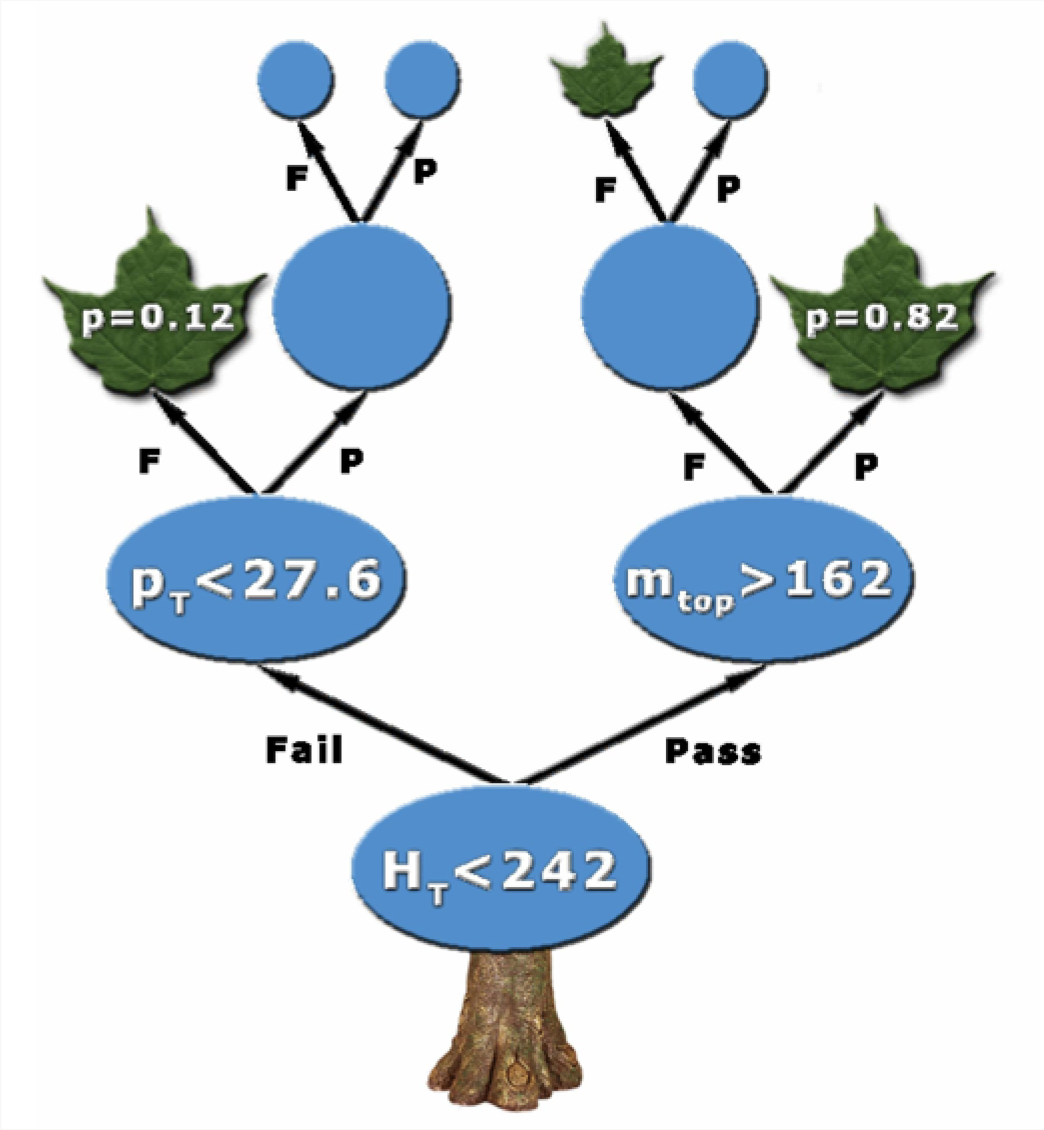
Constructing Decision Trees

- 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

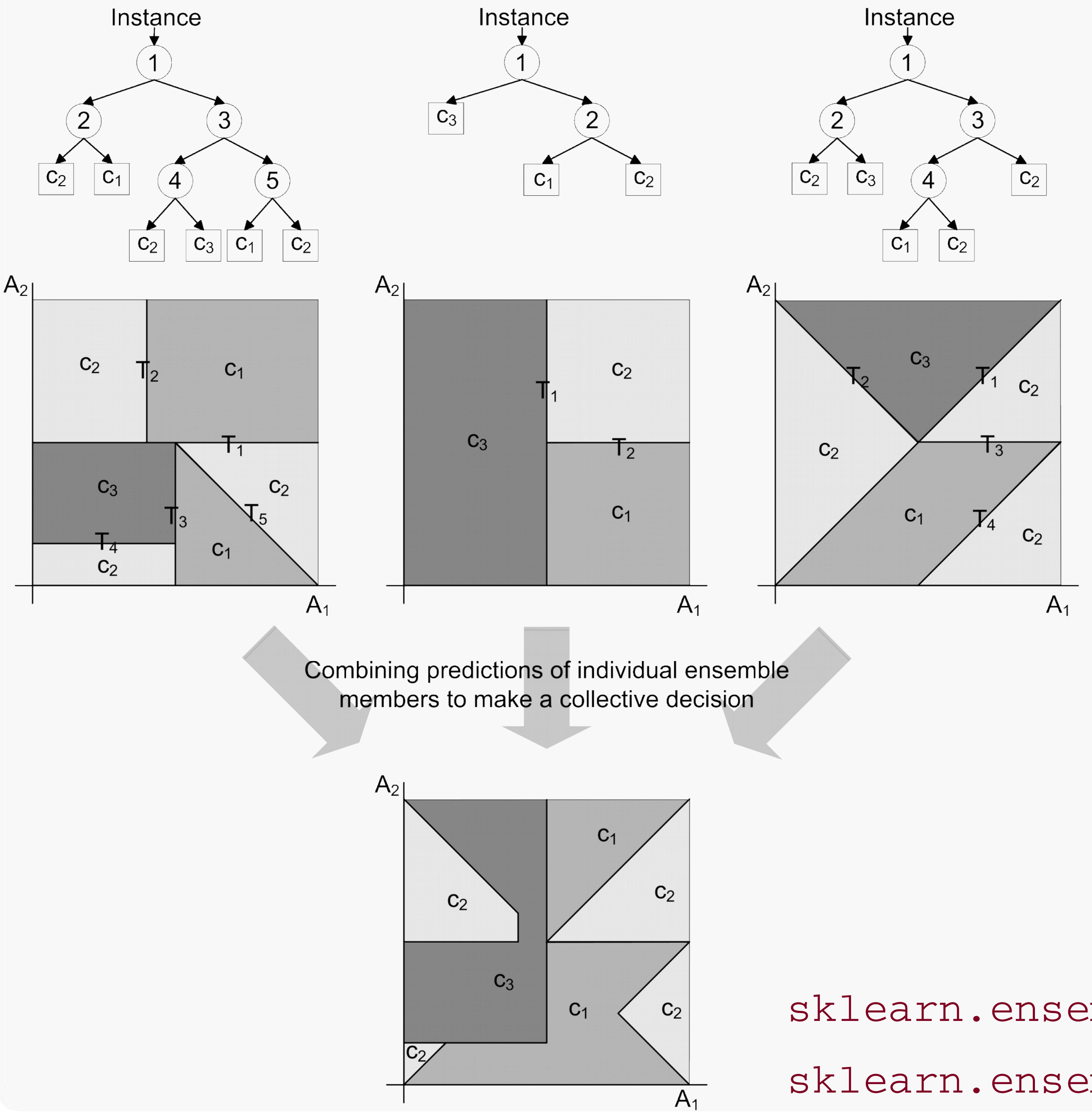
- At each leaf node, decide which is the best feature to split the data on (such that it separates best between classes), and what's the best split value of that feature
- To make this choice, use either the entropy or the Gini impurity (see extra slides)
- Making a prediction is simple, just propagate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

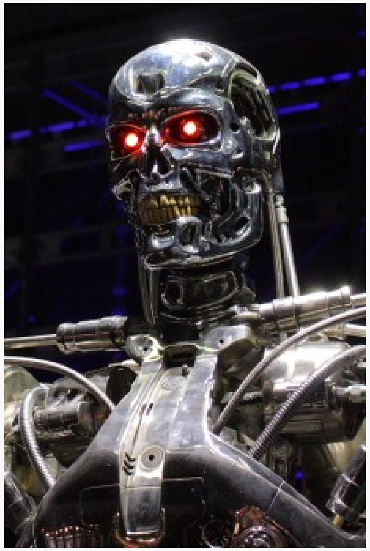
- Incredibly robust (low variance)
- Able to handle mixed feature types
- Robust to high dimensionality
- Able to naturally rank feature importance
- Random Forests is Michelle's favourite algorithm

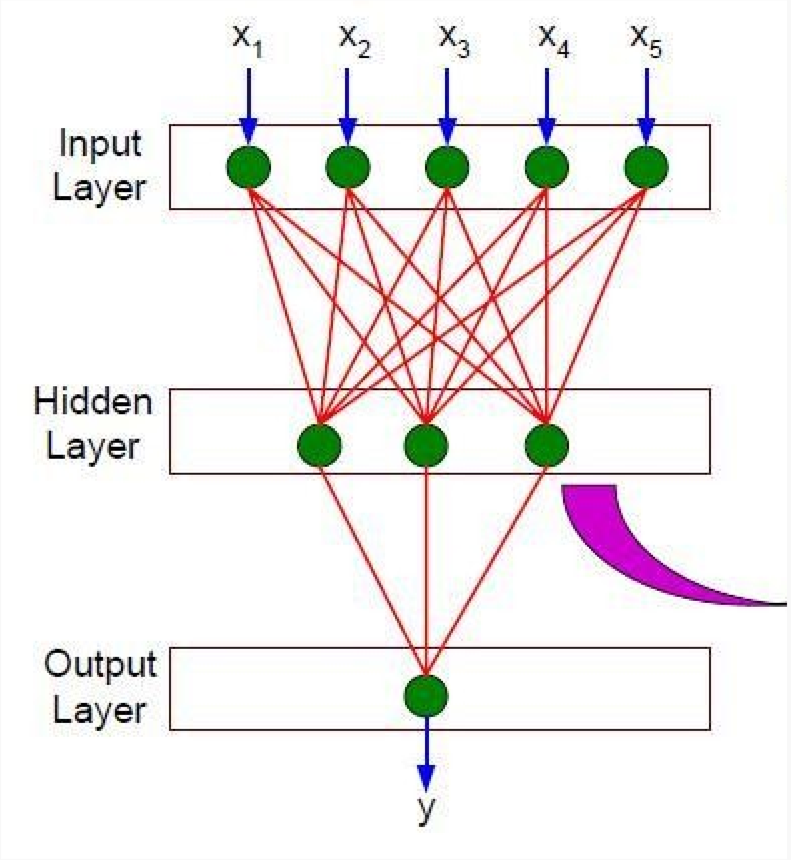
Disadvantages

- 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.

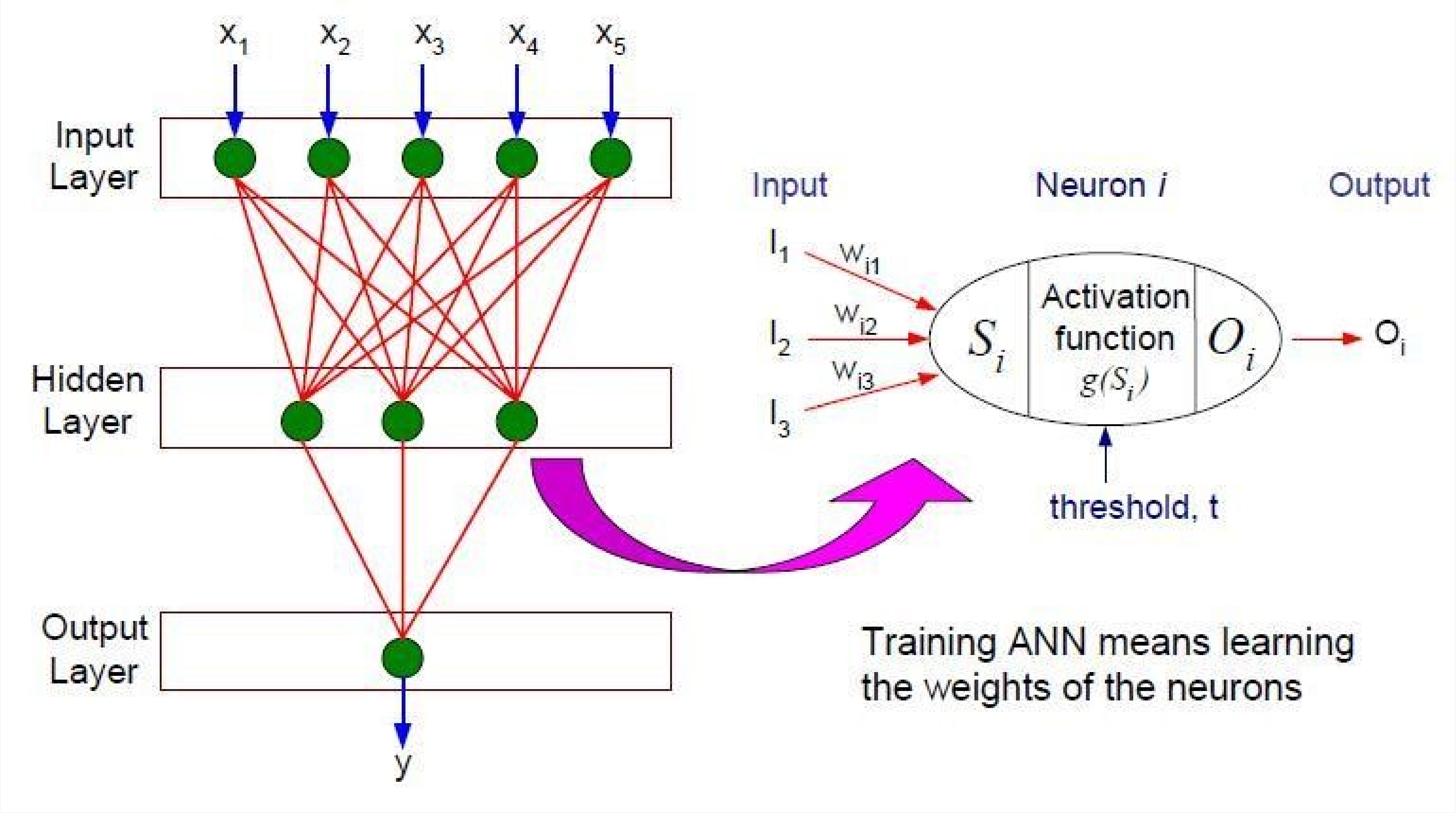


Figure: http://mines.humanoriented.com/

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

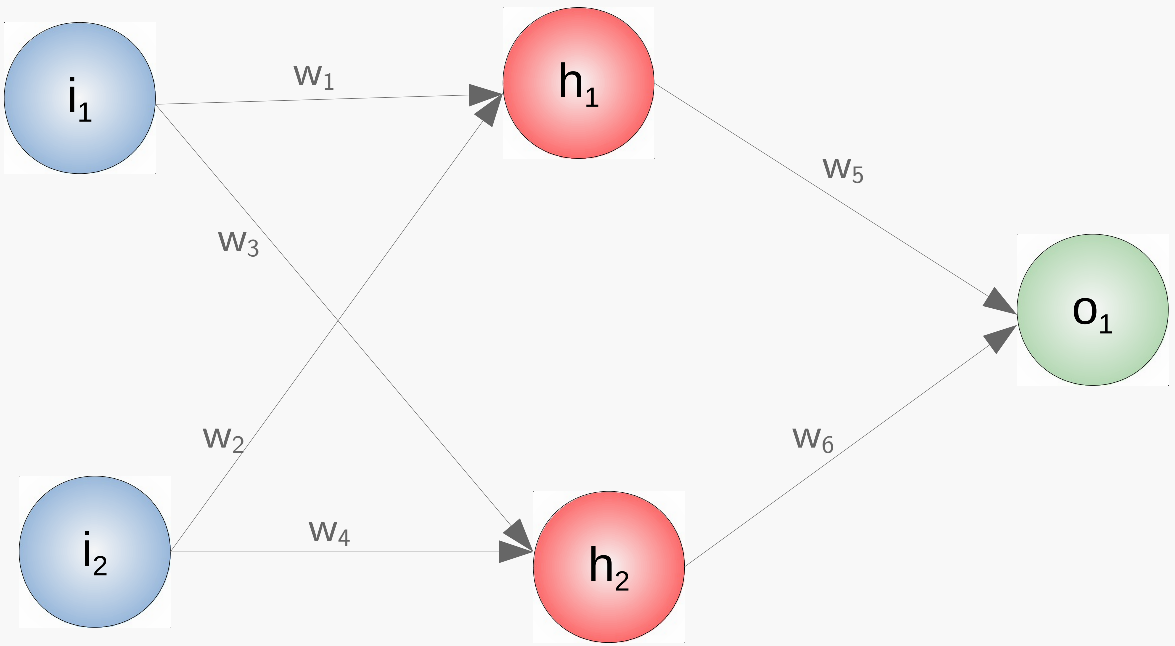
Disadvantages

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

Question

Question

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



Question

Question

- What is representativeness? In what ways can data be non-representative?

Question

- What is representativeness? In what ways can data be non-representative?
- Why is this a problem for machine learning?

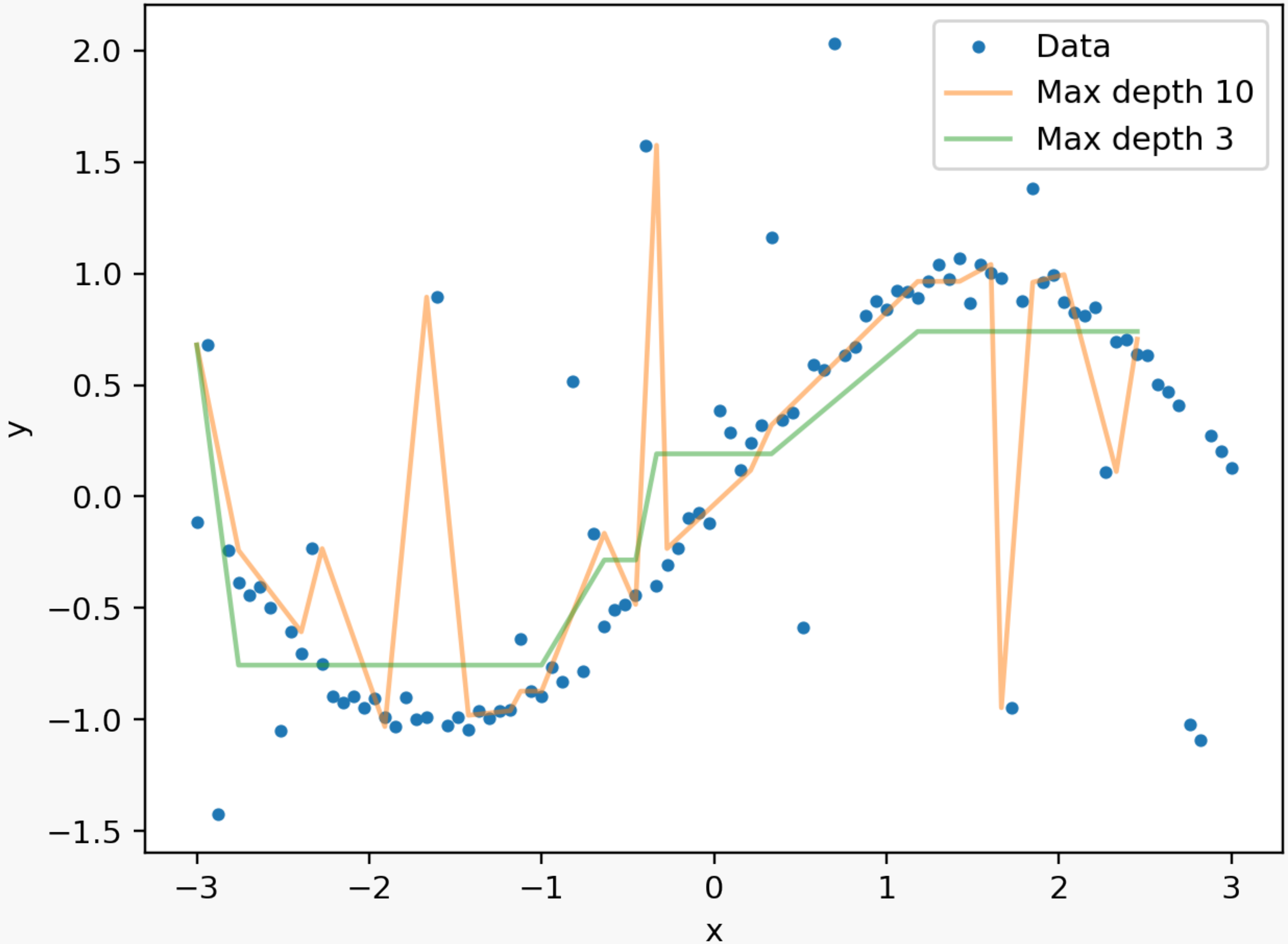
Question

- What is representativeness? In what ways can data be non-representative?
- Why is this a problem for machine learning?
- What can you do to mitigate non-representativeness?

Question

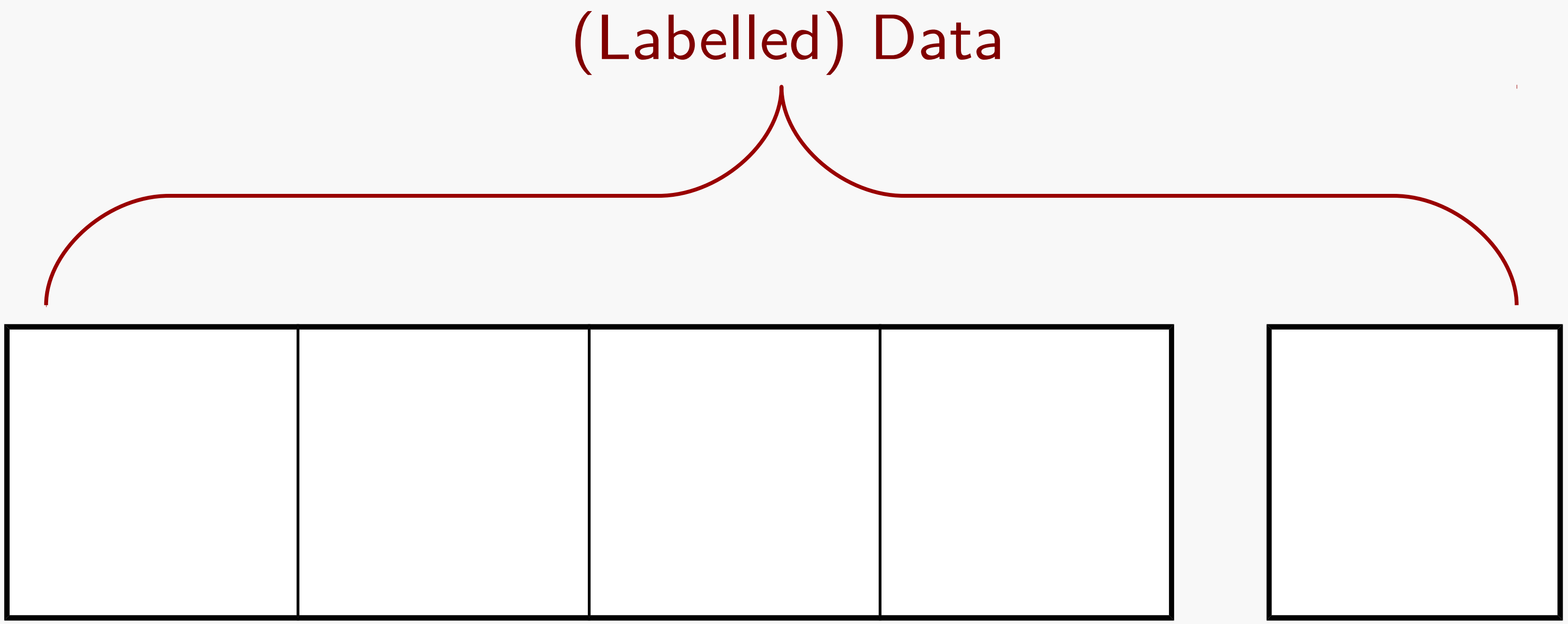
- What is overfitting?

Overfitting

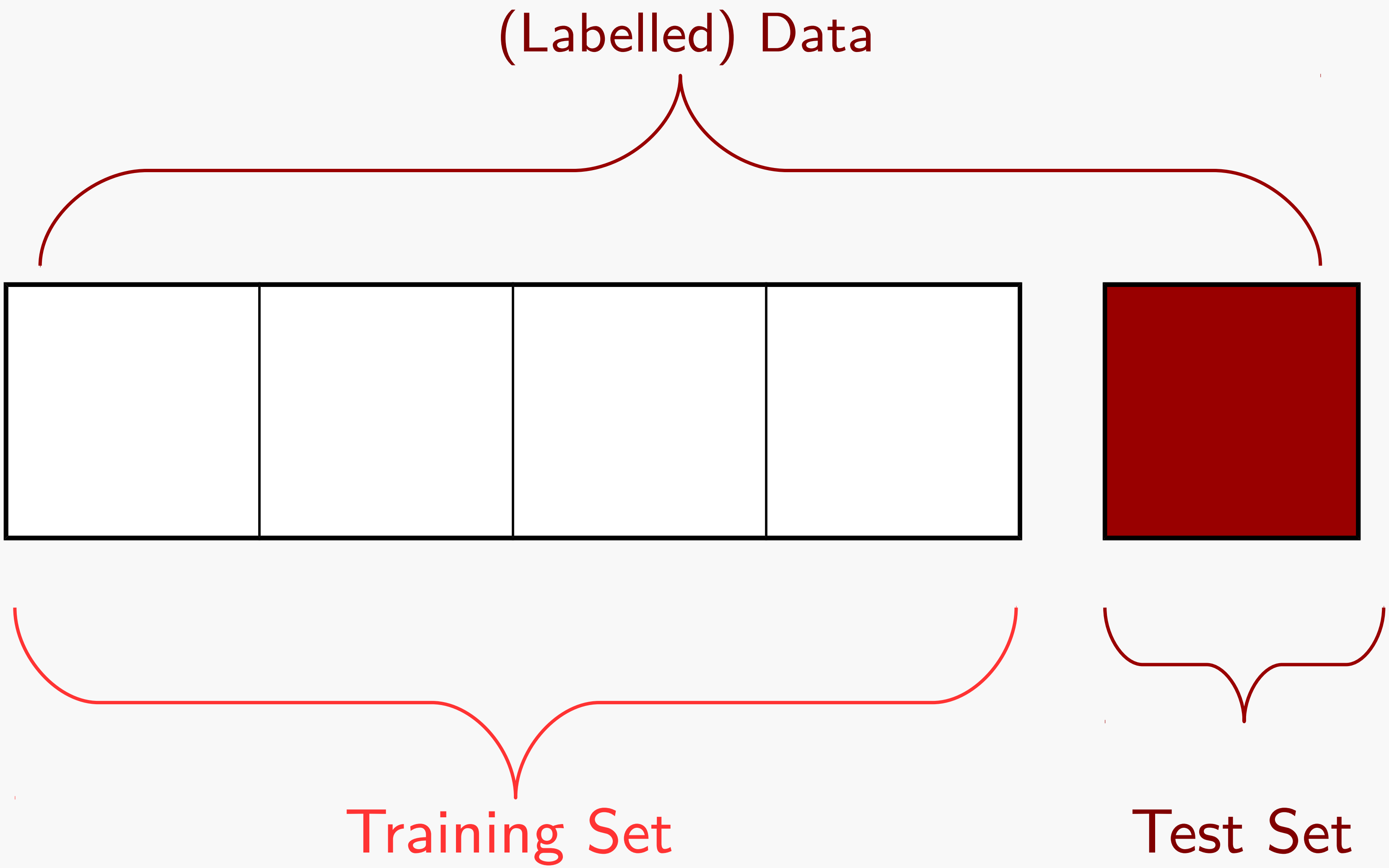


Overfitting example using Decision Trees

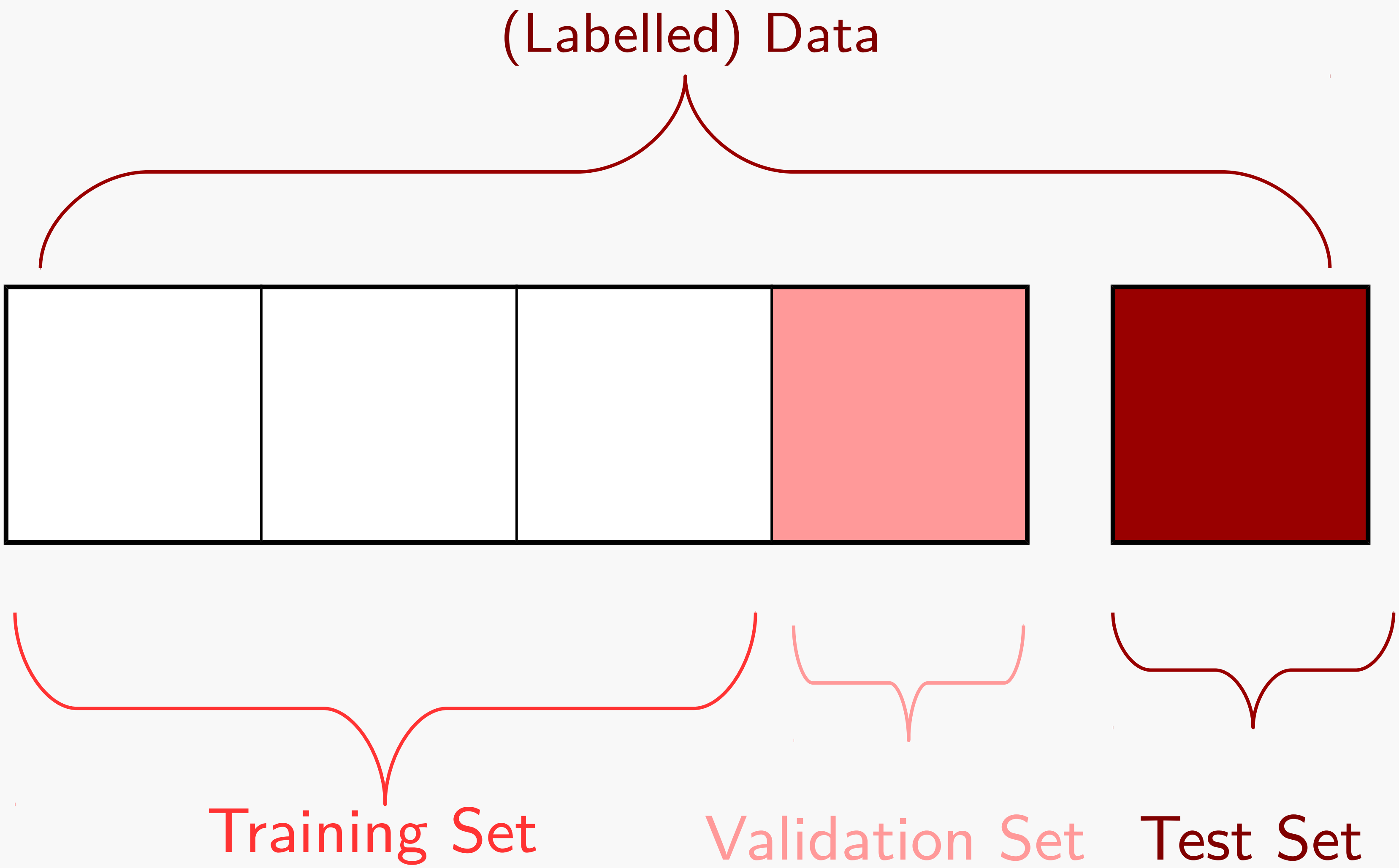
Training, Validation and Testing



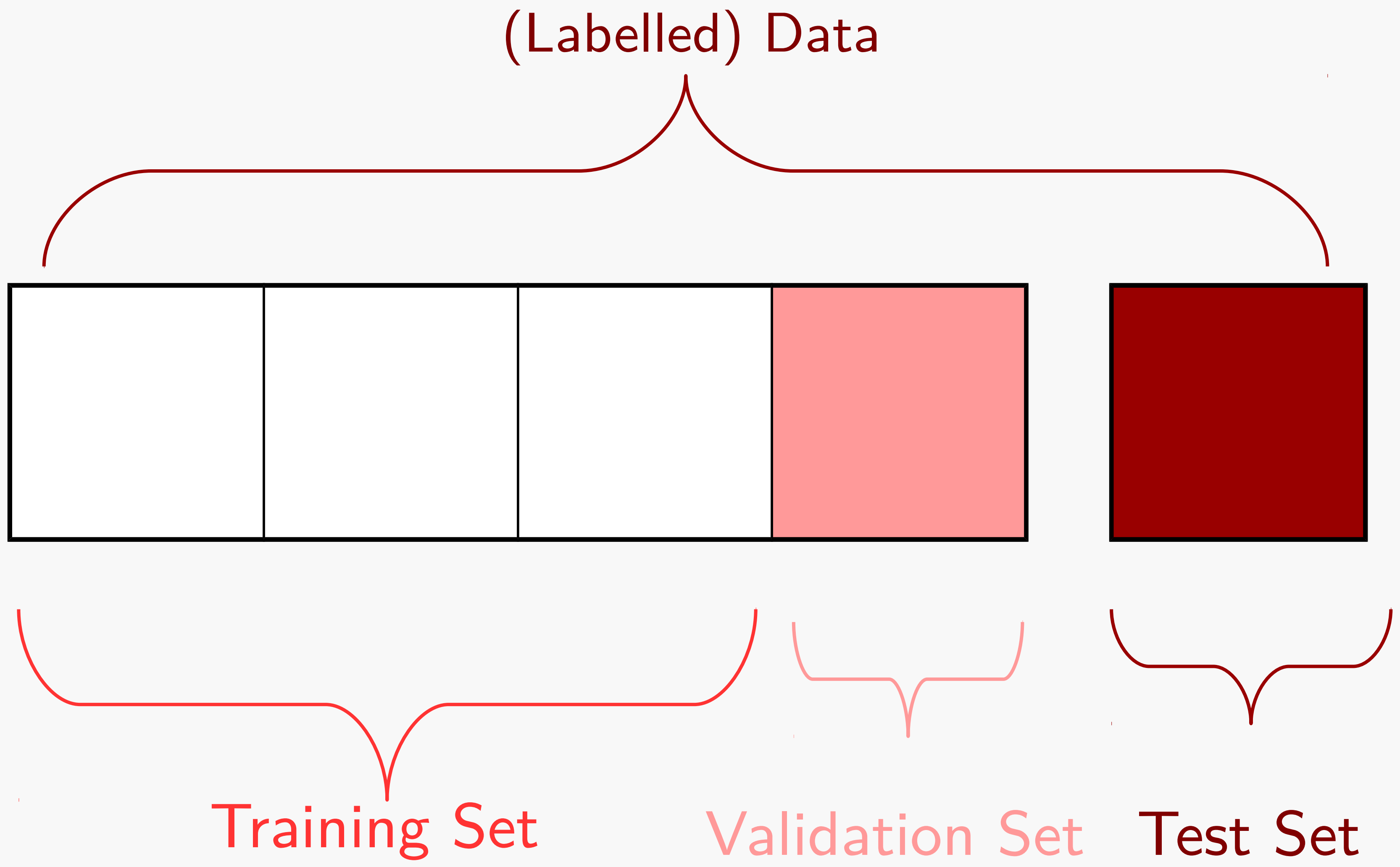
Training, Validation and Testing



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

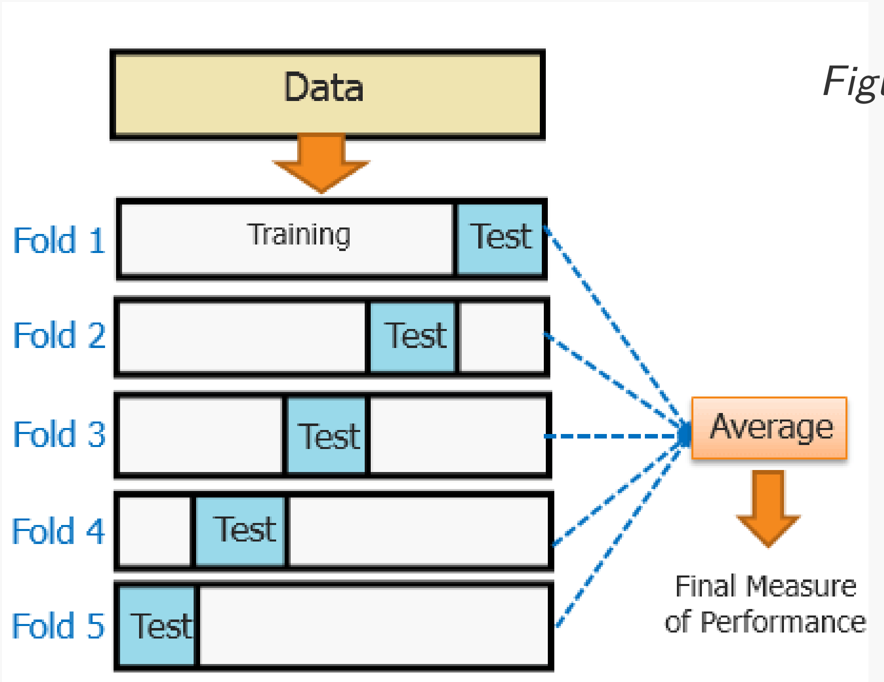
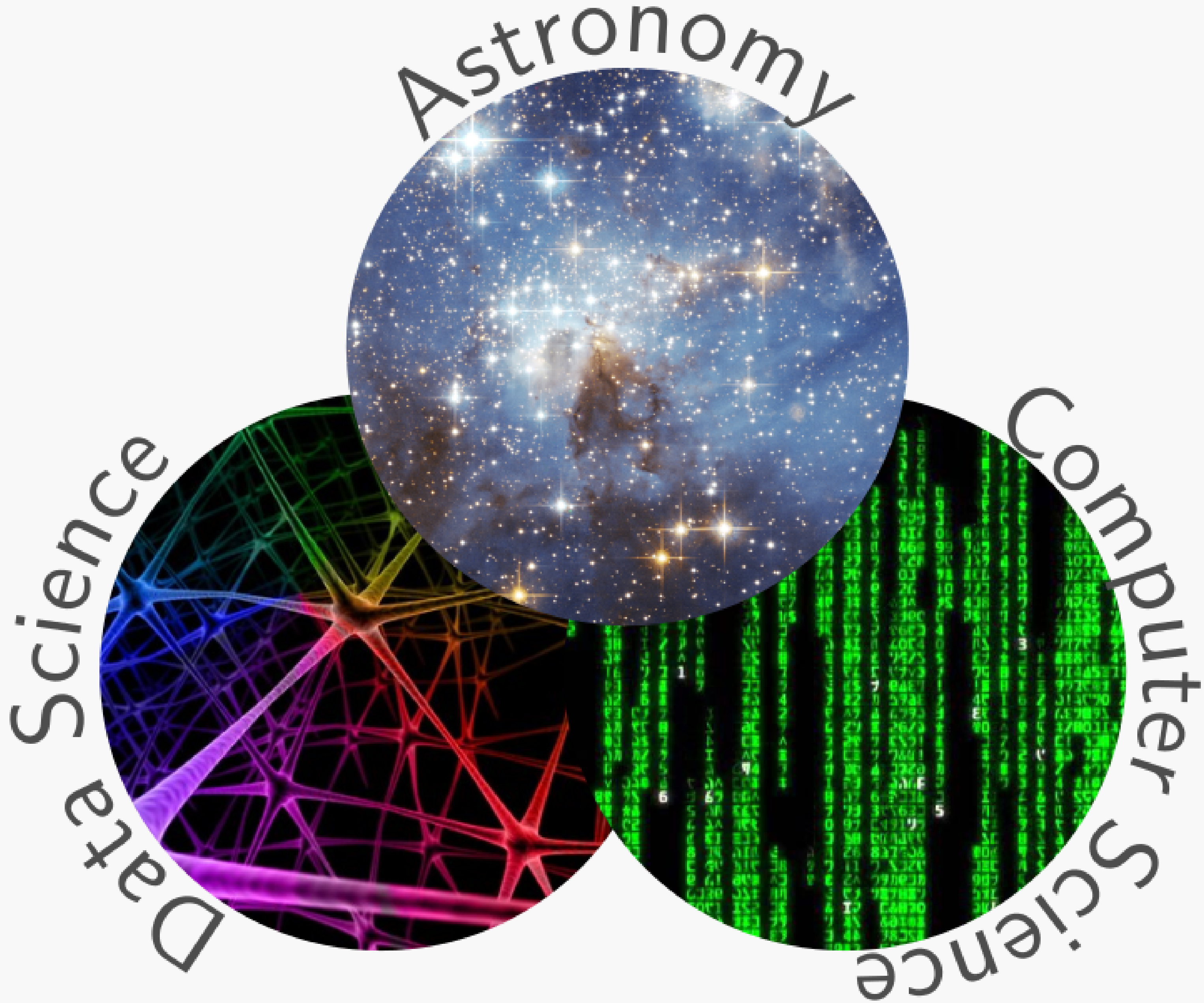


Figure: www.edureka.in/data-science

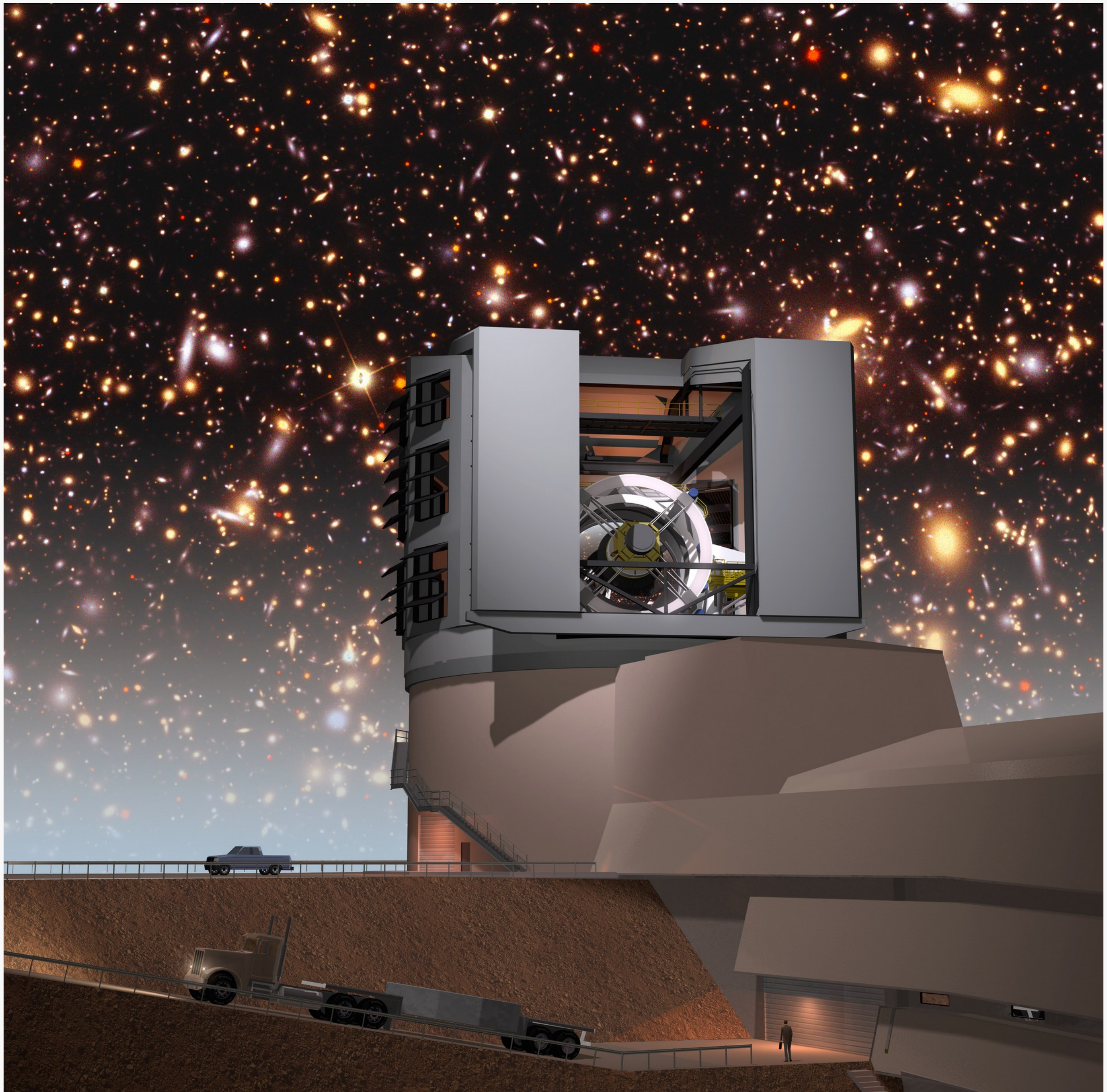
`sklearn.model_selection.GridSearchCV`

Example of Machine Learning

The Era of Survey Astronomy



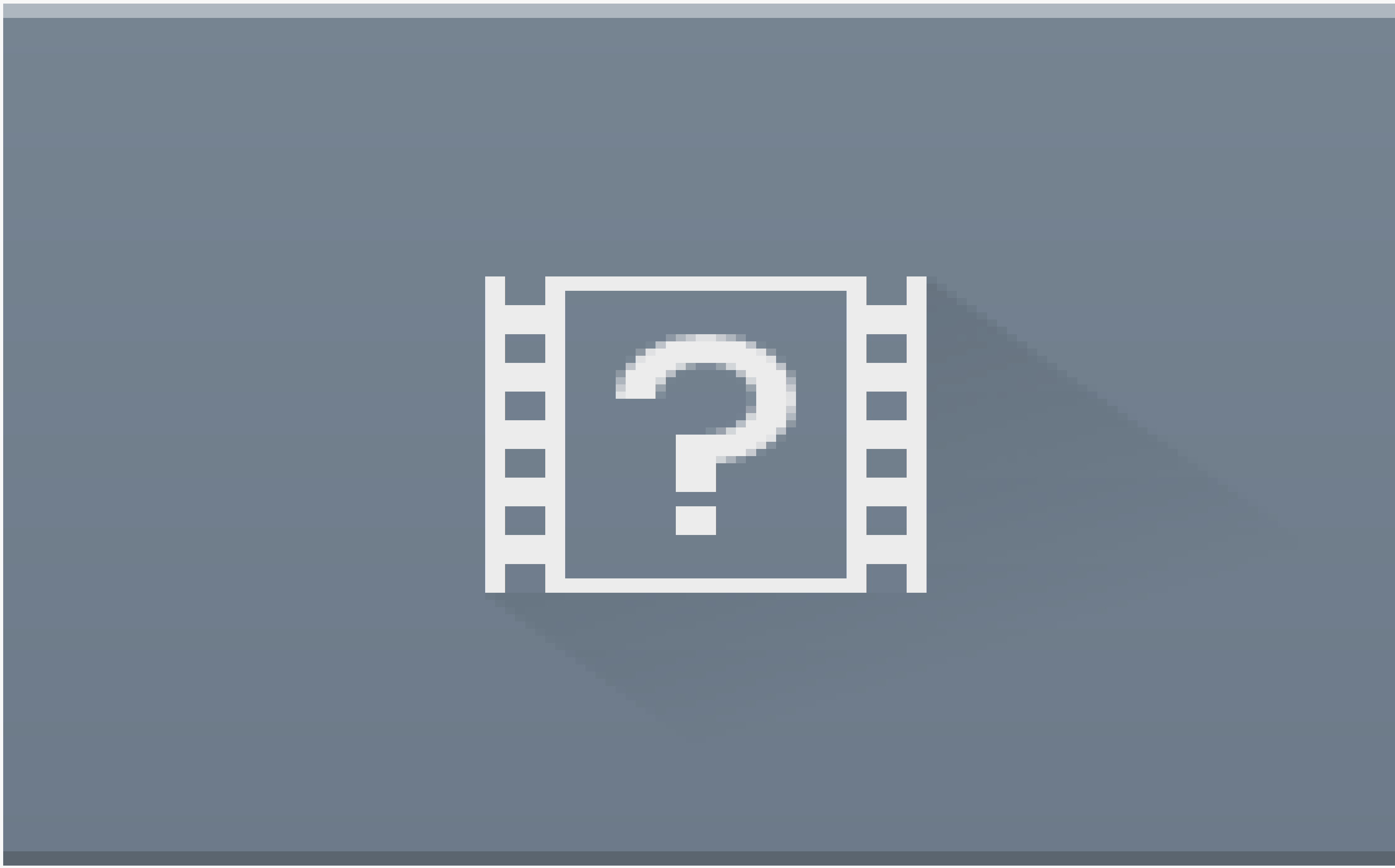
The Large Synoptic Survey Telescope



- The 10-year LSST survey will be the widest, fastest, deepest optical survey ever done.
- Science ranges from asteroids to dark energy
- Will detect ~10 million transient events per night, thousands of which will be new astrophysical objects

LSST

The Large Synoptic Survey Telescope

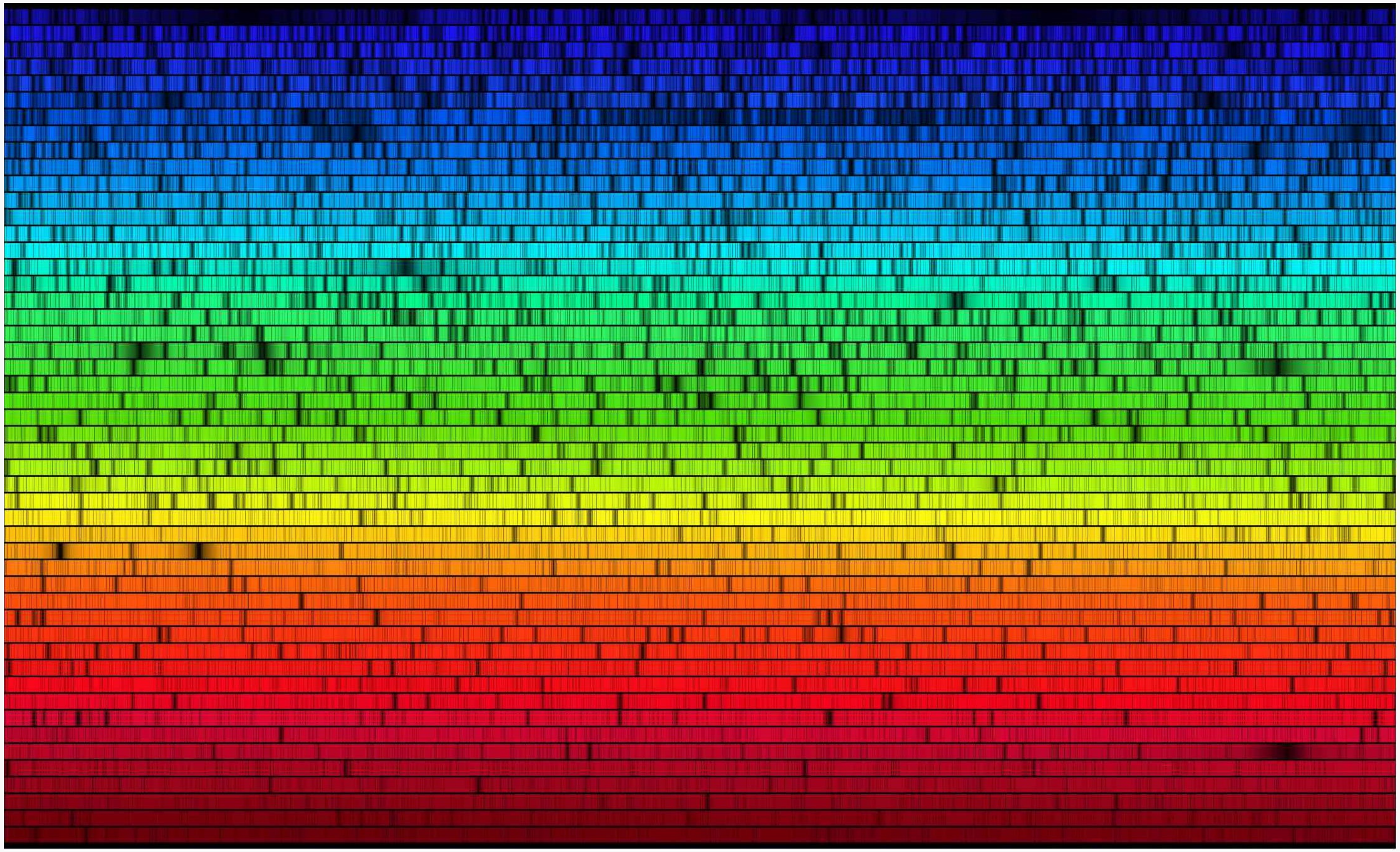


Great Opportunities = Great Challenges

- LSST will be photometric only, very little spectroscopic confirmation of object types!

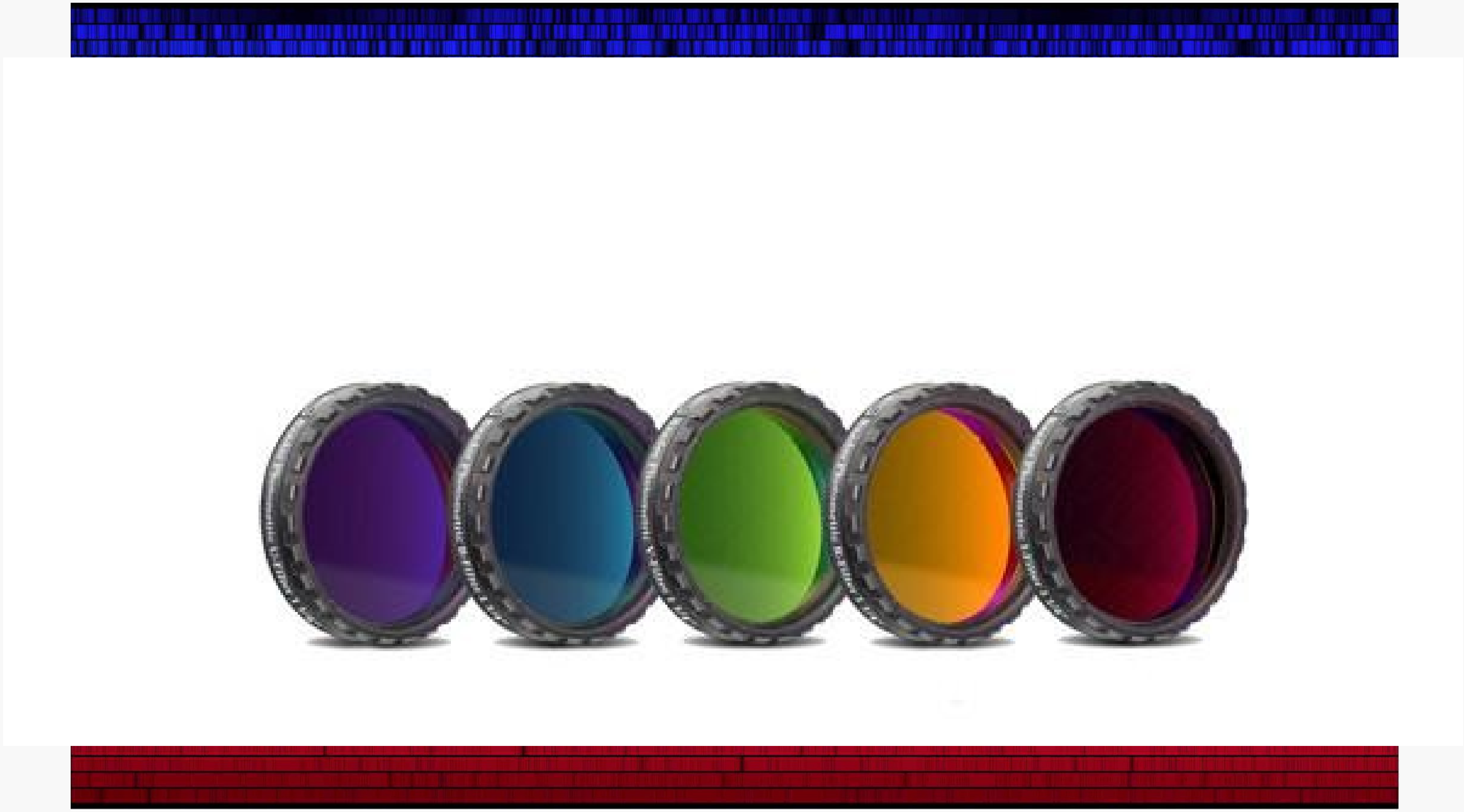
Great Opportunities = Great Challenges

- LSST will be photometric only, very little spectroscopic confirmation of object types!



Great Opportunities = Great Challenges

- LSST will be photometric only, very little spectroscopic confirmation of object types!



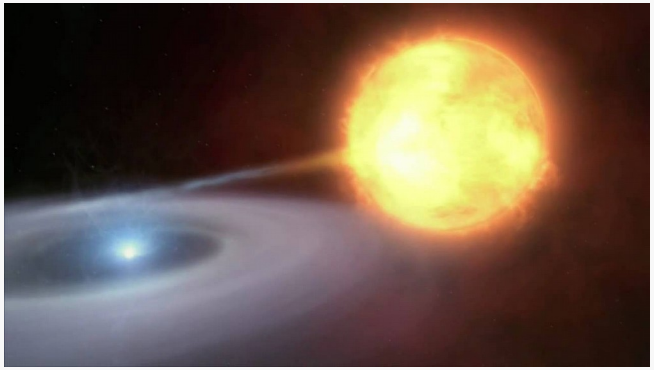
Photometric Classification of Supernovae

Photometric Classification of Supernovae

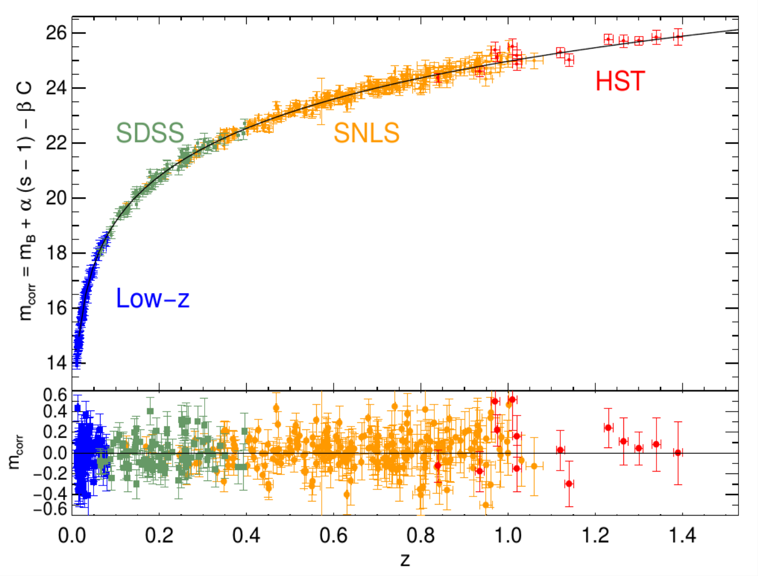
Firstly, what is a supernova?

Different Types of Supernovae

Type Ia supernovae



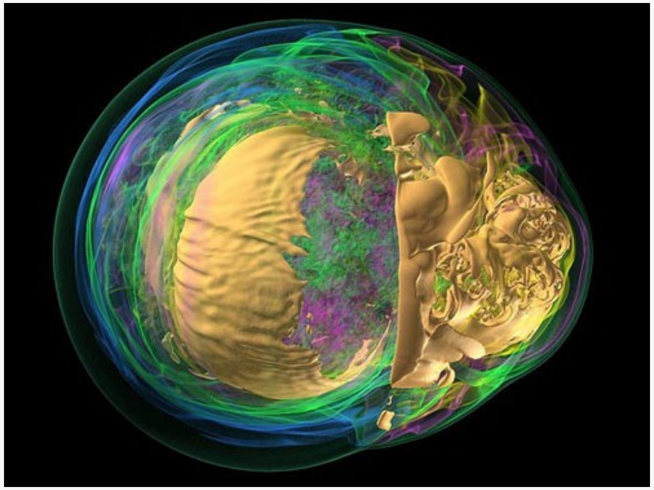
Cosmology



Conley et al. 2011

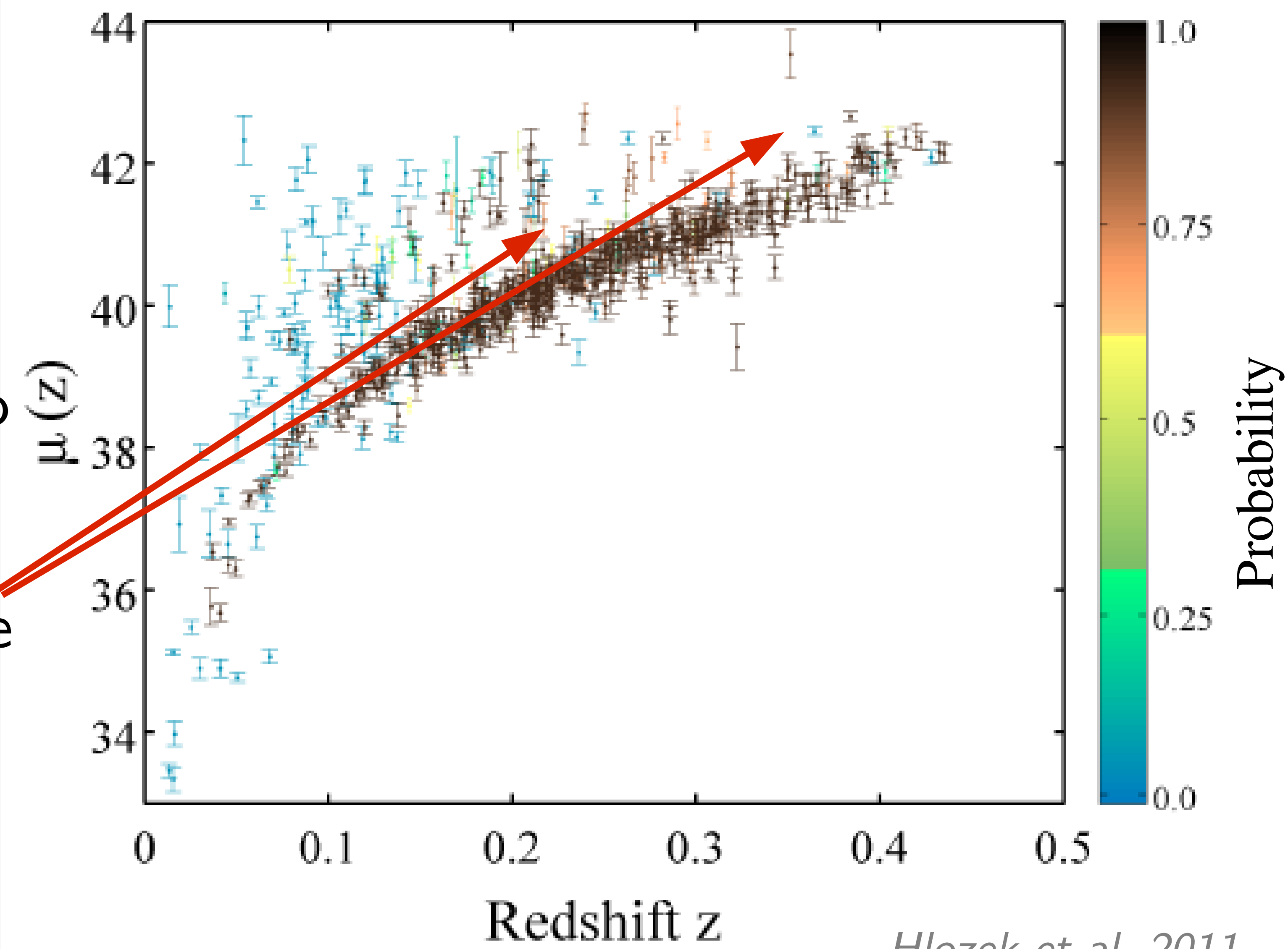
Core collapse supernovae

Supernova astrophysics



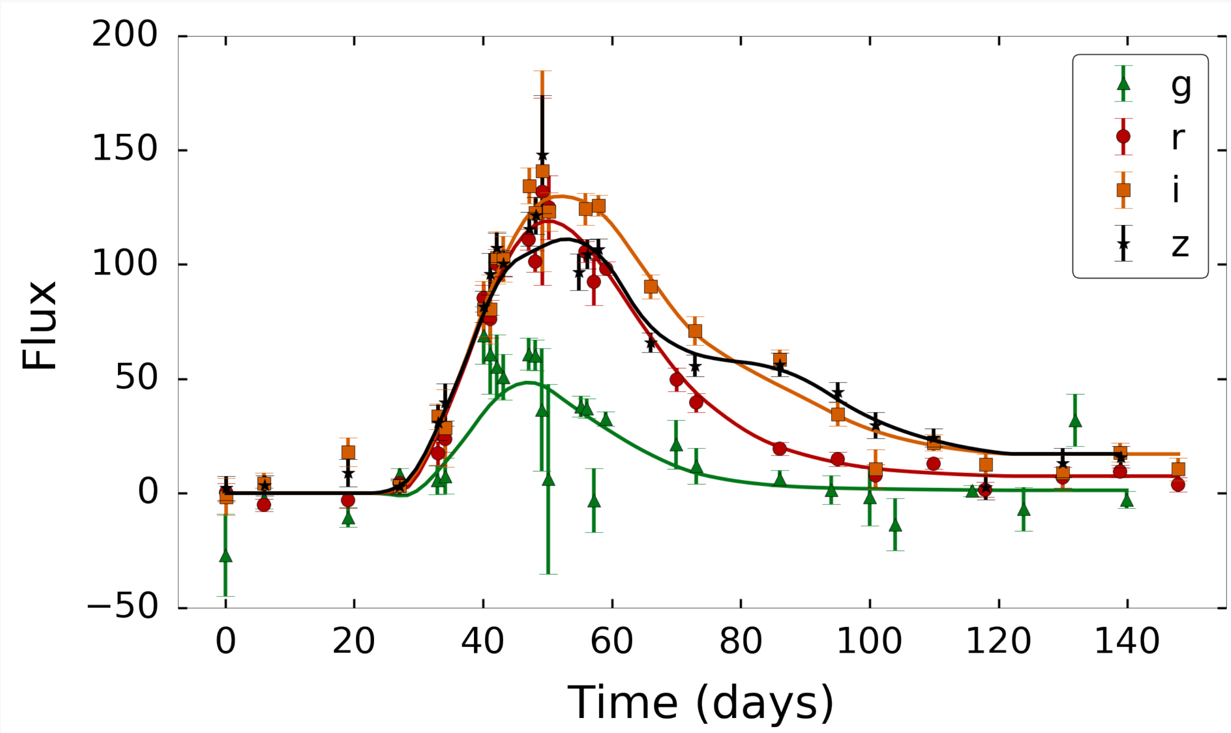
Non-Ia contamination is bad

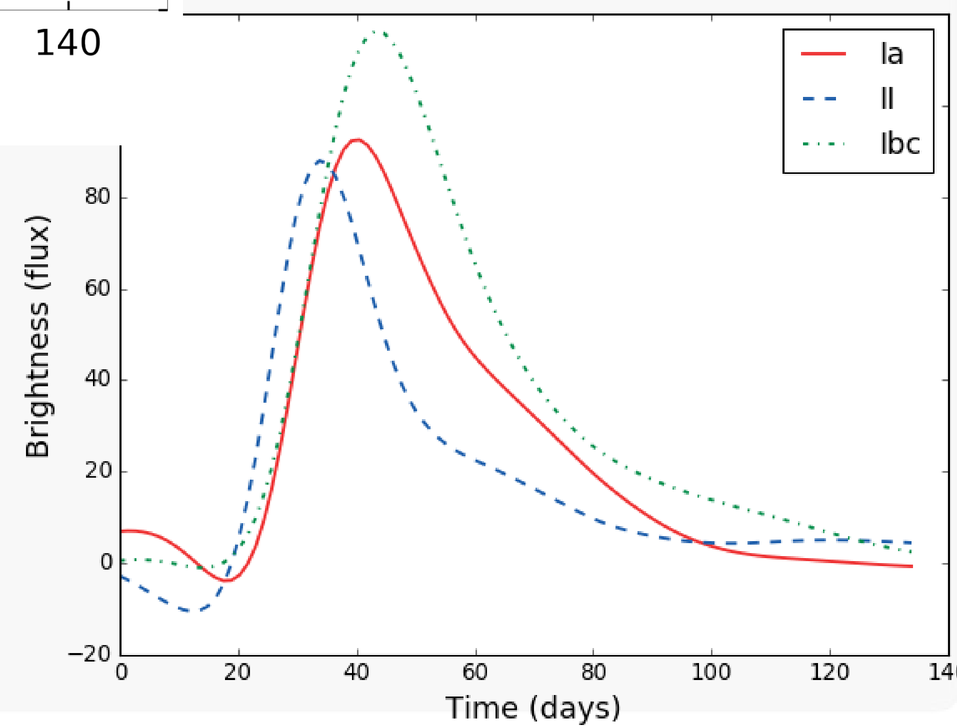
Non-Ia's tend to lie above the true cosmology line and bias the best fit.



Hlozek et al. 2011

Photometric Classification of Supernovae

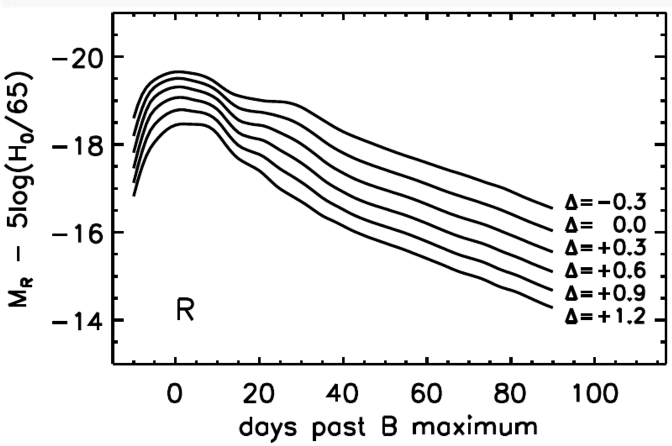


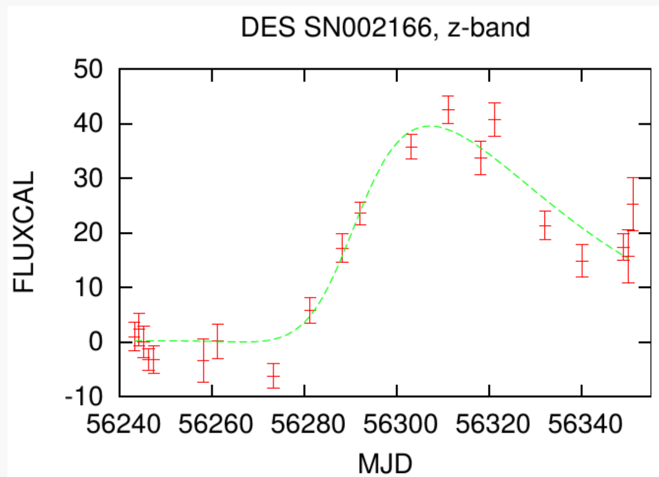


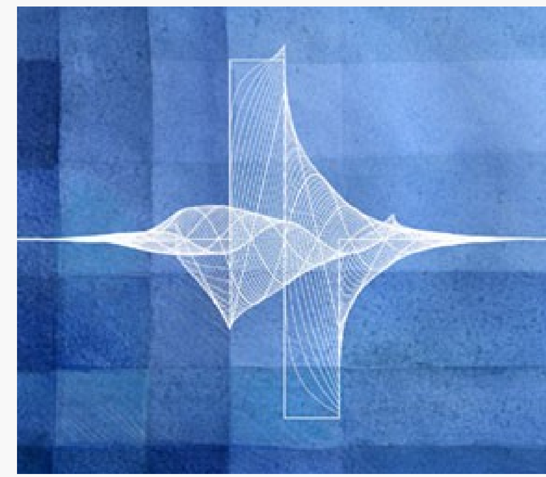
Lochner et al. (2016) 1603.00882

Feature selection

We've identified three promising approaches:







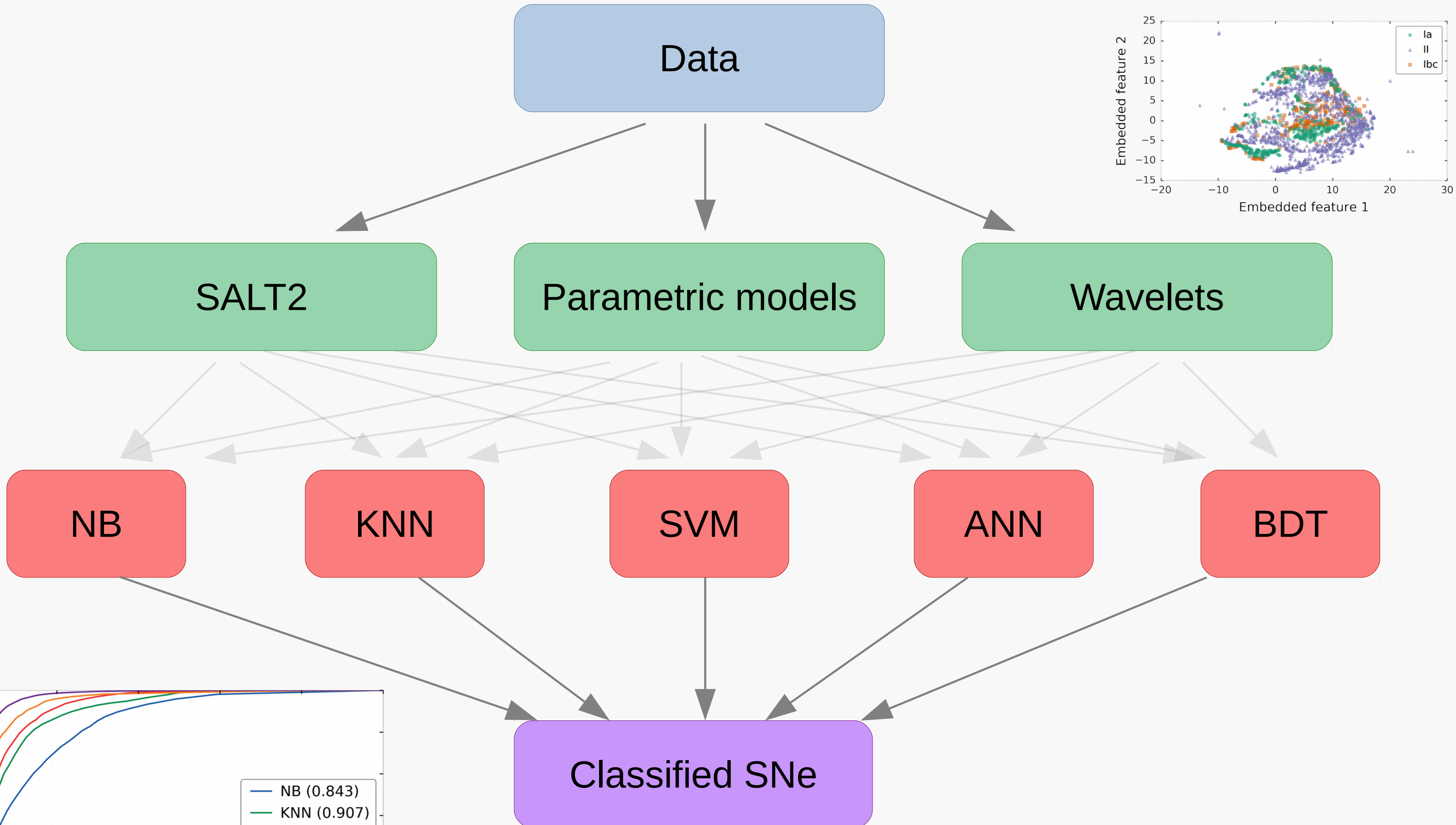
1) Template fitting

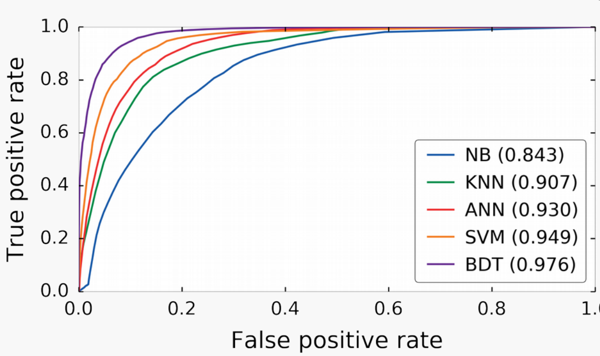
2) General light curve parameterisations

3) Wavelets

Model independence

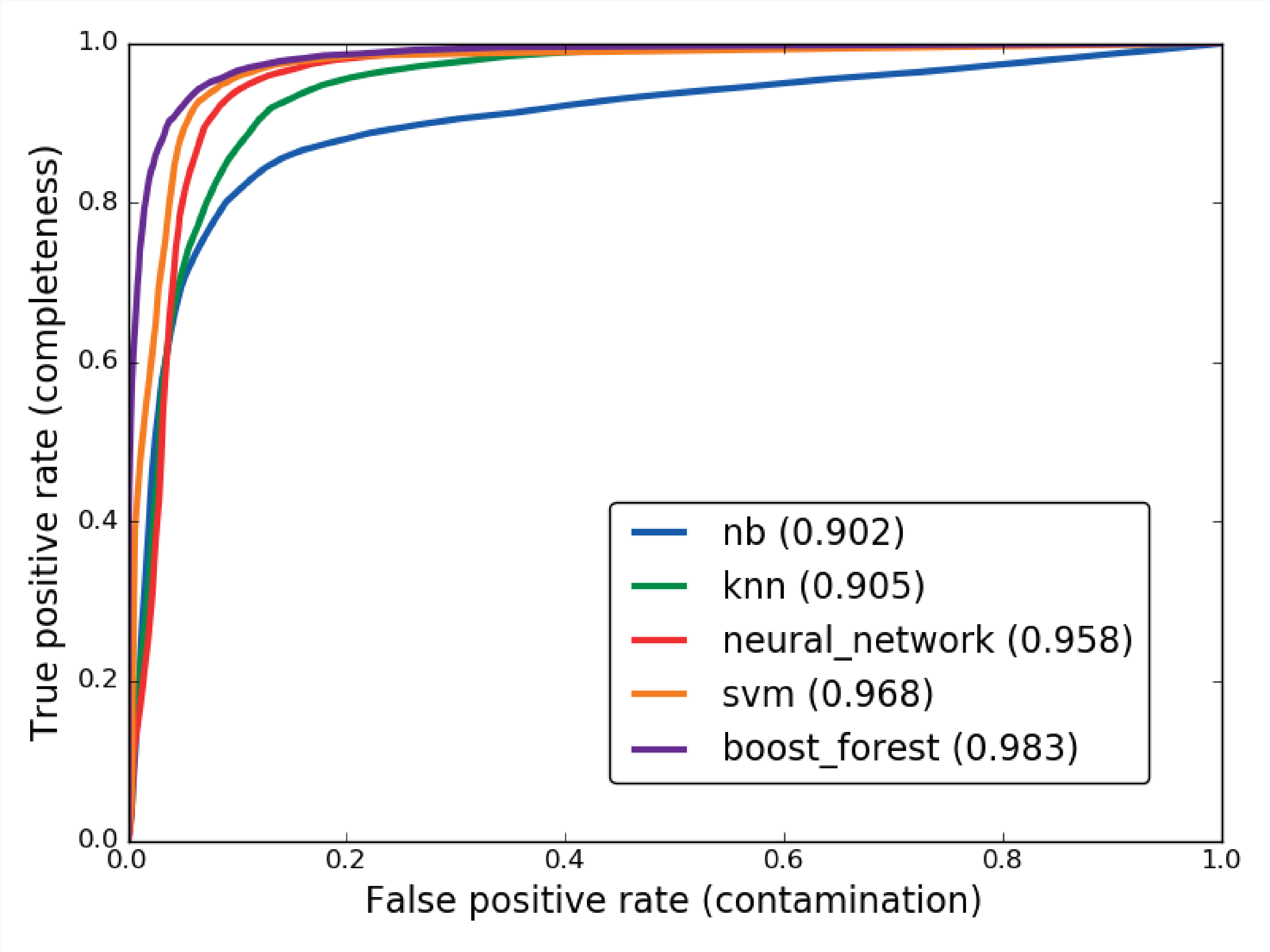
Pipeline





https://github.com/LSSTDESC/snmachine

Results



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

Question

Question

What is the difference between machine learning and statistics?

Machine Learning vs. Statistics

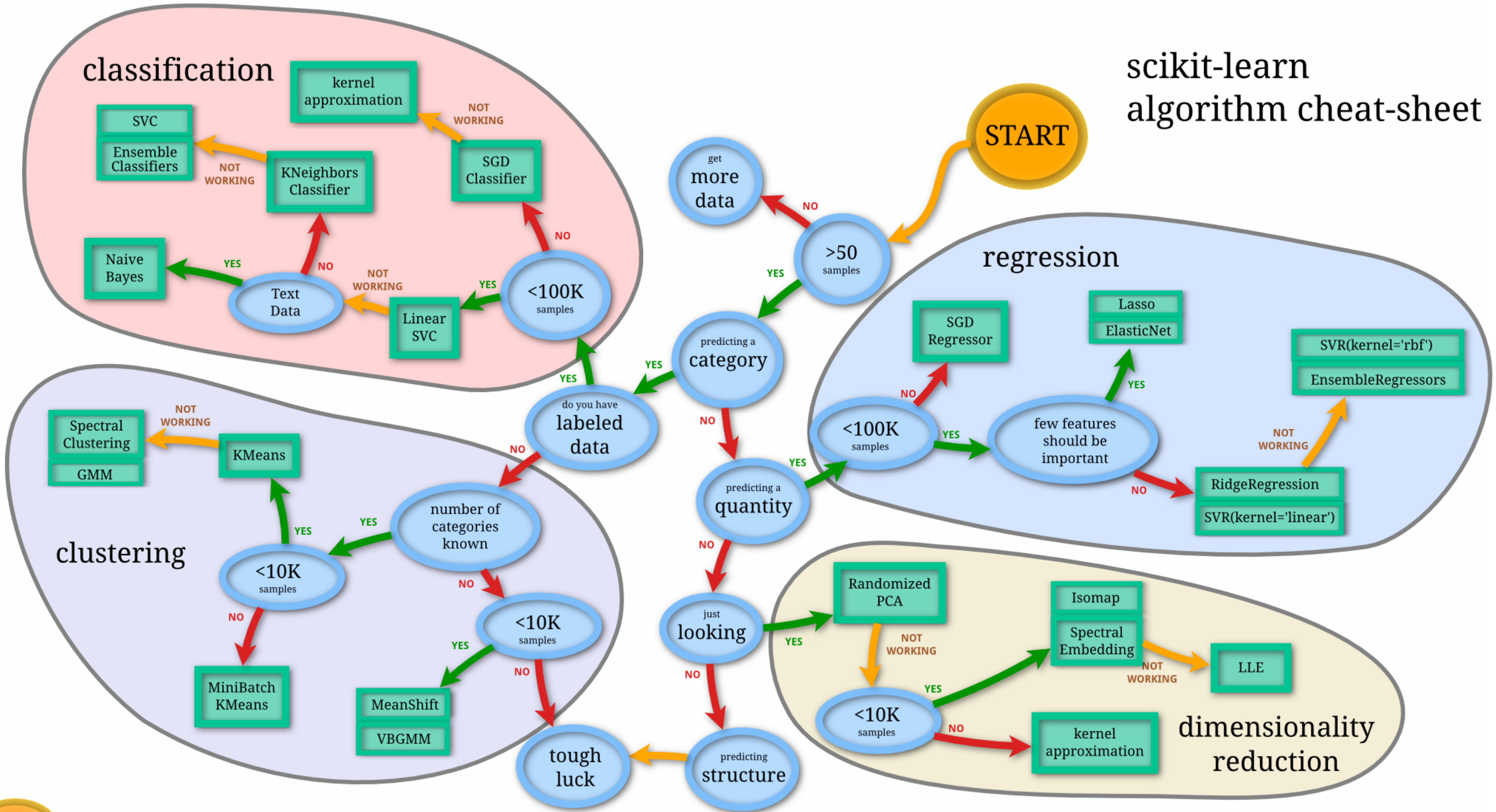
Decide whether machine learning or statistics is the appropriate approach to the following problems:

- Predicting the weather
- Testing for rare diseases (e.g. HIV test)
- Removing artifacts in astronomical images (e.g. CCD errors, aeroplanes etc.)
- Screening for tumours in radiology images
- Deciding which theory of gravity (e.g. Newton vs. Einstein) is correct

Important concepts

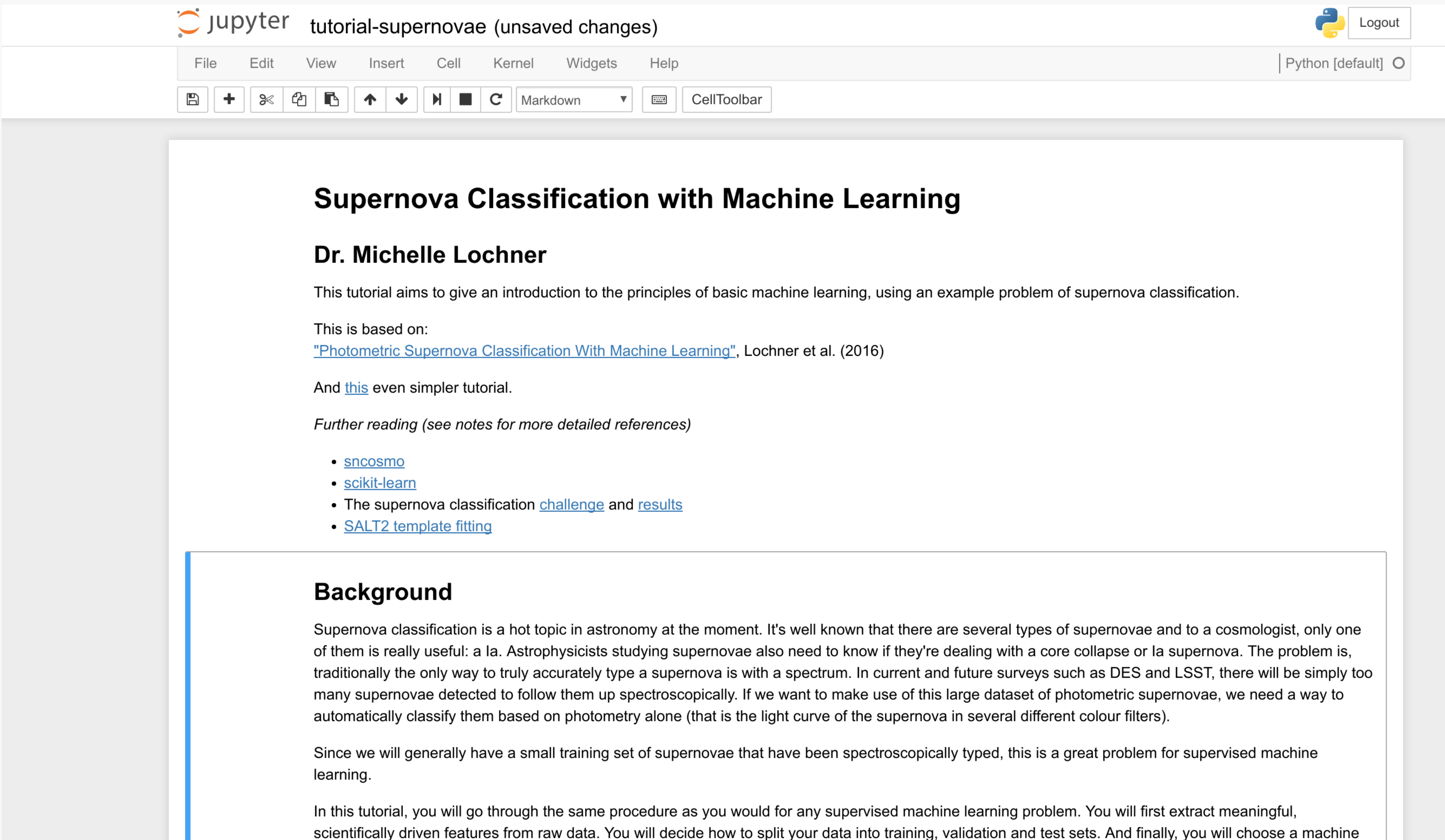
- Feature rescaling
- Overfitting
- Hyperparameters and cross validation
- Dimensionality reduction
- Representativeness (N.B!)
- Performance metrics

Tips and Tricks



The Tutorials

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


jupyter tutorial-supernovae (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Python [default]

Markdown

CellToolbar

Supernova Classification with Machine Learning

Dr. Michelle Lochner

This tutorial aims to give an introduction to the principles of basic machine learning, using an example problem of supernova classification.

This is based on:
"Photometric Supernova Classification With Machine Learning", Lochner et al. (2016)

And this even simpler tutorial.

Further reading (see notes for more detailed references)

- sncosmo
- scikit-learn
- The supernova classification challenge and results
- SALT2 template fitting

Background

Supernova classification is a hot topic in astronomy at the moment. It's well known that there are several types of supernovae and to a cosmologist, only one of them is really useful: a Ia. Astrophysicists studying supernovae also need to know if they're dealing with a core collapse or Ia supernova. The problem is, traditionally the only way to truly accurately type a supernova is with a spectrum. In current and future surveys such as DES and LSST, there will be simply too many supernovae detected to follow them up spectroscopically. If we want to make use of this large dataset of photometric supernovae, we need a way to automatically classify them based on photometry alone (that is the light curve of the supernova in several different colour filters).

Since we will generally have a small training set of supernovae that have been spectroscopically typed, this is a great problem for supervised machine learning.

In this tutorial, you will go through the same procedure as you would for any supervised machine learning problem. You will first extract meaningful, scientifically driven features from raw data. You will decide how to split your data into training, validation and test sets. And finally, you will choose a machine

Four Tutorials to Choose From!

tutorial-basic.ipynb – Very basic recap of concepts covered

tutorial-supernovae.ipynb – The main tutorial using a classification example, I'd like everyone to finish this one!

tutorial-galaxies.ipynb – I provide some data for a regression problem and very little guidance.

tutorial-deep-learning.ipynb – Simple deep learning example to get you started

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/

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

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

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.

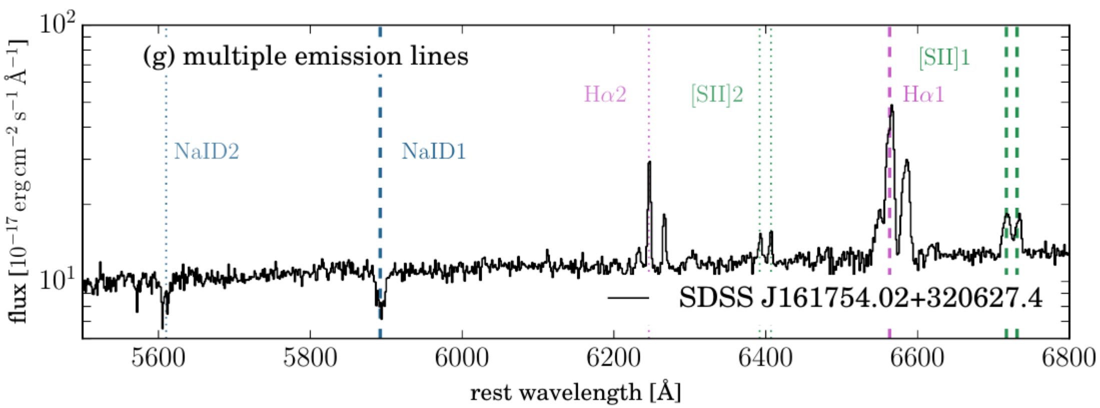
Backpropagation

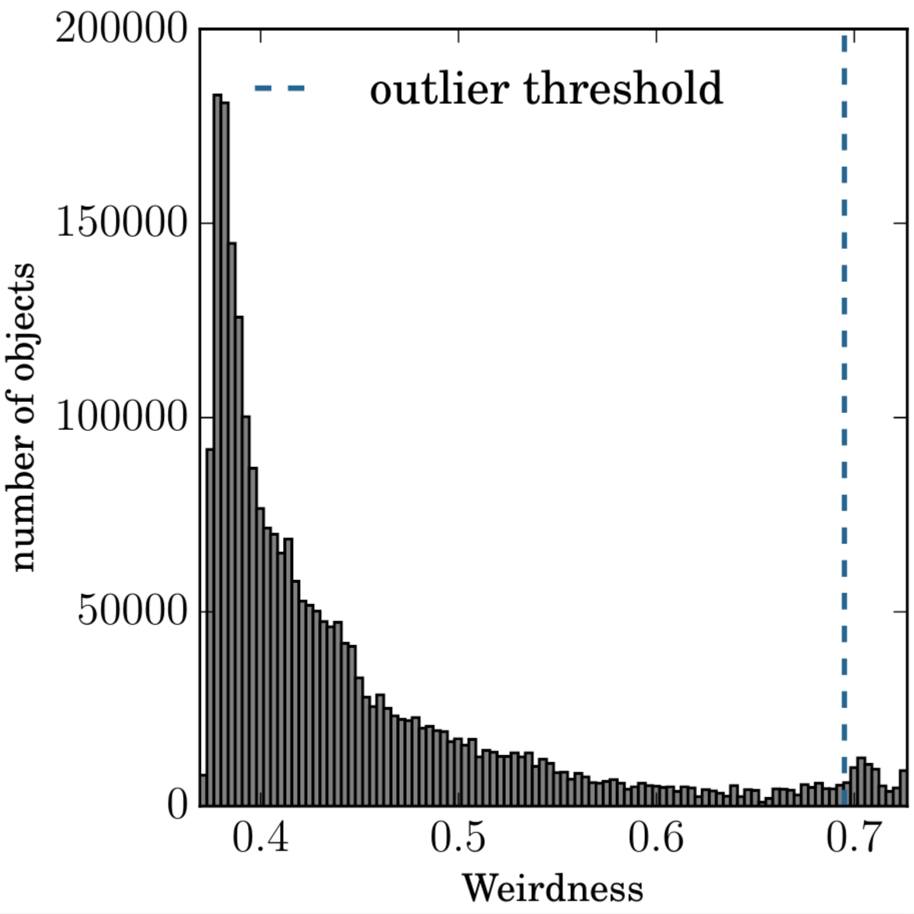
You'll find the weights at the *beginning* of the network depend on the weights at the *end*, which is why backpropagation starts at the end of the network and works backwards.

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.

Examples of Machine Learning

The weirdest SDSS galaxies: results from an outlier detection algorithm

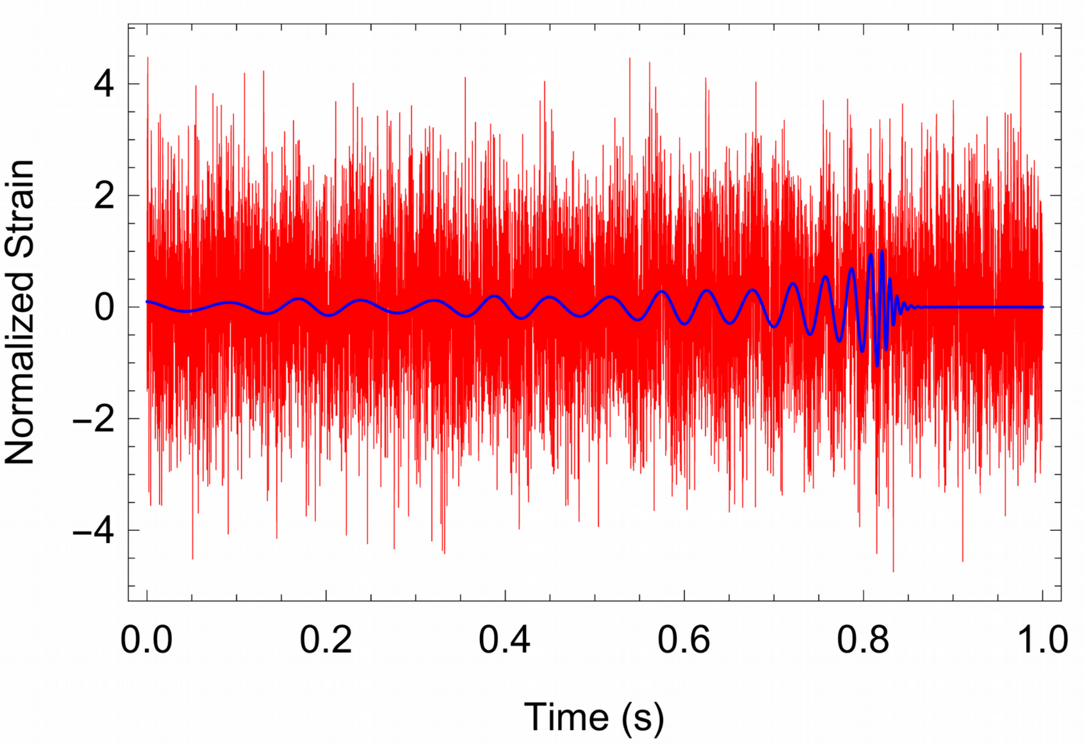


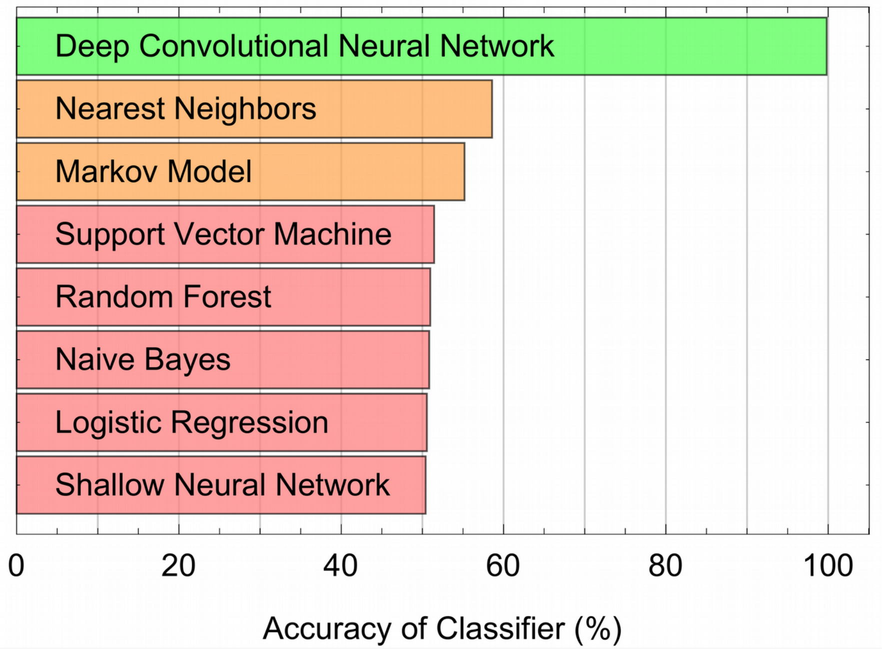


Baron & Poznanski (2016) http://arxiv.org/abs/1611.07526v1

Examples of Machine Learning

Deep Neural Networks to Enable Real-time Multimessenger Astrophysics





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