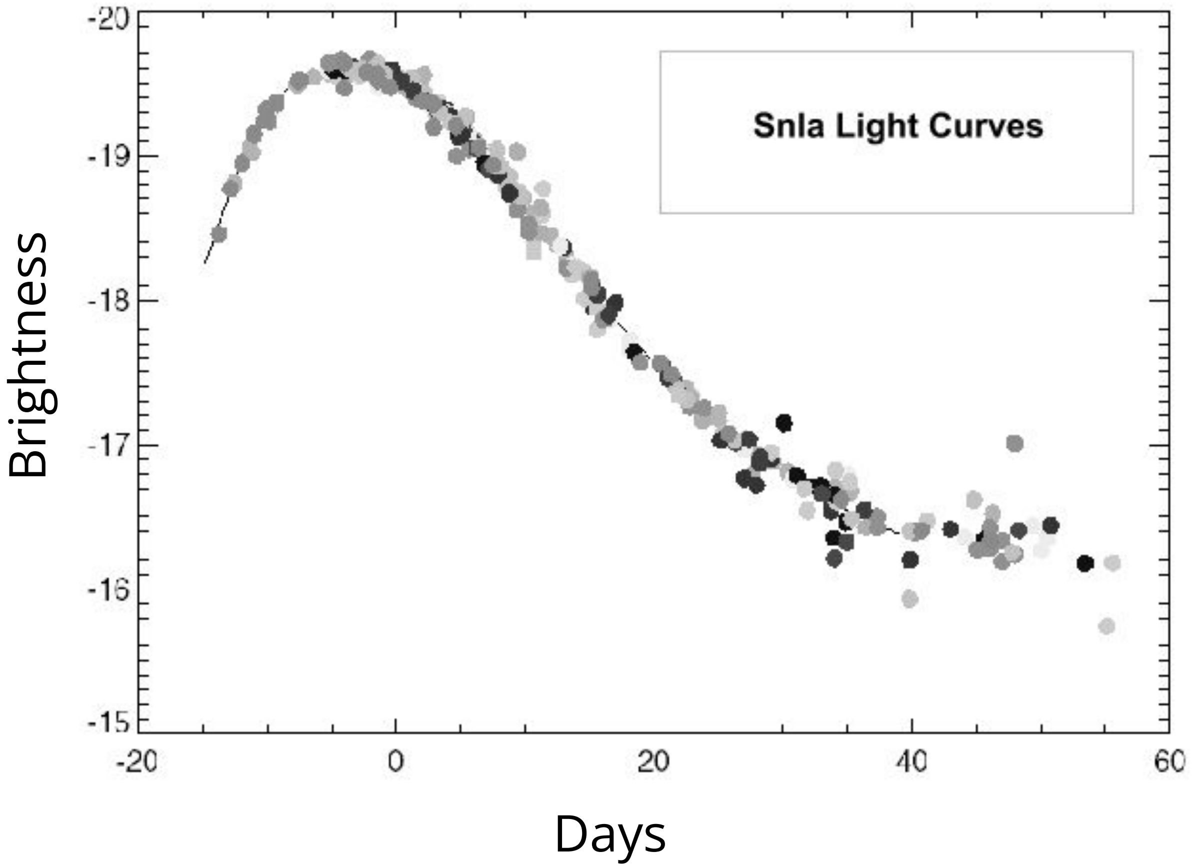
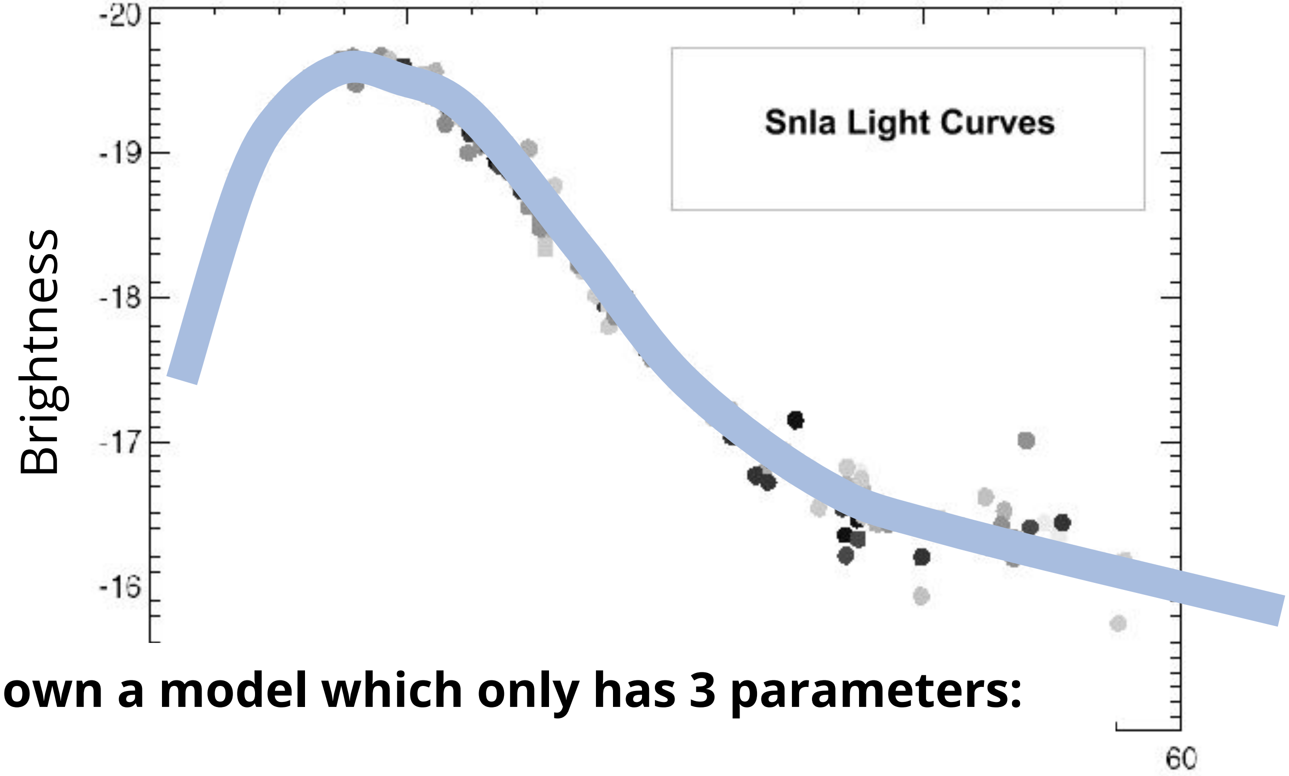
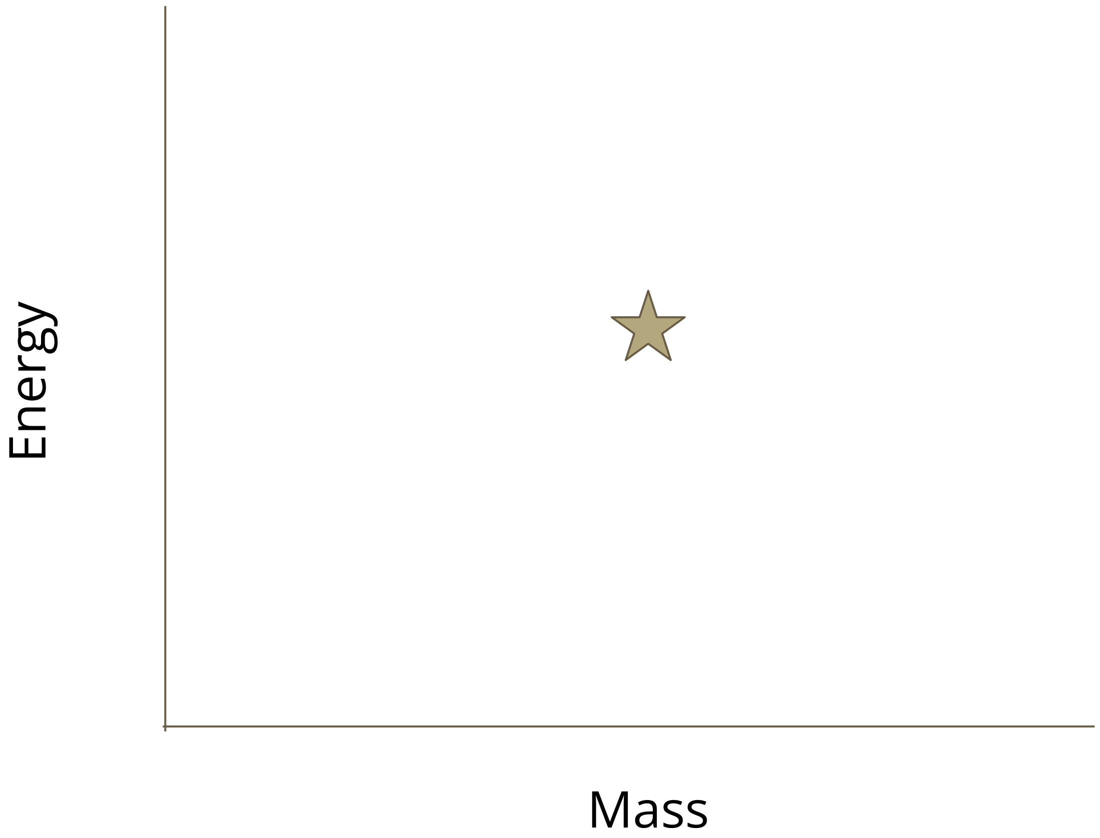
Dimensionality Reduction, Representation Learning and Autoencoders

March 9, 2022

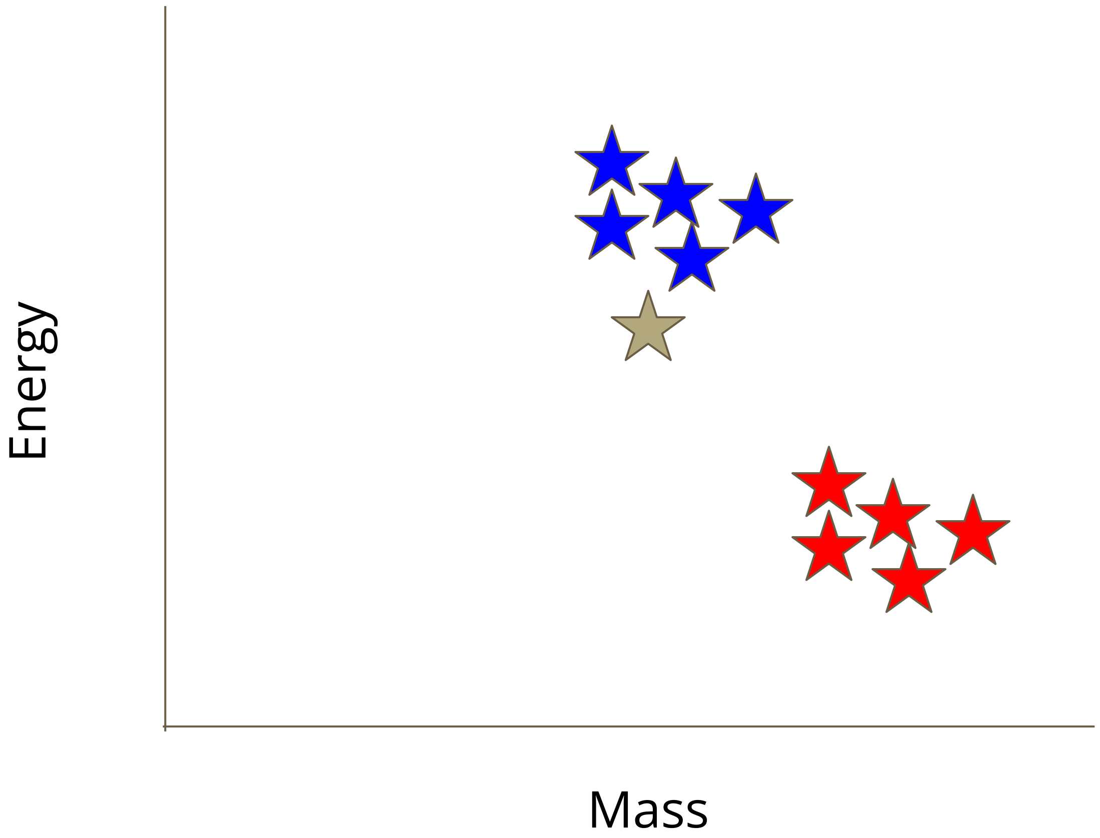

SnIa Light Curves
Brightness
Days
-20
-19
-18
-17
-16
-15
-20
0
20
40
60



I can write down a model which only has 3 parameters:
Energy,
Mass,
Amount of radioactive material

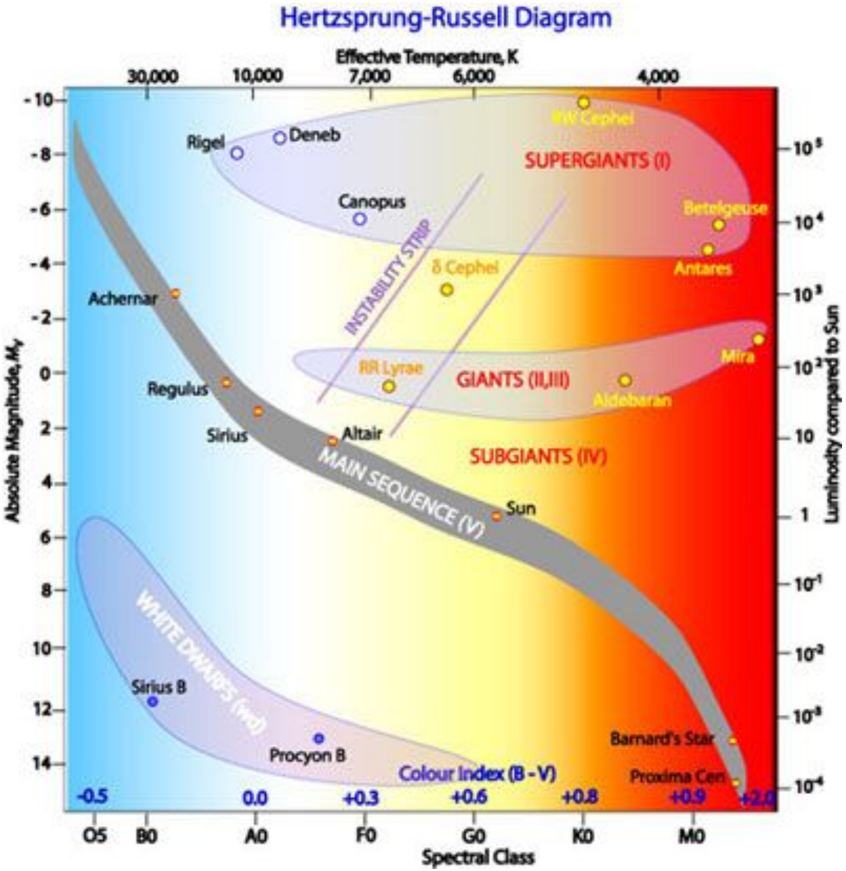


Energy
Mass


Energy
Mass

Let’s call this space a “latent” space. Why “latent”?

Why do we care about latent spaces?

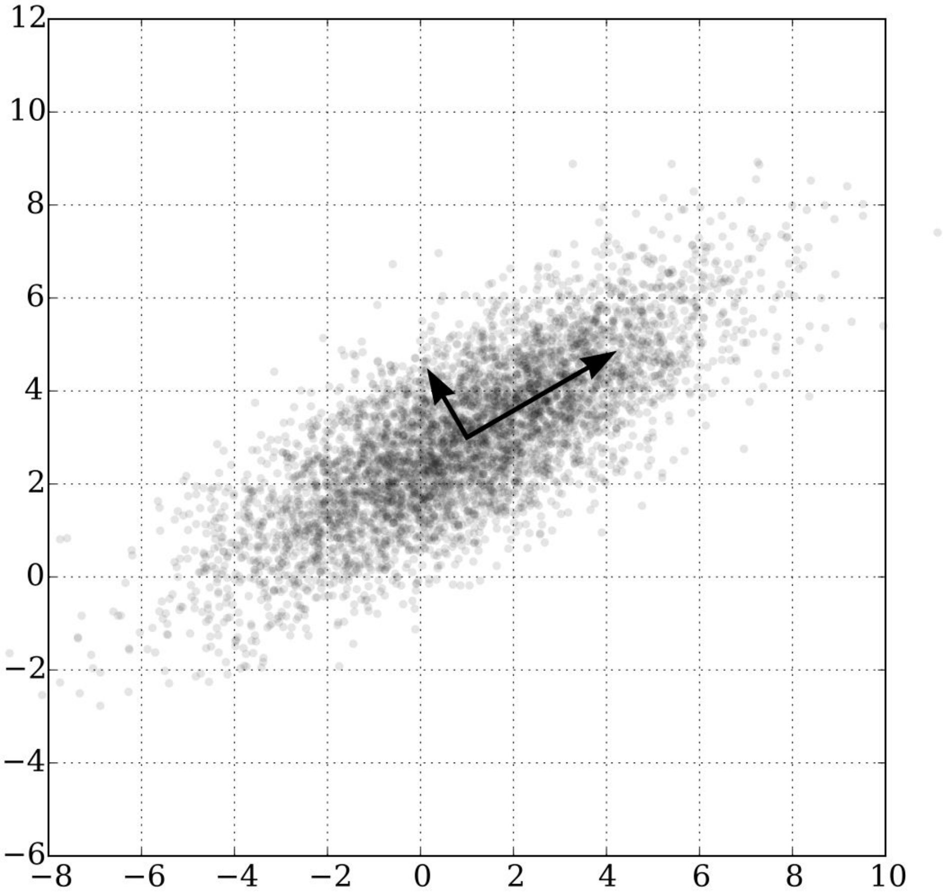


Once our data is in a low-dimensional space, we can complete a number of tasks:

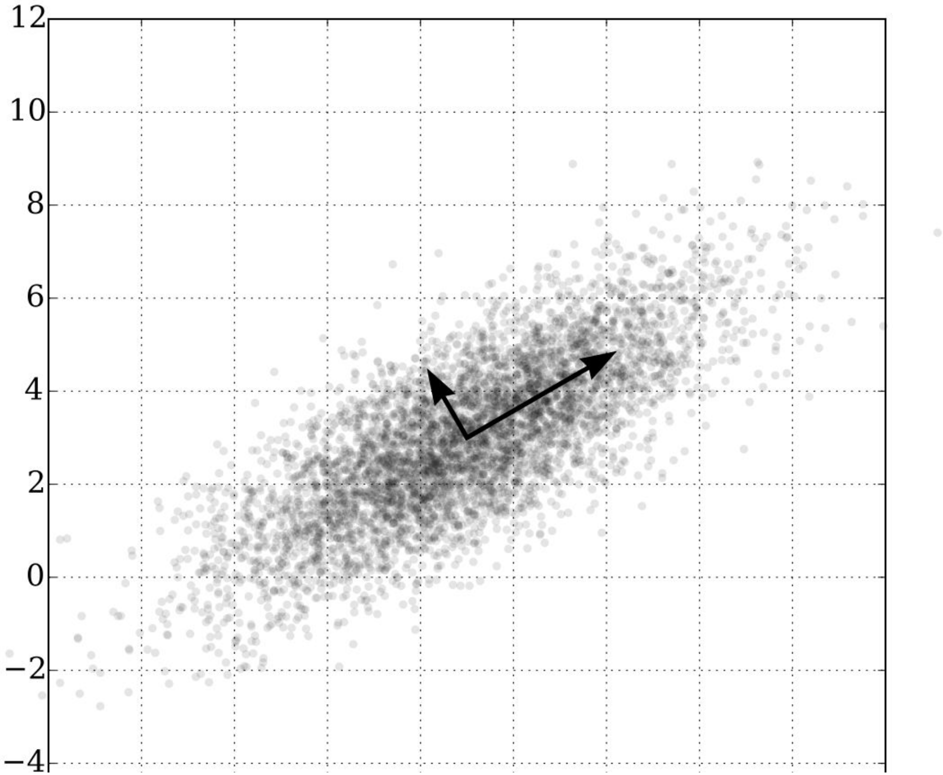
1. Better feature selection for classification
2. Anomaly detection
3. Data simulations
4. Physical interpretability

How can we create a low-dimensional representation without directly fitting a physical model?

The simplest solution is to break down data into **basis vectors**

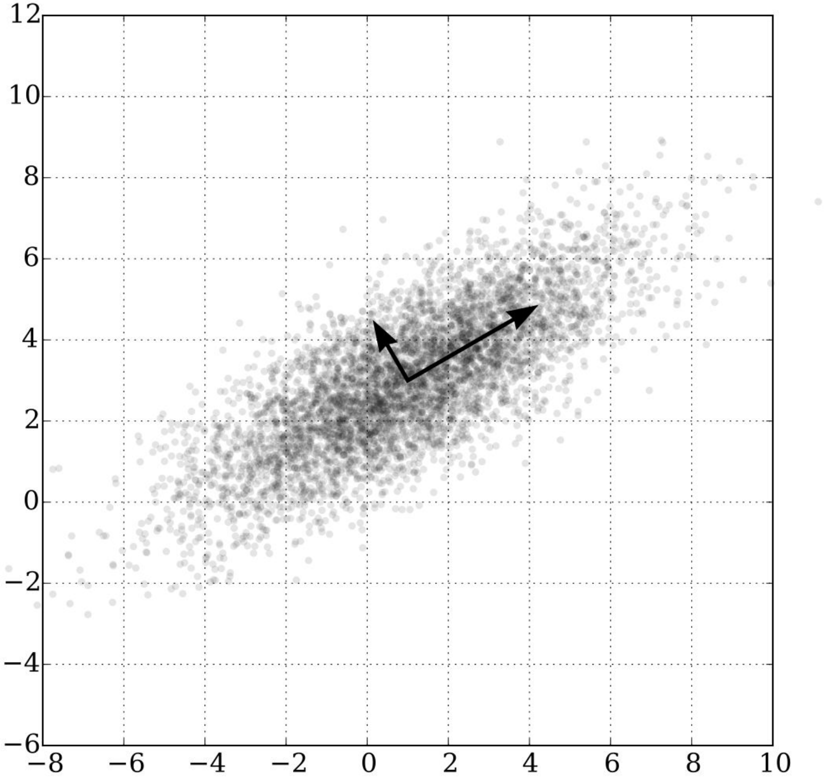


The simplest solution is to break down data into **basis vectors**

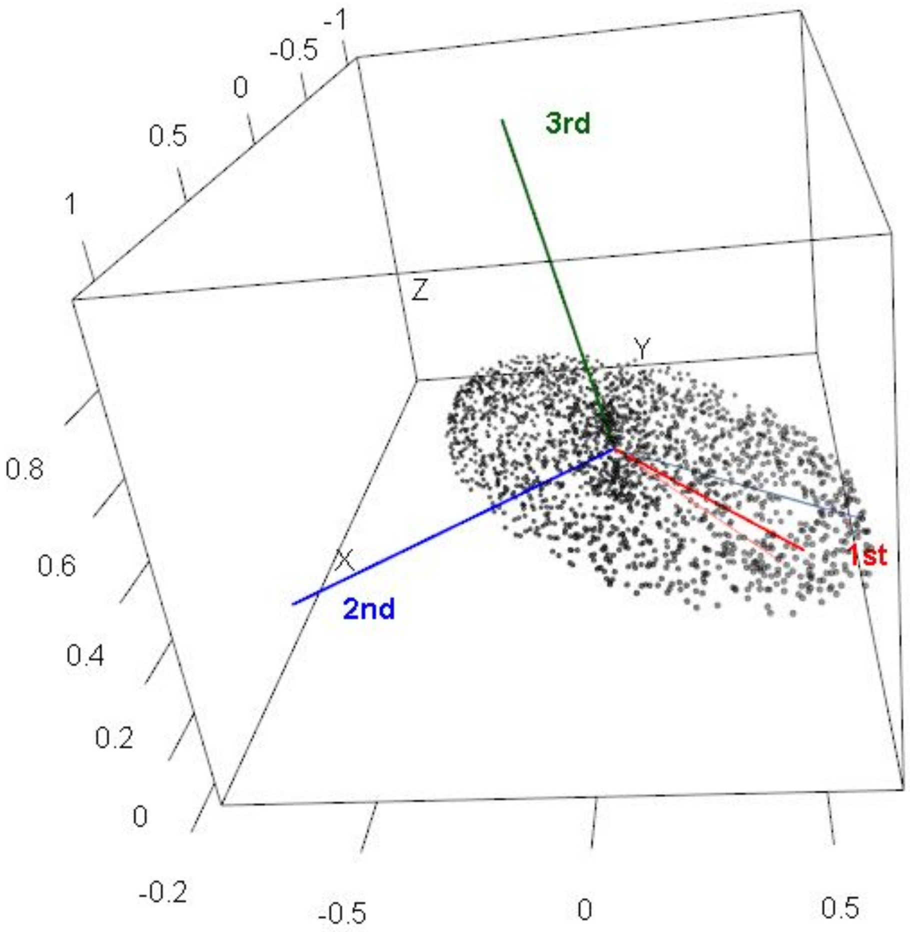


An eigenbasis is one with orthogonal, normalized vectors

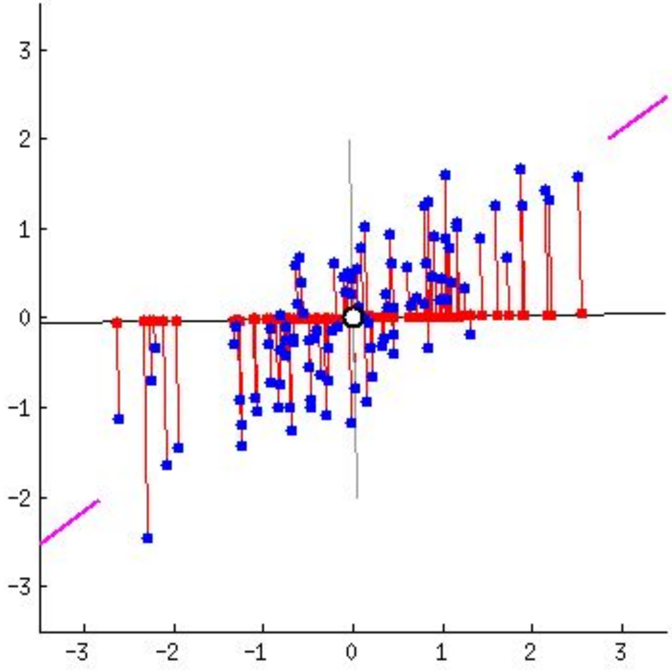
In this 2D example, every point is exactly described by the sum of two eigenvectors



In higher dimensions, 2 detectors may describe ‘most’ of the variance within our data



Each observed datapoint can be projected onto a basis vector

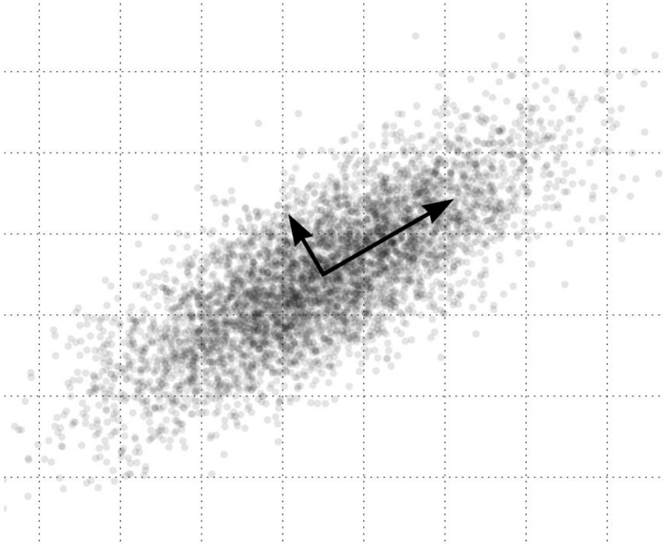


Principal Component Analysis (roughly) has the following steps:

- Find the eigenvectors of the dataset

Principal Component Analysis (roughly) has the following steps:

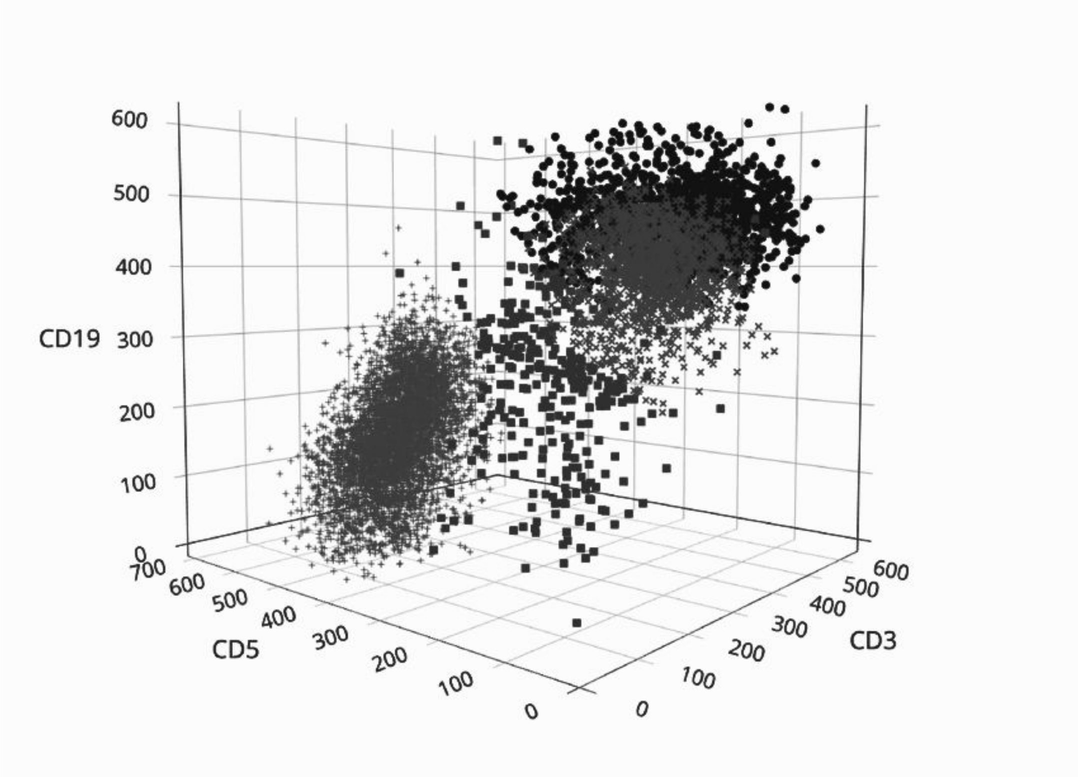
- Find the eigenvectors of the dataset
- Sort the eigenvectors by explained variance
    - Which ones explain the majority of the scatter within the data?

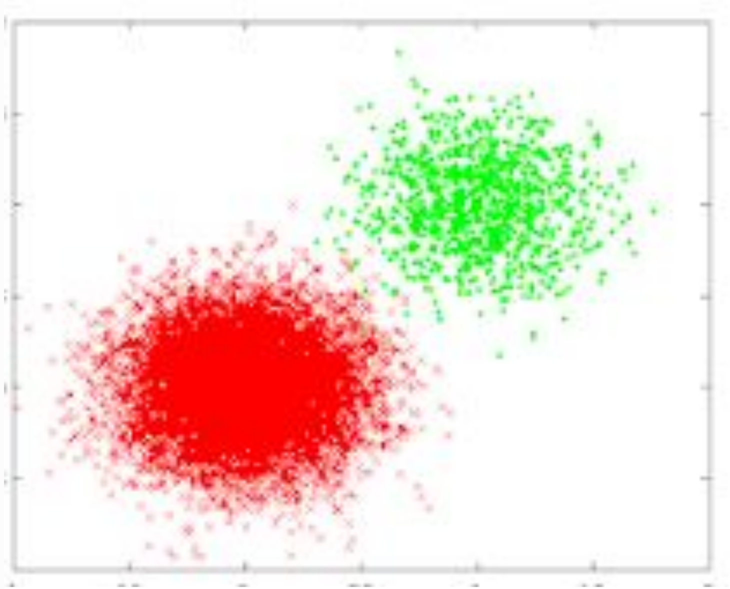


Principal Component Analysis (roughly) has the following steps:

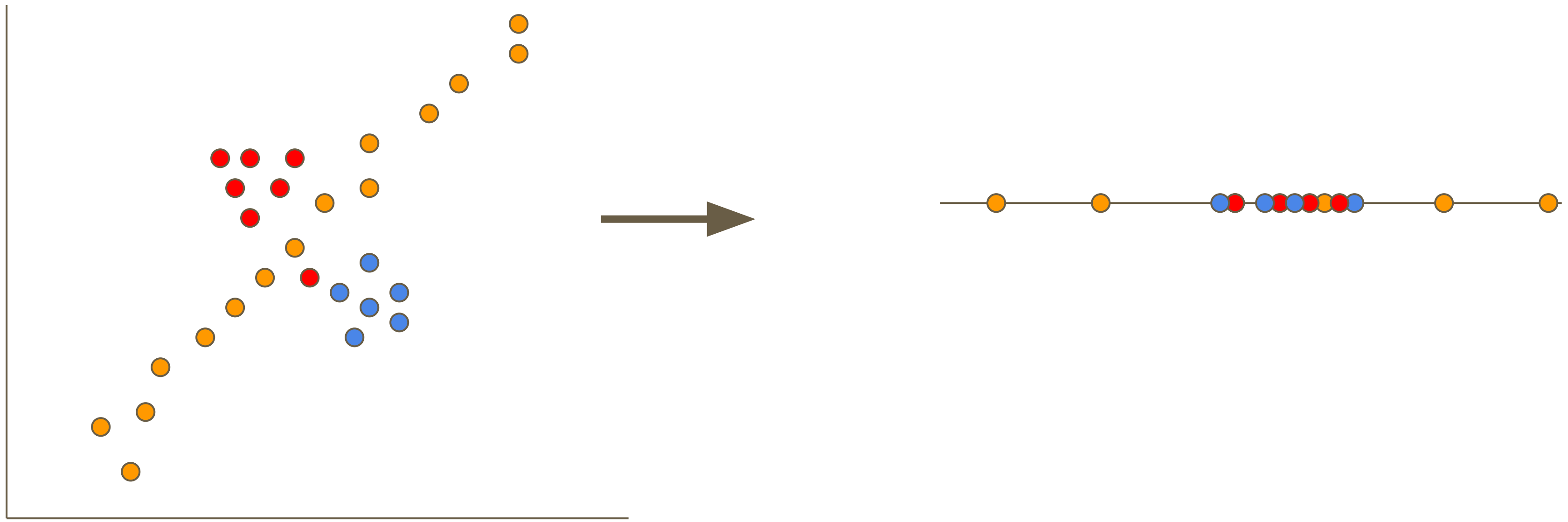
- Find the eigenvectors of the dataset
- Sort the eigenvectors by explained variance
- Project the data onto each basis, tracking the weights
    - For N-dimensional data, each point should be exactly described with N-vectors. So we want to grab the M vectors which describe most of the variance in our data, with M<N

PCA transforms our high-dimensional observed space, to a low-dimension **latent space**





PCA is limited by it linear nature

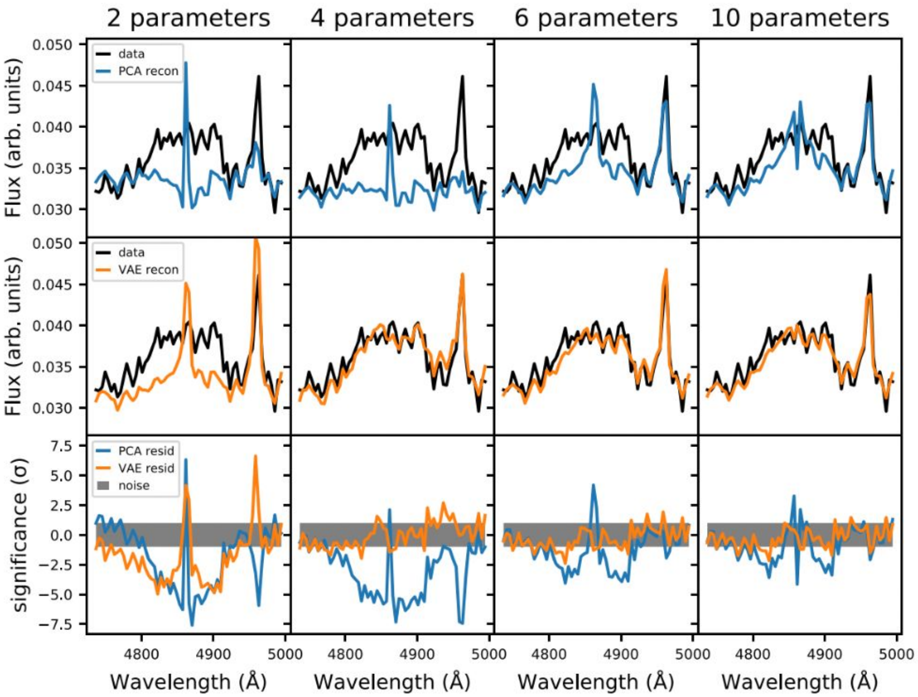


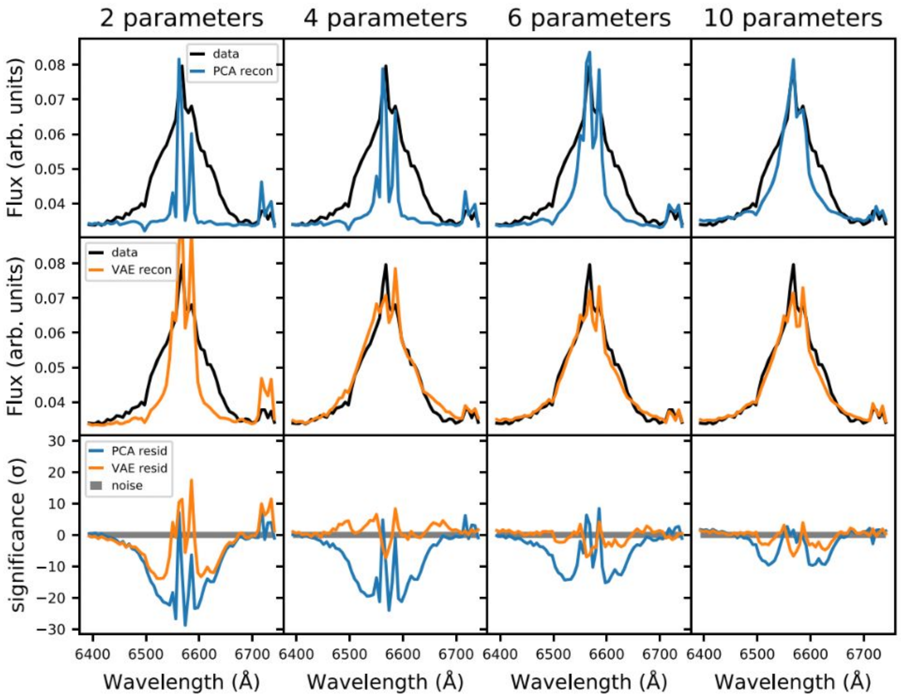
Group chat - discuss the following questions

1. What does it mean to have a ‘linear’ transformation of our data?
2. Can you give an example of a non-linear transformation in astronomical data?

PCA is limited in its linear assumption, but other, more sophisticated, methods exist as well

If a nonlinear transformation affects our data, PCA is insufficient





Portillo et al 2020

Instead, we will use a non-linear method: an autoencoder

What is an autoencoder?

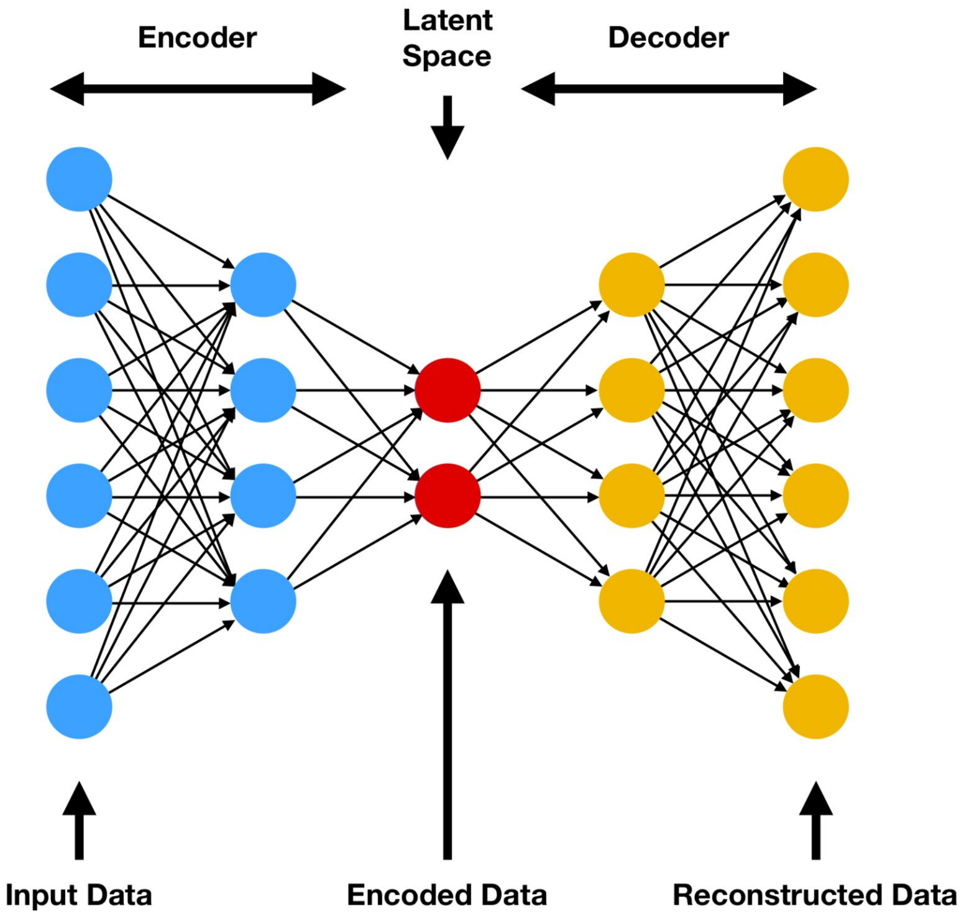
An autoencoder is a type of neural network which we will use to reduce the dimensionality of our data.

It is one example of a larger class of “representation learning” in deep learning

How to fit a model using MOO

1. **Model** to fit the data (e.g. physics).
2. **Objective Function** (or 'loss/cost function') which is a metric that you will choose to quantify how well the model fits the data (e.g. chi-squared).
3. **Optimization Method** which you will use to find the best model (e.g. gradient descent).

Autoencoder Architecture



Objective function

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

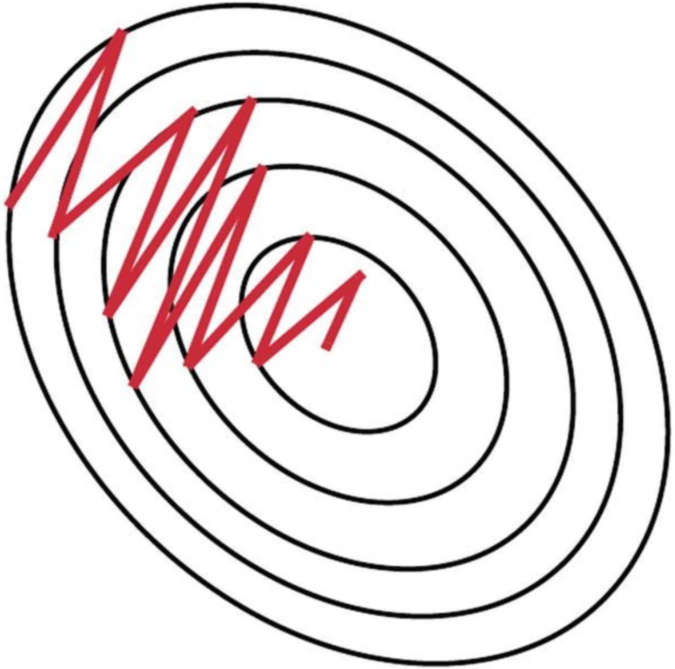
MSE = mean squared error

n = number of data points

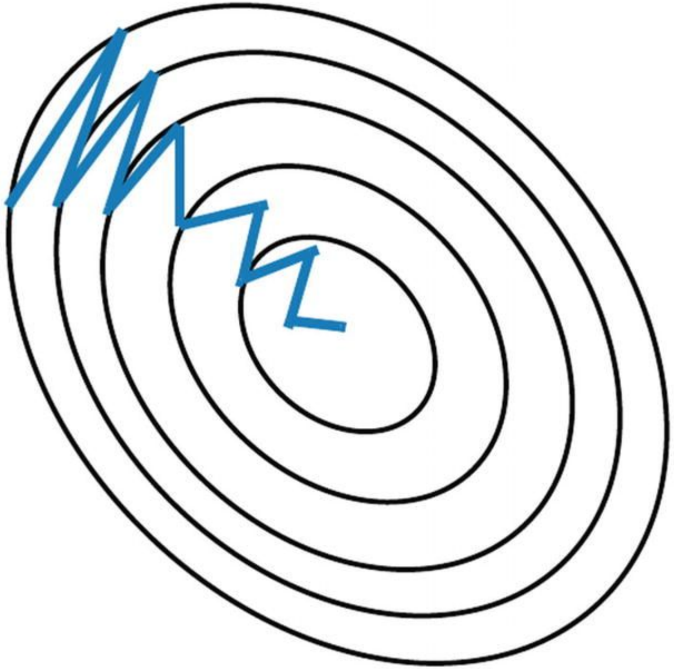
Y_i = observed values

$\hat{Y}_i$ = predicted values

We will still optimize with gradient descent!



Stochastic Gradient Descent **withhout** Momentum



Stochastic Gradient Descent **with** Momentum

Adam algorithm

$$m_t = \boxed{\beta_1} * m_{t-1} + (1 - \beta_1) * \nabla w_t$$

$$v_t = \boxed{\beta_2} * v_{t-1} + (1 - \beta_2) * (\nabla w_t)^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \qquad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

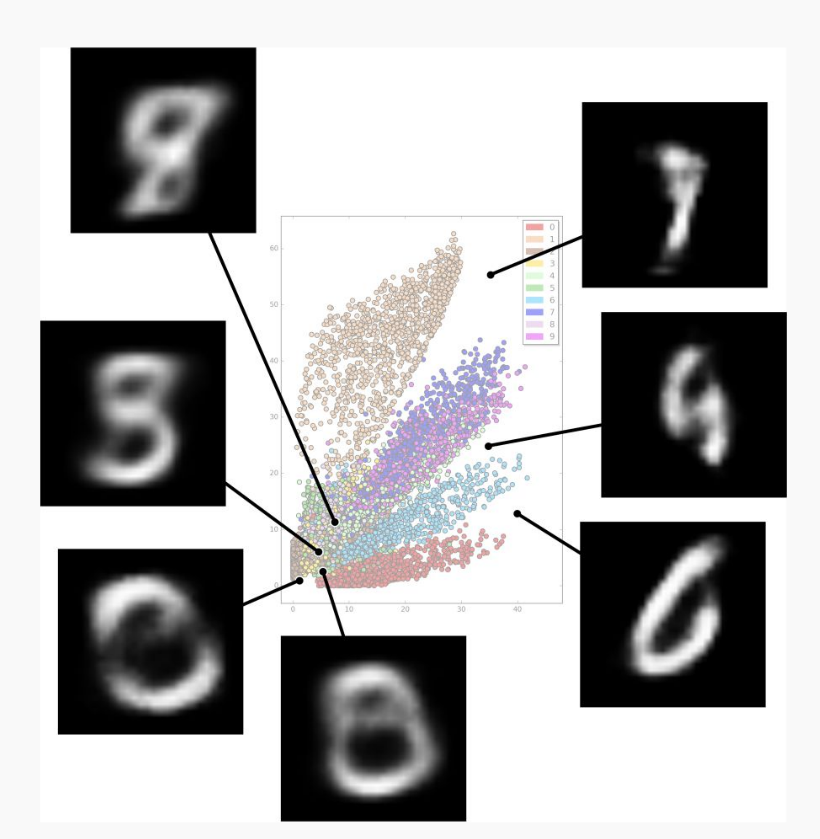
$$w_{t+1} = w_t - \frac{\boxed{\eta}}{\sqrt{\hat{v}_t + \epsilon}} * \hat{m}_t$$

η = Learning Rate

β1 = 0.9

β2 = 0.999

Using autoencoders for anomaly detection



Specifically, we'll use an isolation forest

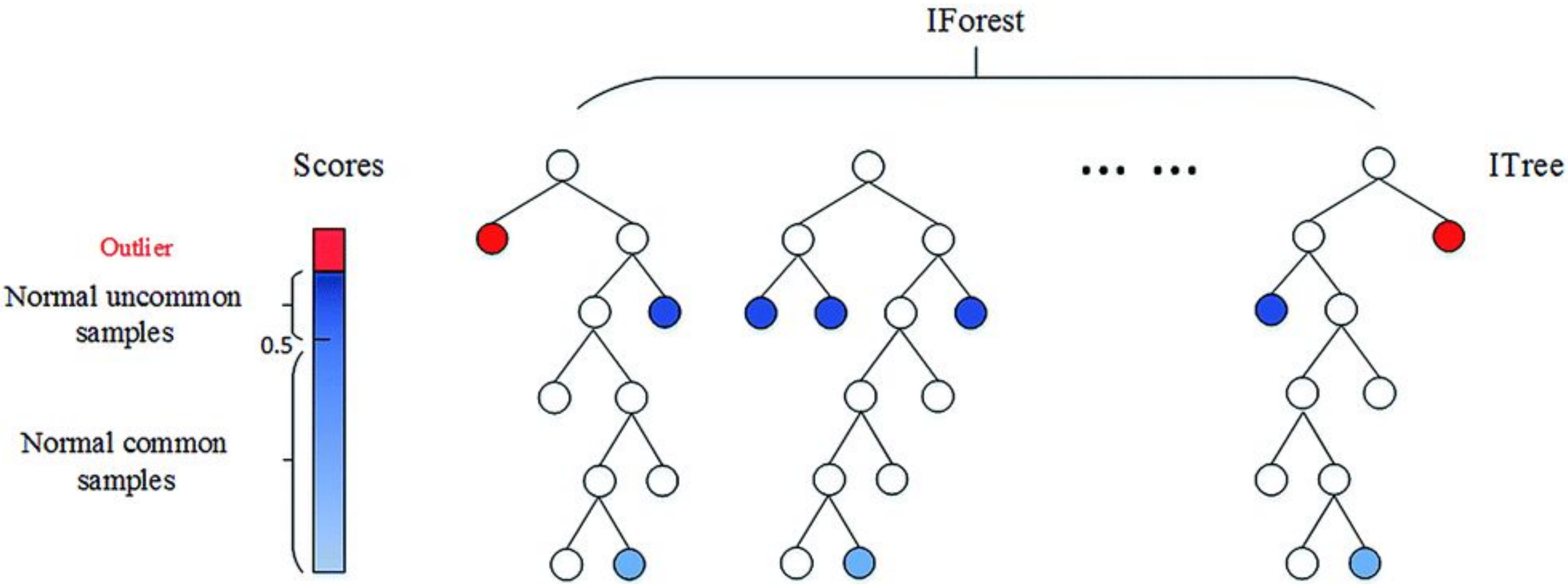
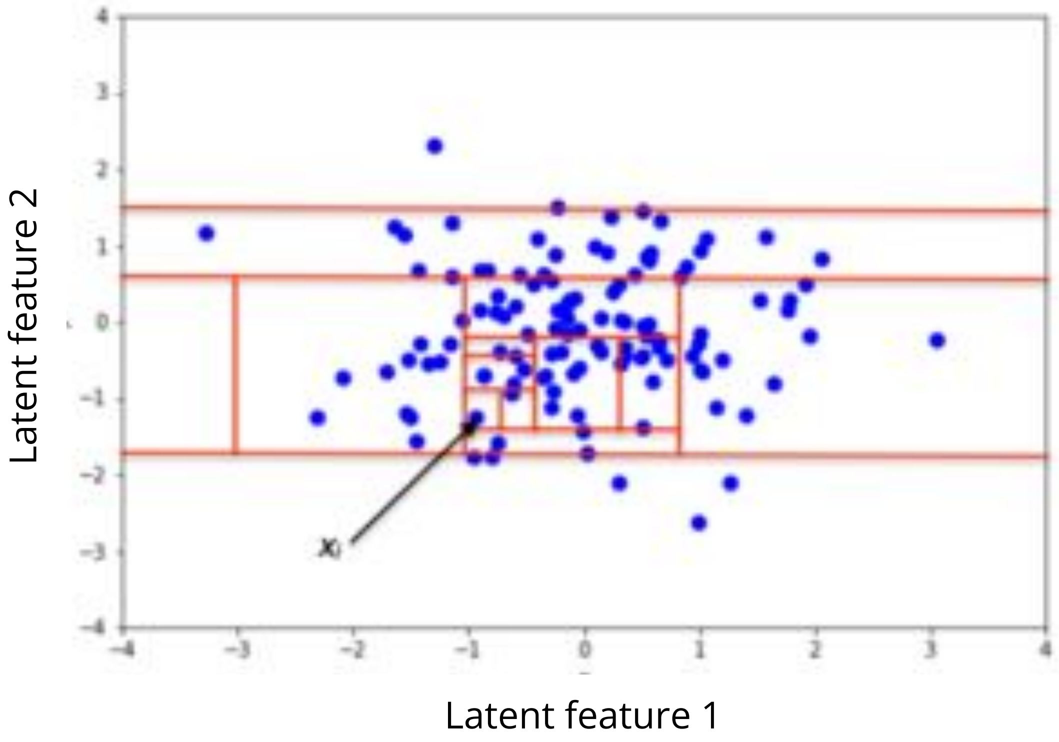


image: https://donghwa-kim.github.io/iforest.ht

Specifically, we'll use an isolation forest



Modification: variational autoencoders

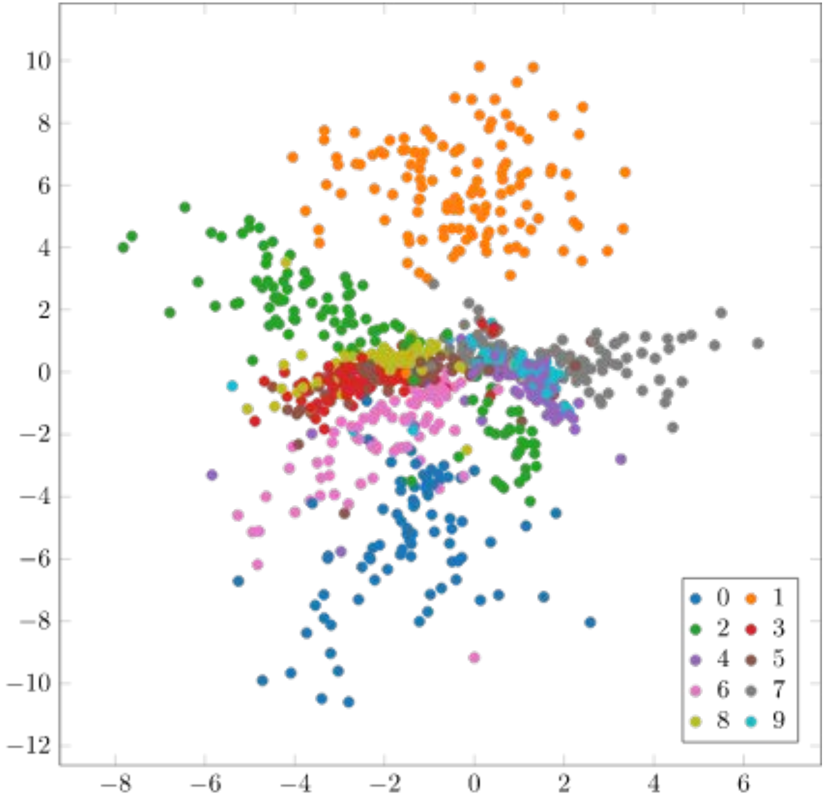


image: https://donghwa-kim.github.io/iforest.ht

Summary from today

Dimensionality reduction is universally helpful for science! It’s really what we do!

Principal component analysis is a linear, data-driven dimensionality reduction technique

The autoencoder is a dimensionality reduction technique which uses neural networks to find low-dimensional representations of our data