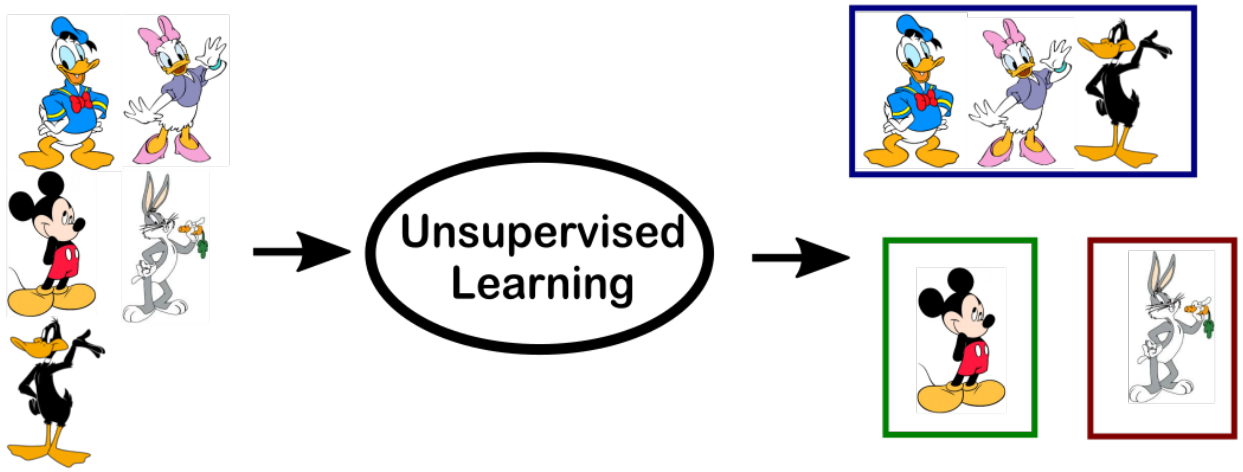
Unsupervised ML

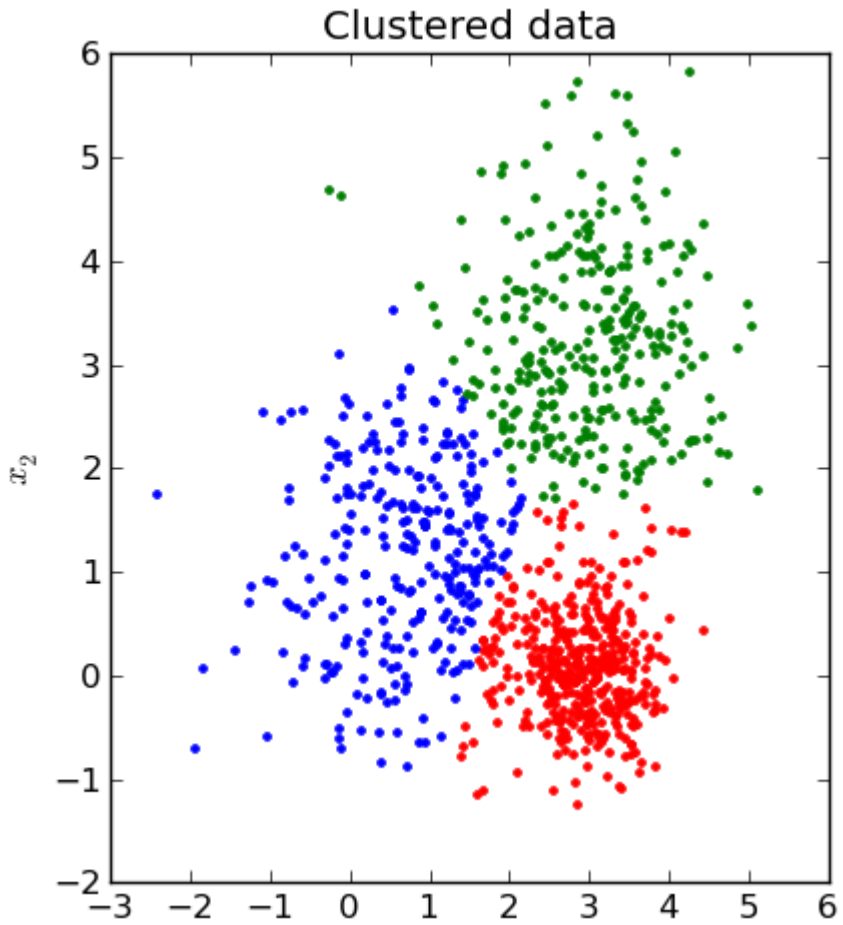
- **Idea: Find patterns & trends** in the data, without any prior knowledge
- These patterns may give us new insights into our data
- **Main Types:**
  - **Clustering**
  - **Dimensionality reduction**



NOSERYOUNG

Clustering

- Group datapoints into «close» groups
  - ➔Works on some measure of similarity / distance
- Applications:
  - Customer segmentation
  - Recomender Systems
  - Anomaly Detection



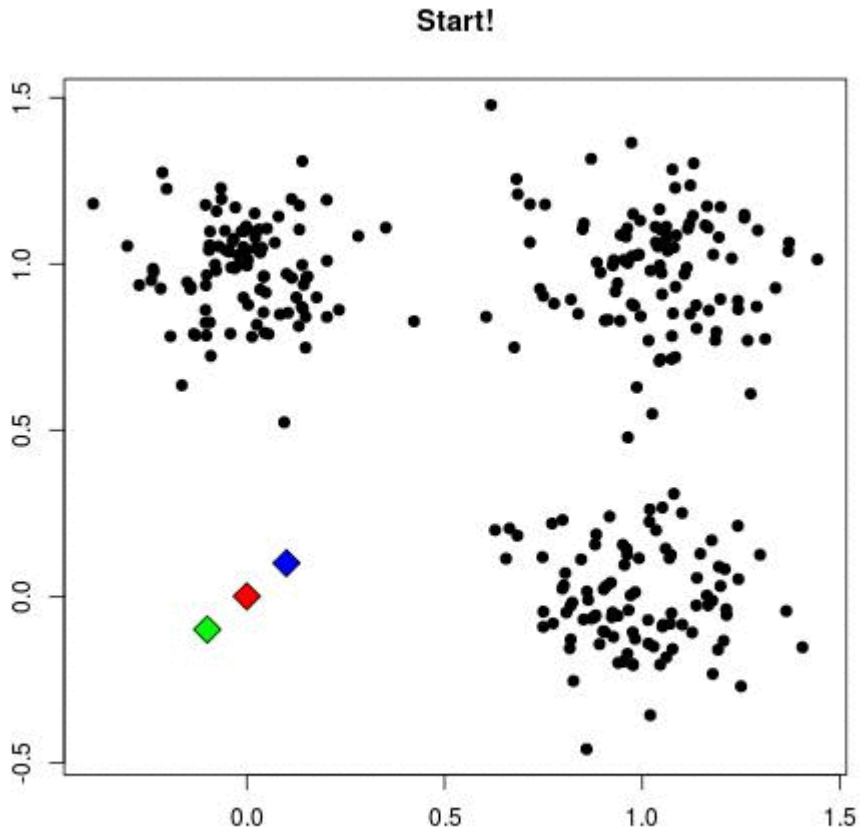
NOSERYOUNG

K-Means

Algorithm

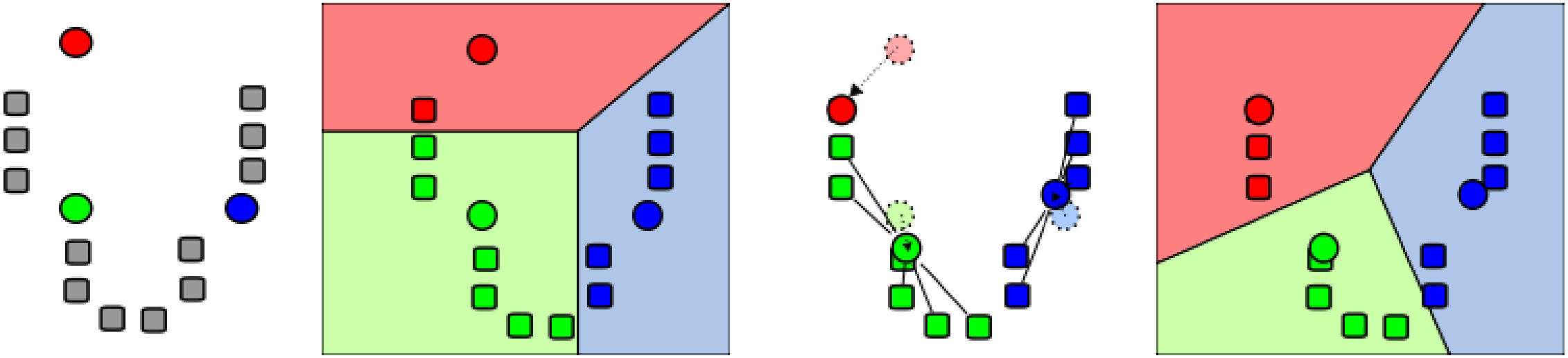
1. Initialize **k** random centroids
2. do{
   1. re-assign all points to closest centroid
   2. Recalculate centroid

   } while #reasignements > 0



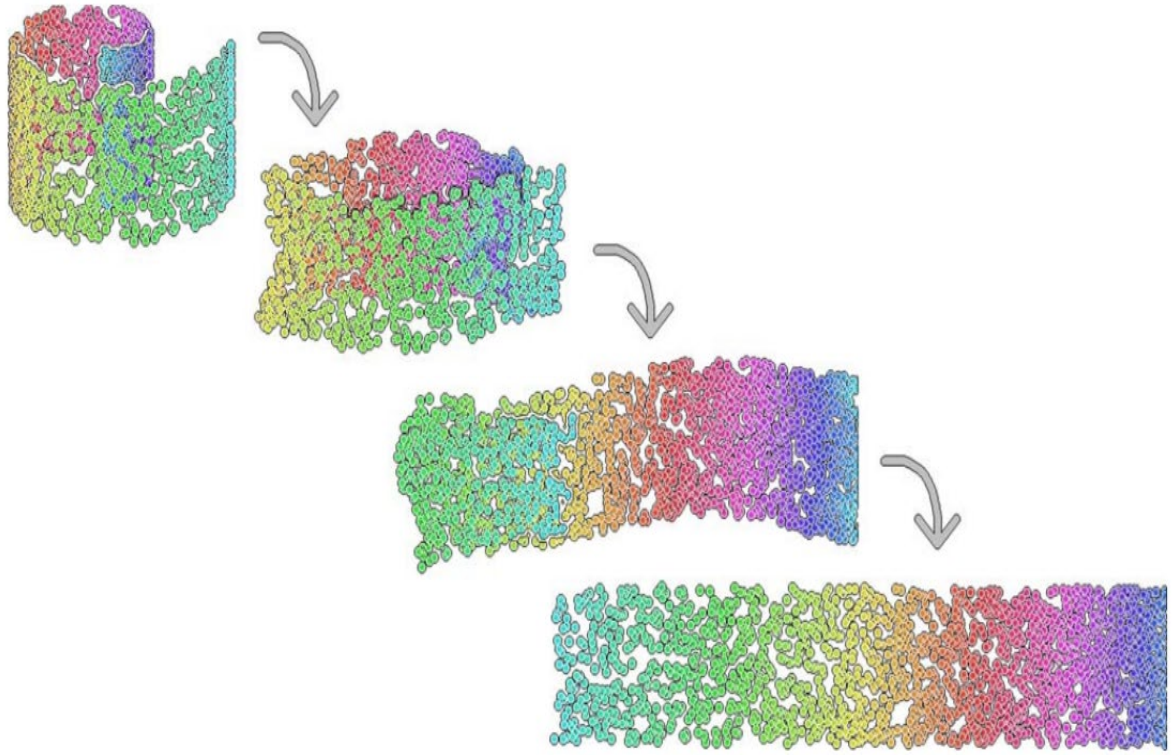
NOSERYOUNG

K-Means



Dimensionality Reduction

- **Idea**: represent a high-dimensional dataset in lower dimensions, while preserving local structures
- **Uses:**
  - **Data visualisation**
  - **Denoising**



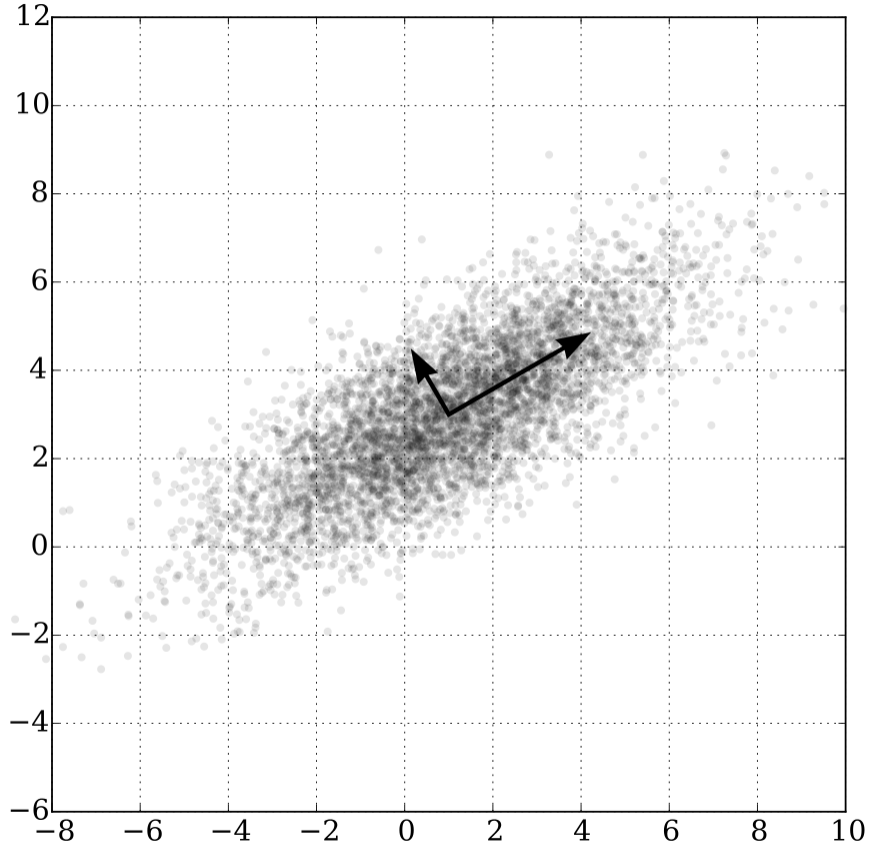
NOSERYOUNG

Principal Component Analysis

For dimensionality reduction

- **Idea**: Find the **principal components** that best describe the variations in the data
  - Intuitive explanation of PC: Main axis of variance

- Implemented in `sklearn.decomposition.PCA`



NOSERYOUNG

Additional Information
Unsupervised Learning

- kMeans: https://www.youtube.com/watch?v=mfqmoUN-Cuw
- PCA: https://www.youtube.com/watch?v=_UVHneBUBW0

NOSERYOUNG

Hands-On

Part 3

1. Implement the K-Means Algorithm for a set of random 2D datapoints (use the ‘`sklearn make_blobs`’ function to get a random dataset with underlying clusters
    1. Visualize your results (Bonus: can you animate the graph to show each iteration of the algorithm?)
2. Think about how you could use your implementation to categorize a new (previously unknown) datapoint.
    1. Bonus: Implement your idea and visualize the result

NOSERYOUNG