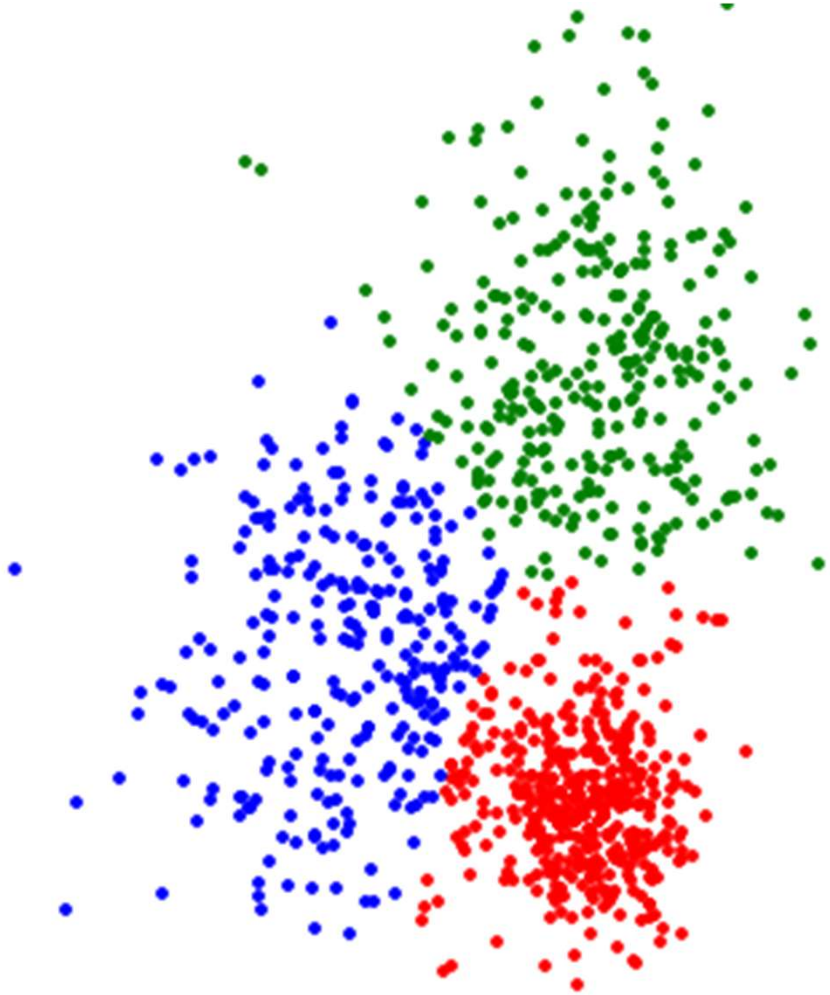
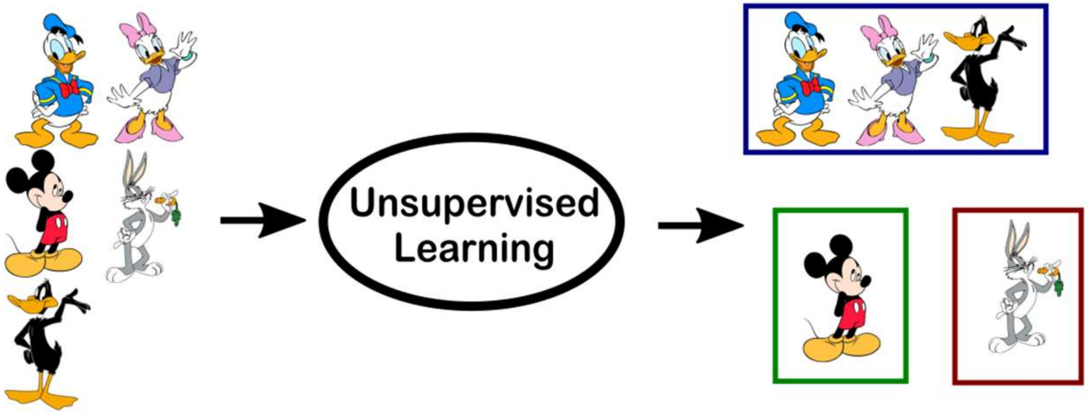


Unsupervised Machine Learning

NOSERYOUNG

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**



Clustering

- Group datapoints into «close» groups

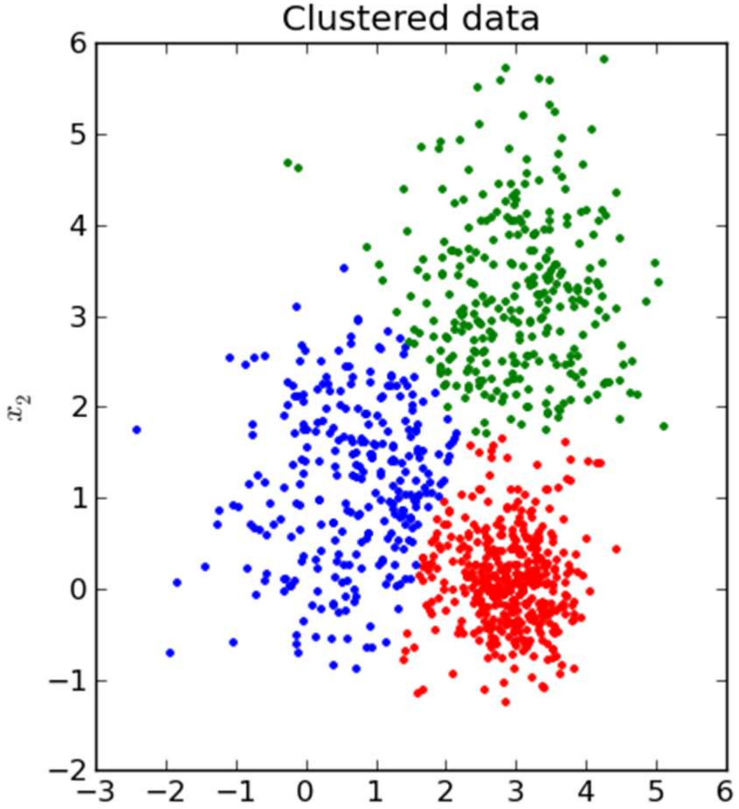
  ➔Works on some measure of similarity / distance

- **Applications**:
  - Customer segmentation

    «what are the main groups in my customer base?»
  - Recommender Systems

    «customers like you also bought…»
  - Anomaly Detection

    «this does not look like the others»

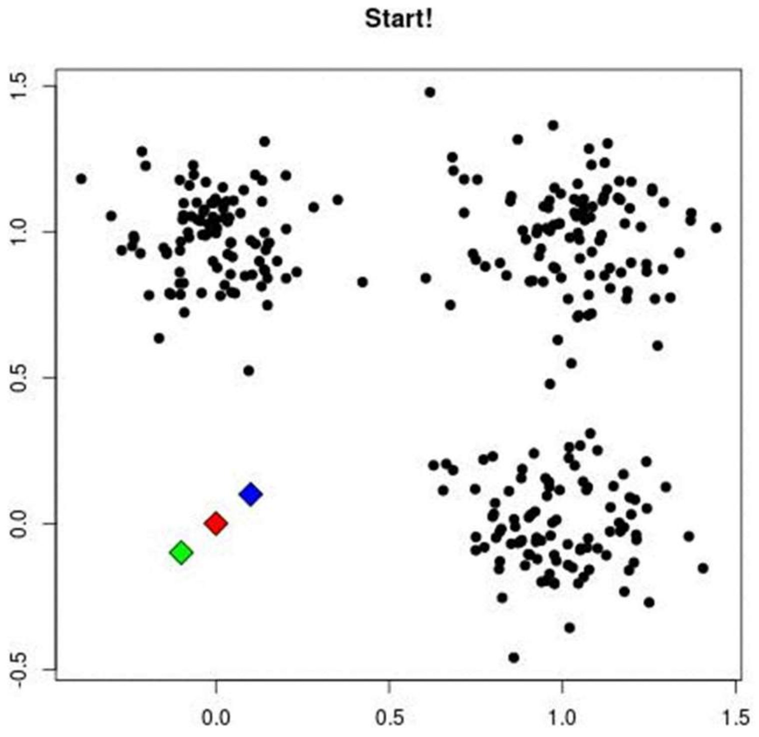


NOSERYOUNG

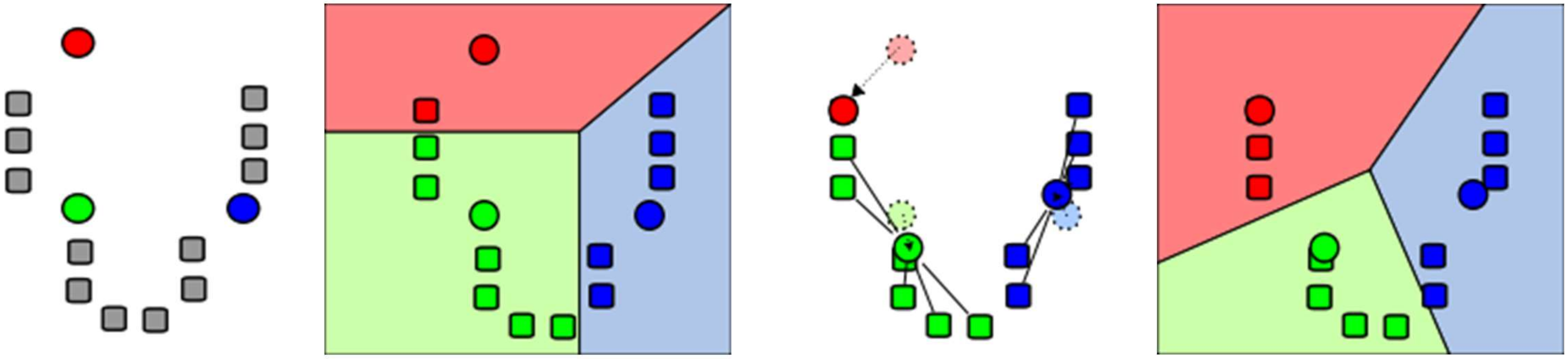
K-Means

Algorithm for Clustering

```
1. Initialize k random
   cluster-centers
2. do{
     1. Re-assign all points to
        closest cluster-center
     2. Recalculate cluster-
        centers
    } while #reasignements > 0
```



K-Means



Dimensionality Reduction

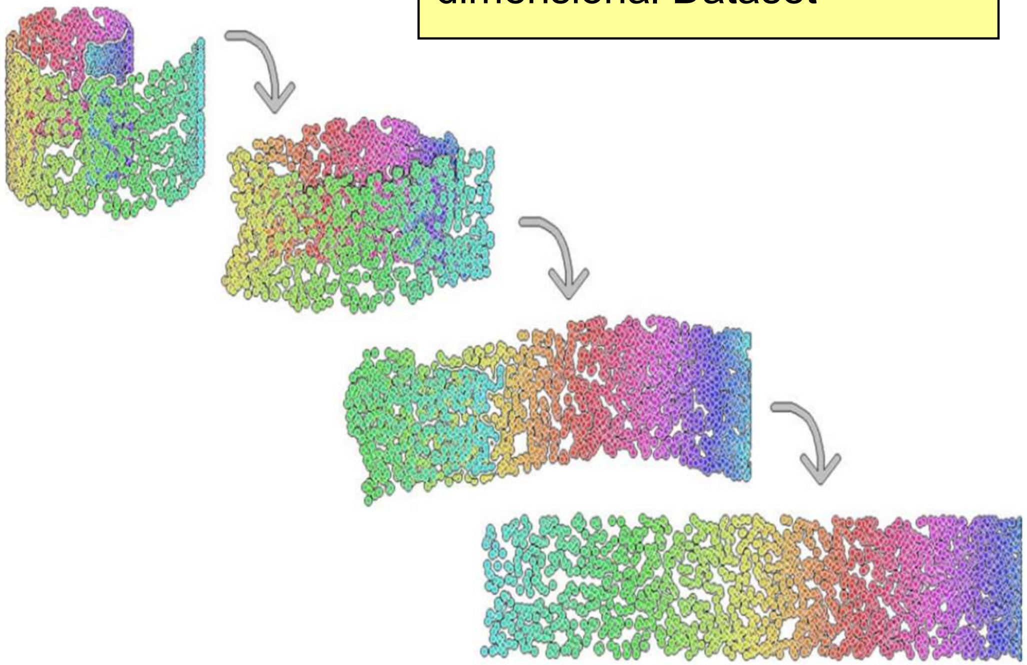
Dimension in Data-Science / Mathematics:

One axis or column of a dataset. E.g. a Dataset with 10 Columns is a 10 dimensional Dataset

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

    «how do I visualize a 10-D dataset?!»
  - **Denoising**

    «real world variance vs. measurment-error»



Principal Component Analysis

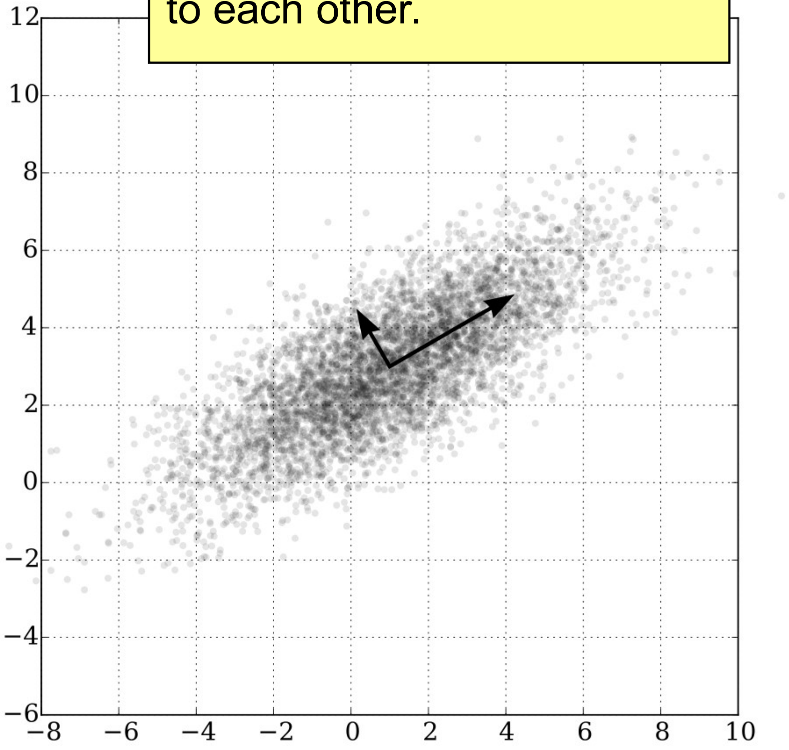
For dimensionality reduction

- **Idea**: Find the **principal components** that best describe the variations in the data
  - “Intuitive” explanations of PCA:
    - Shift the coordinate system such that you can discard one or more axis without loosing much information

      ➔ PC is the main axis of variance
    - Combine multiple columns of the dataset into one in the optimal way
- Implemented in `sklearn.decomposition.PCA`

Principal Component:

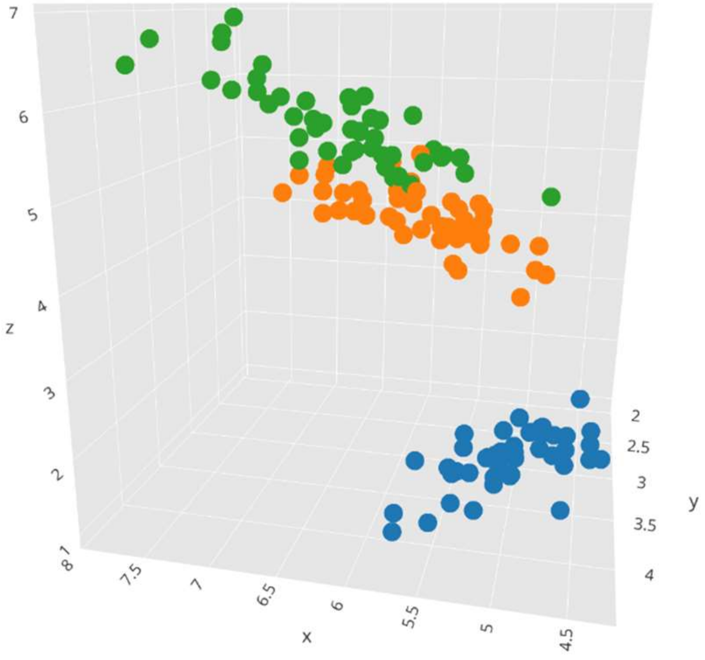
Main Axis of Variance.
Always perpendicular (90°) to each other.



Additional Information Unsupervised Learning

- **kMeans**: https://www.youtube.com/watch?v=mfqmoUN-Cuw
- **PCA**: https://www.youtube.com/watch?v=_UVHneBUBW0
  - Math: https://www.youtube.com/watch?v=PFDu9oVAE-g

curse of dimensionality



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
   - Visualize your results (Bonus: can you animate the graph to show each iteration of the algorithm?)
   - How could you improve the initialization-step to reduce strange results?
2. Think about how you could use your implementation to categorize a new (previously unknown) datapoint.
   - Bonus: Implement your idea and visualize the result