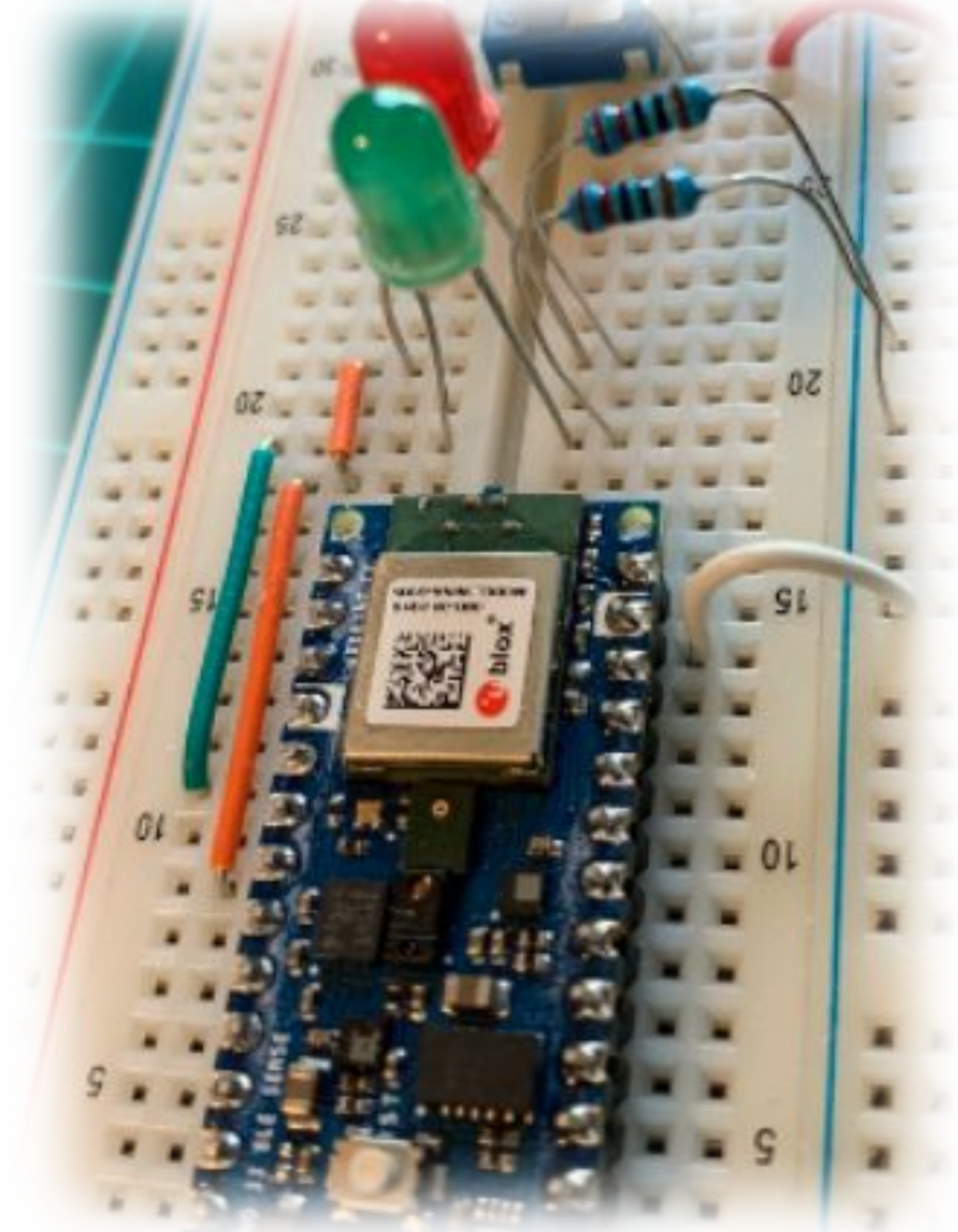
IESTI01 – TinyML

Embedded Machine Learning

21. K-means Clustering & Anomaly Detection

Prof. Marcelo Rovai
UNIFEI



K-means Clustering

Machine Learning can be…

Supervised learning

Task-driven

- Regression
- Classification
- Object detection

Unsupervised learning

Data-driven

- Clustering
- Segmentation
- **Anomaly detection**

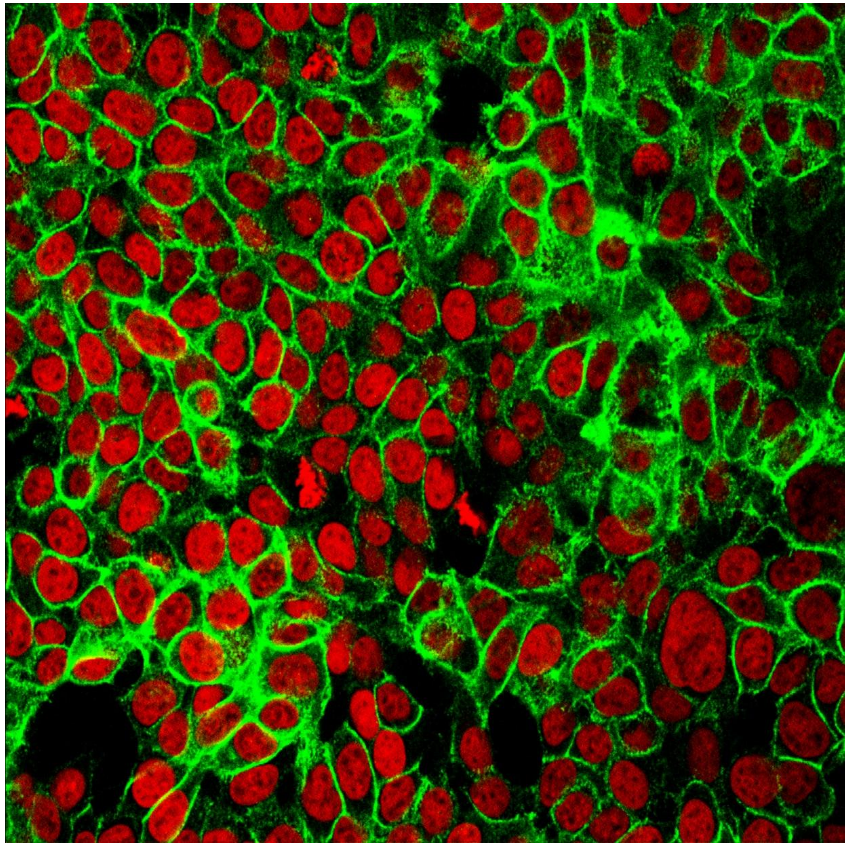
Reinforcement learning

Learn from experience

- Robotics
- Games
- Recommender systems

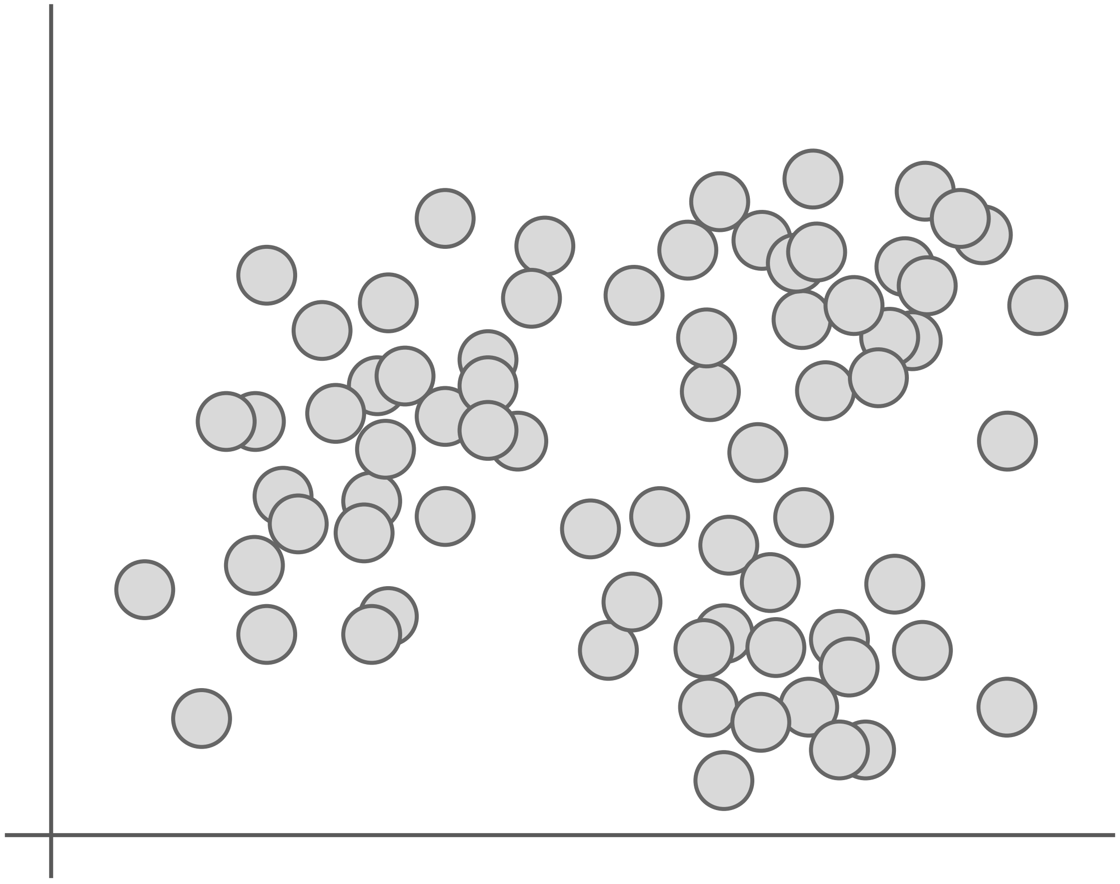
Unsupervised Learning

- **No labels**!
- Model automatically discovers patterns in the data
- Uses
  - Segmentation
  - Clustering
  - Dimensionality reduction
  - Anomaly detection



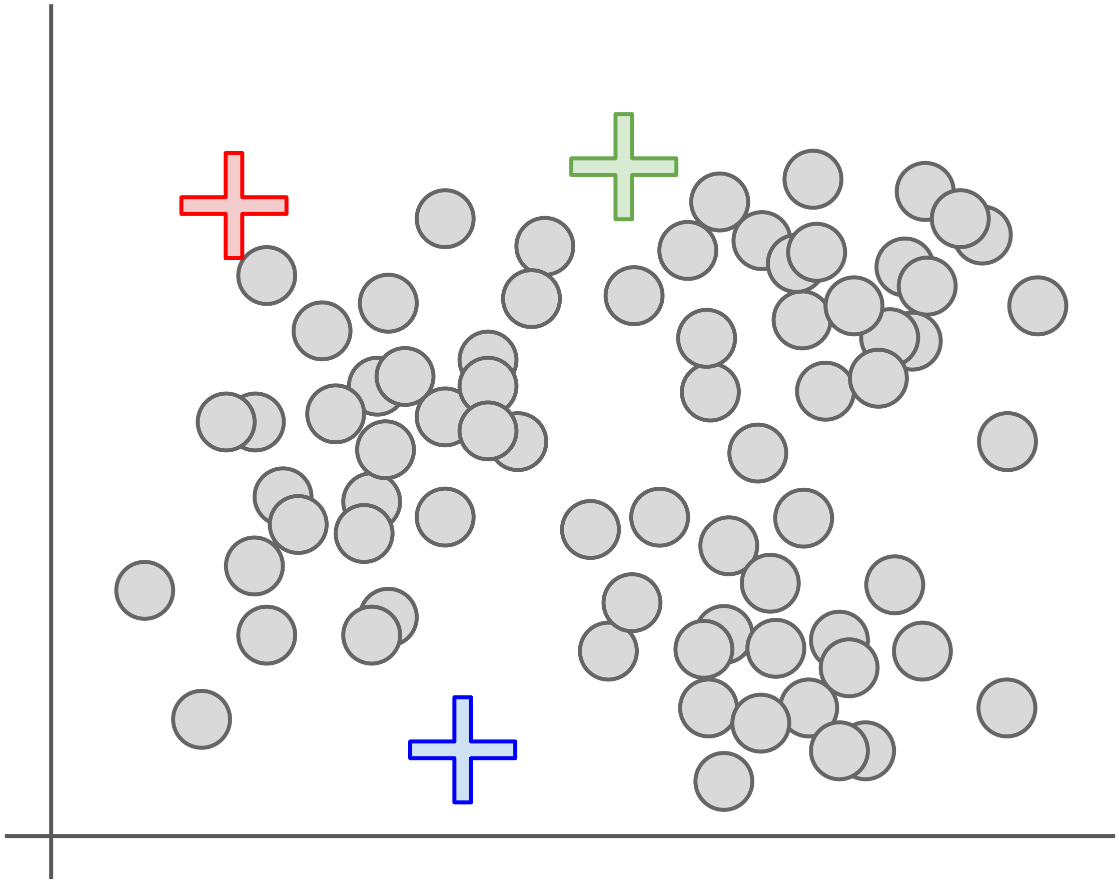
K-means Clustering

1. Define k (e.g. k=3)



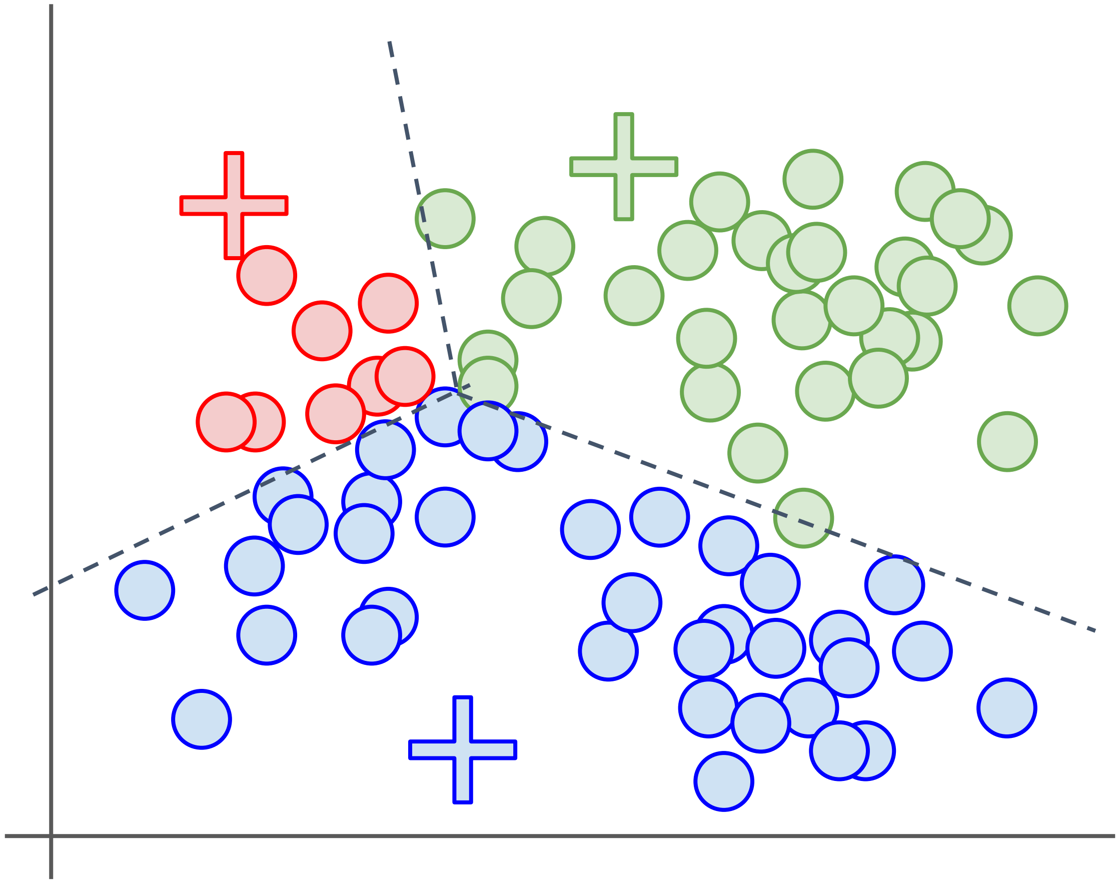
K-means Clustering

1. Define k (e.g. k=3)
2. Randomly choose centroid for each cluster



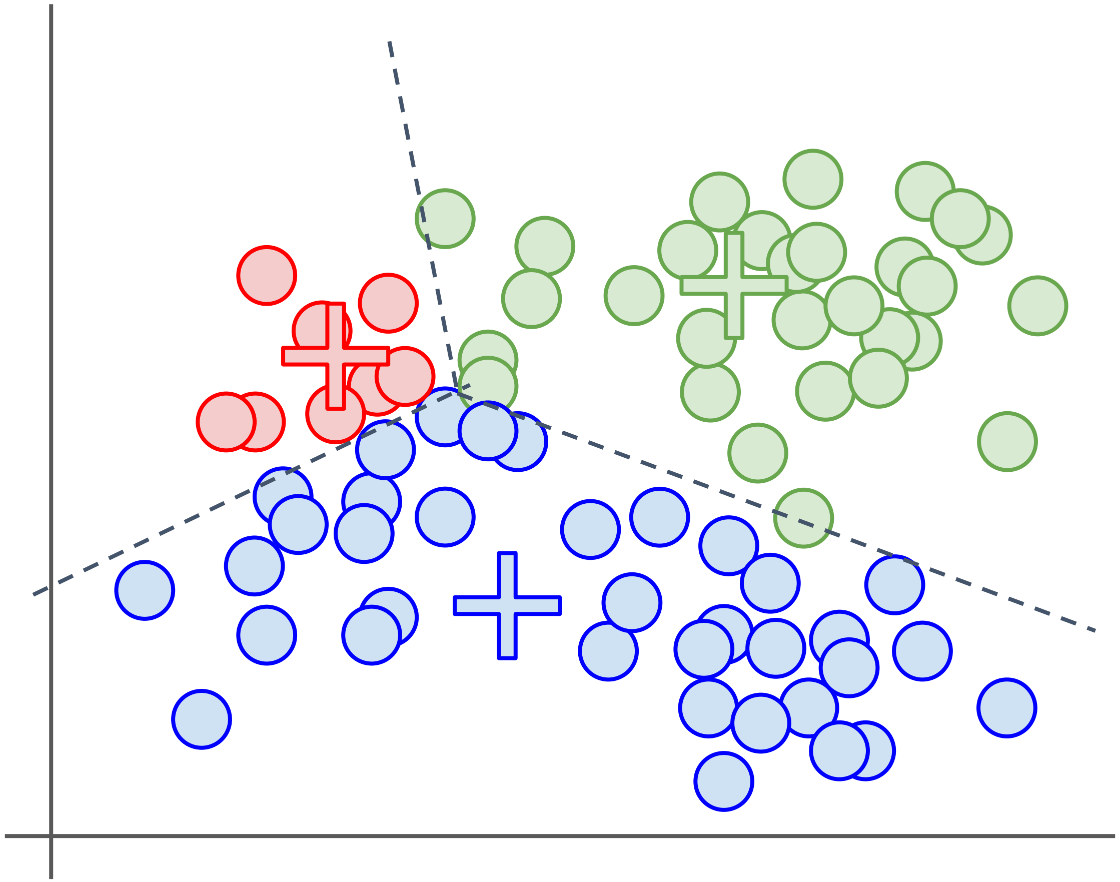
K-means Clustering

1. Define k (e.g. k=3)
2. Randomly choose centroid for each cluster
3. Assign every sample to nearest centroid based on Euclidean distance



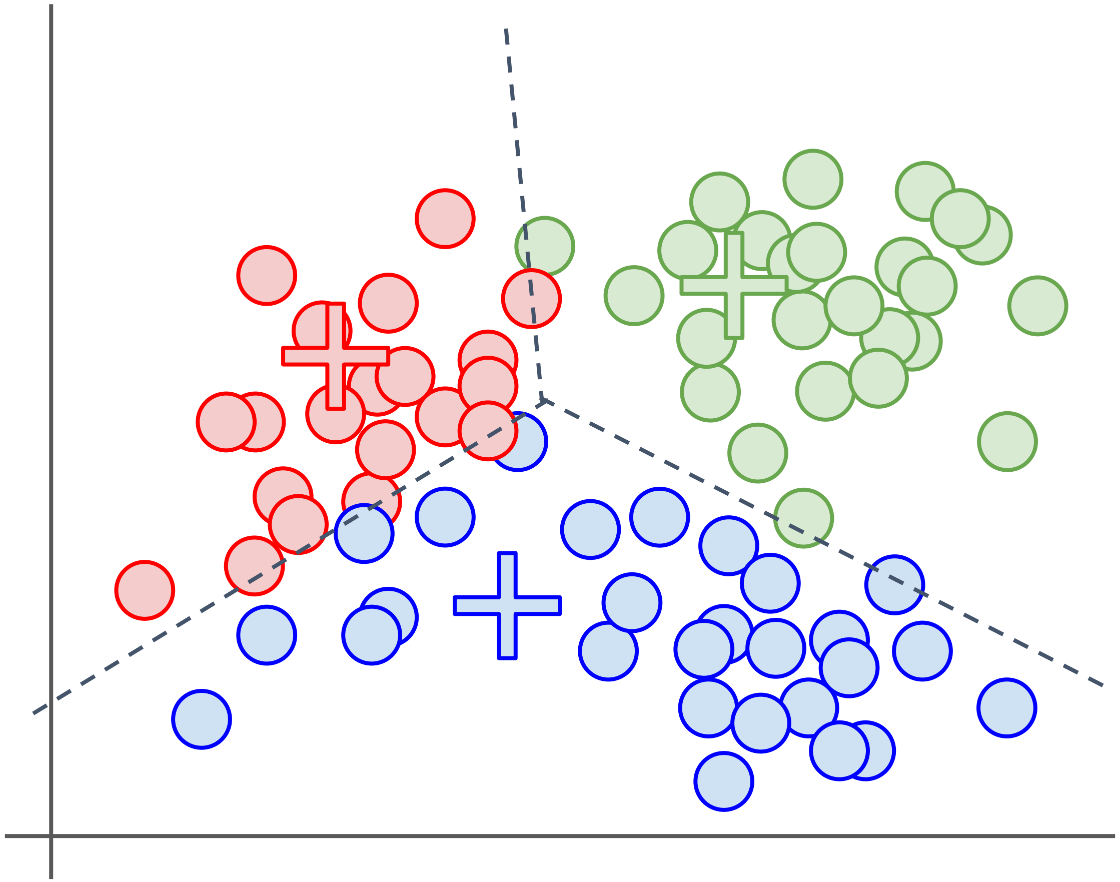
K-means Clustering

1. Define k (e.g. k=3)
2. Randomly choose centroid for each cluster
3. Assign every sample to nearest centroid based on Euclidean distance
4. Re-compute the centroid of the cluster



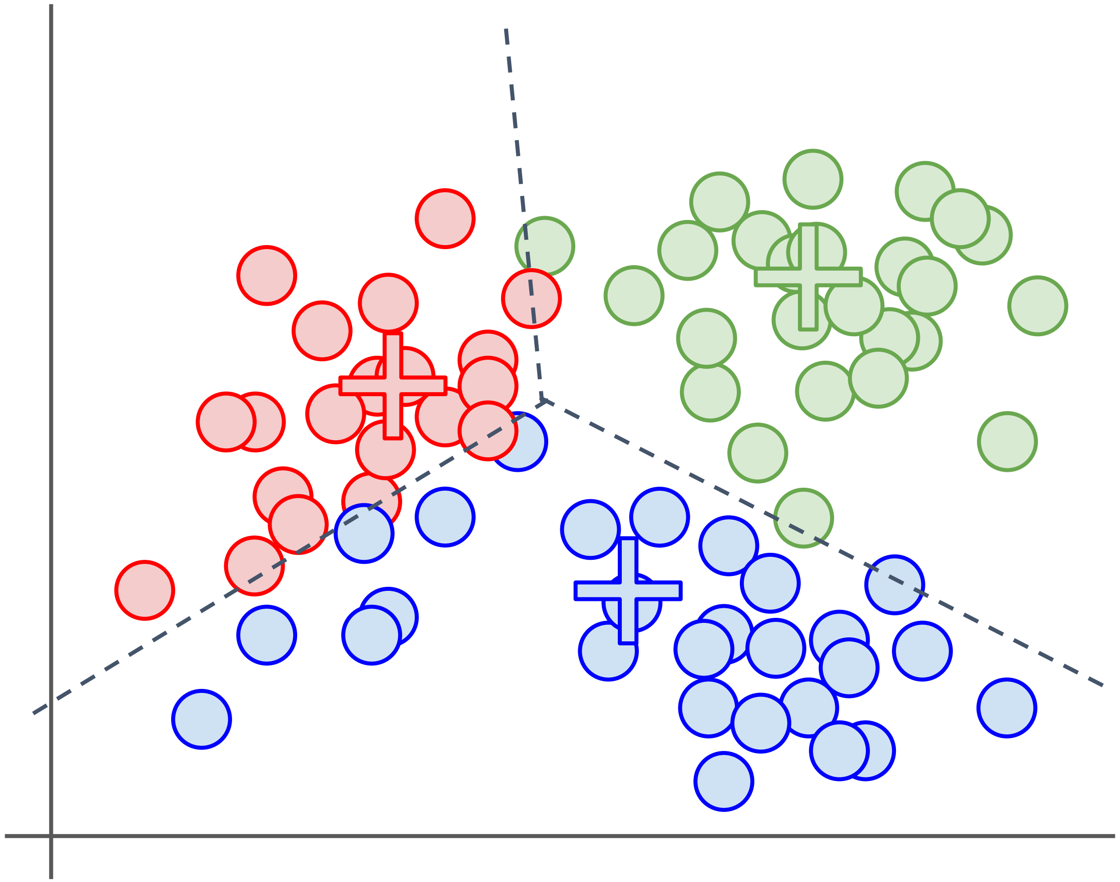
K-means Clustering

1. Define k (e.g. k=3)
2. Randomly choose centroid for each cluster
3. Assign every sample to nearest centroid based on Euclidean distance
4. Re-compute the centroid of the cluster
5. Repeat steps 3-4



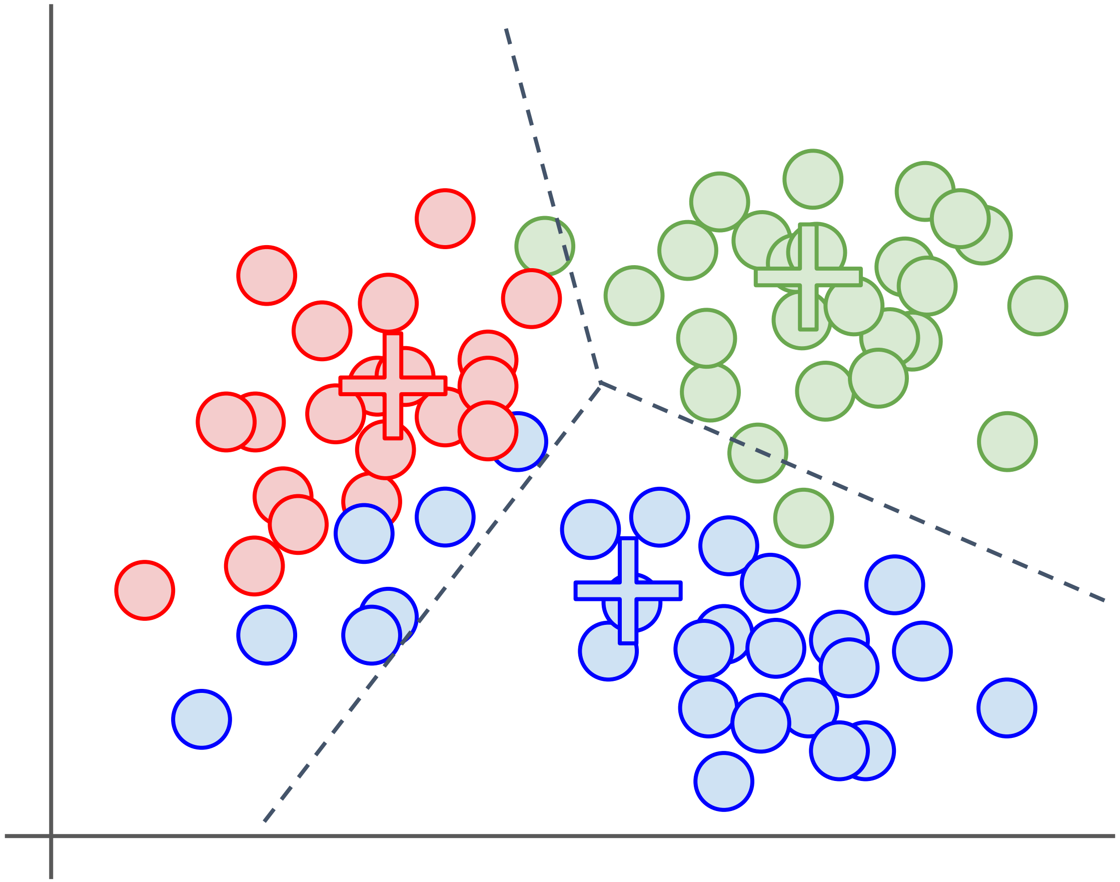
K-means Clustering

1. Define k (e.g. k=3)
2. Randomly choose centroid for each cluster
3. Assign every sample to nearest centroid based on Euclidean distance
4. Re-compute the centroid of the cluster
5. Repeat steps 3-4



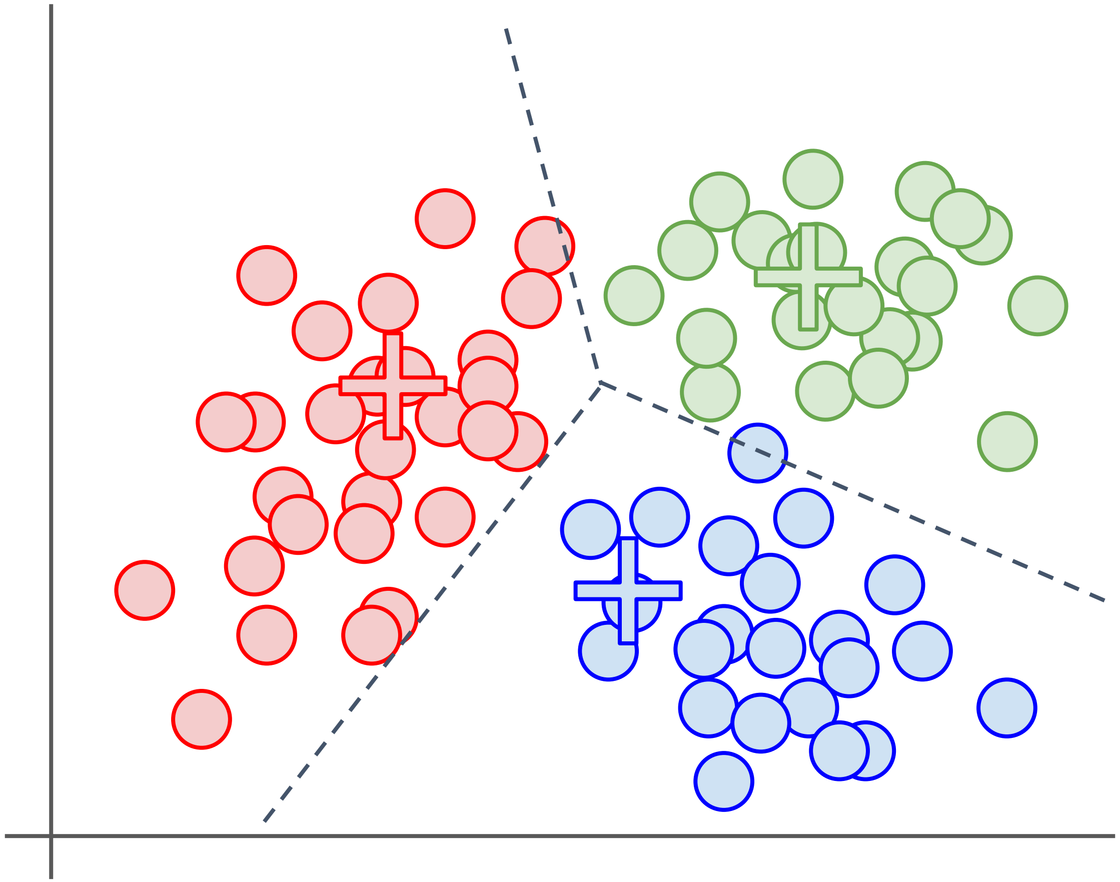
K-means Clustering

1. Define k (e.g. k=3)
2. Randomly choose centroid for each cluster
3. Assign every sample to nearest centroid based on Euclidean distance
4. Re-compute the centroid of the cluster
5. Repeat steps 3-4



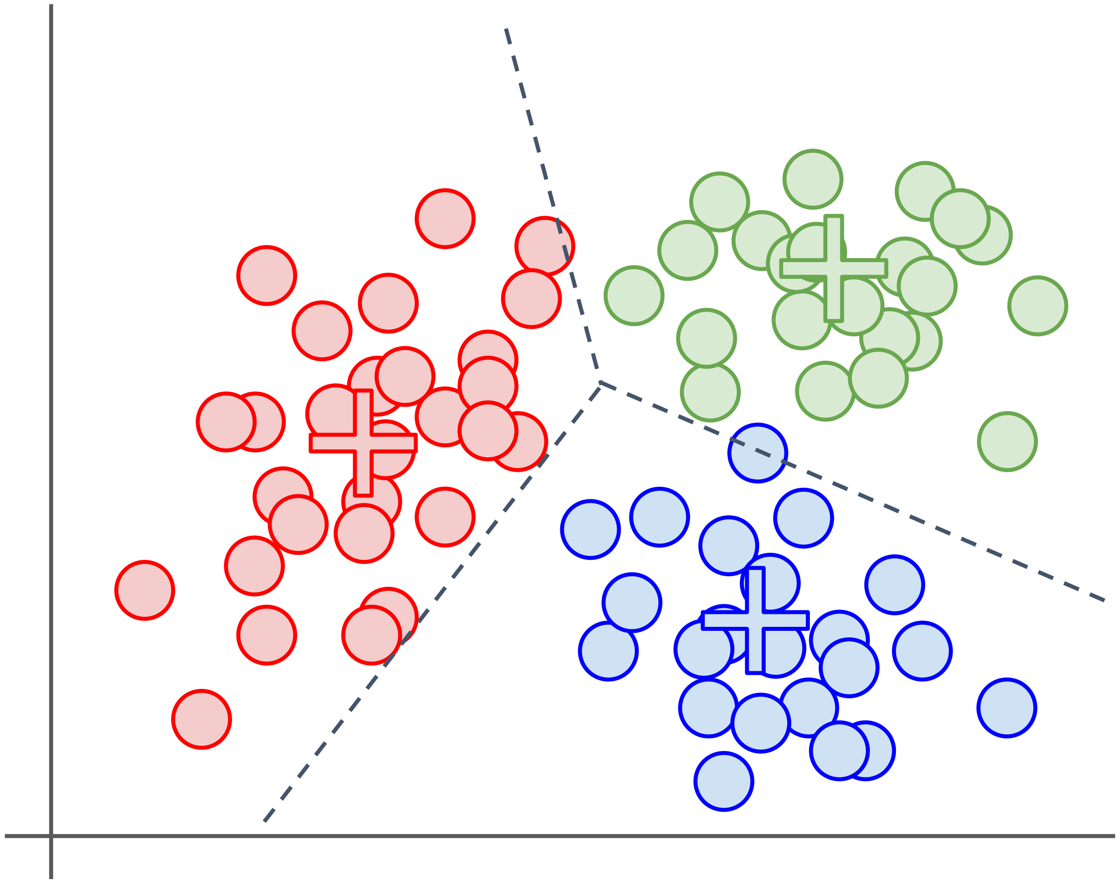
K-means Clustering

1. Define k (e.g. k=3)
2. Randomly choose centroid for each cluster
3. Assign every sample to nearest centroid based on Euclidean distance
4. Re-compute the centroid of the cluster
5. Repeat steps 3-4



K-means Clustering

1. Define k (e.g. k=3)
2. Randomly choose centroid for each cluster
3. Assign every sample to nearest centroid based on Euclidean distance
4. Re-compute the centroid of the cluster
5. Repeat steps 3-4



K-means Clustering

1. Define k (e.g. k=3)
2. Randomly choose centroid for each cluster
3. Assign every sample to nearest centroid based on Euclidean distance
4. Re-compute the centroid of the cluster
5. Repeat steps 3-4
6. …until one of:
   a. Sum of distances between data points and corresponding centroid is minimized
   b. No change in centroids
   c. Maximum iterations reached

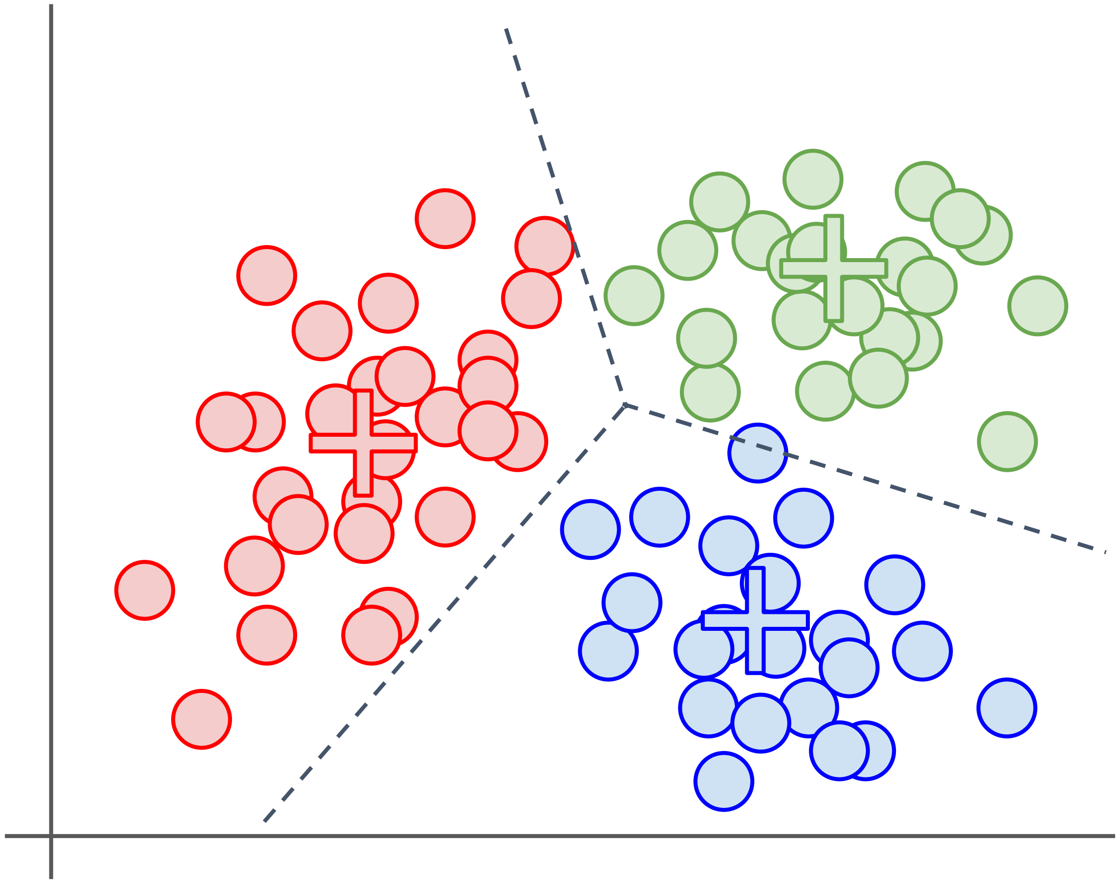
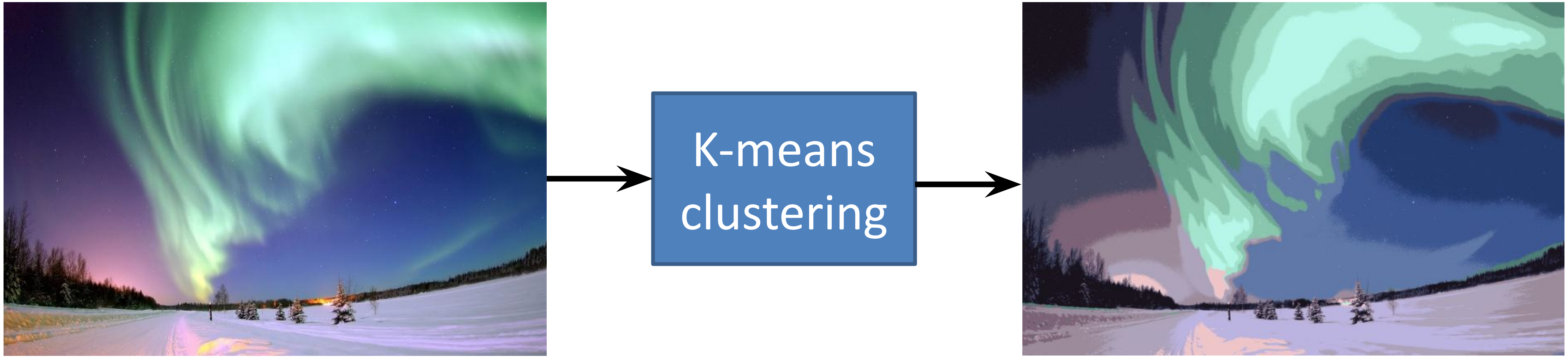
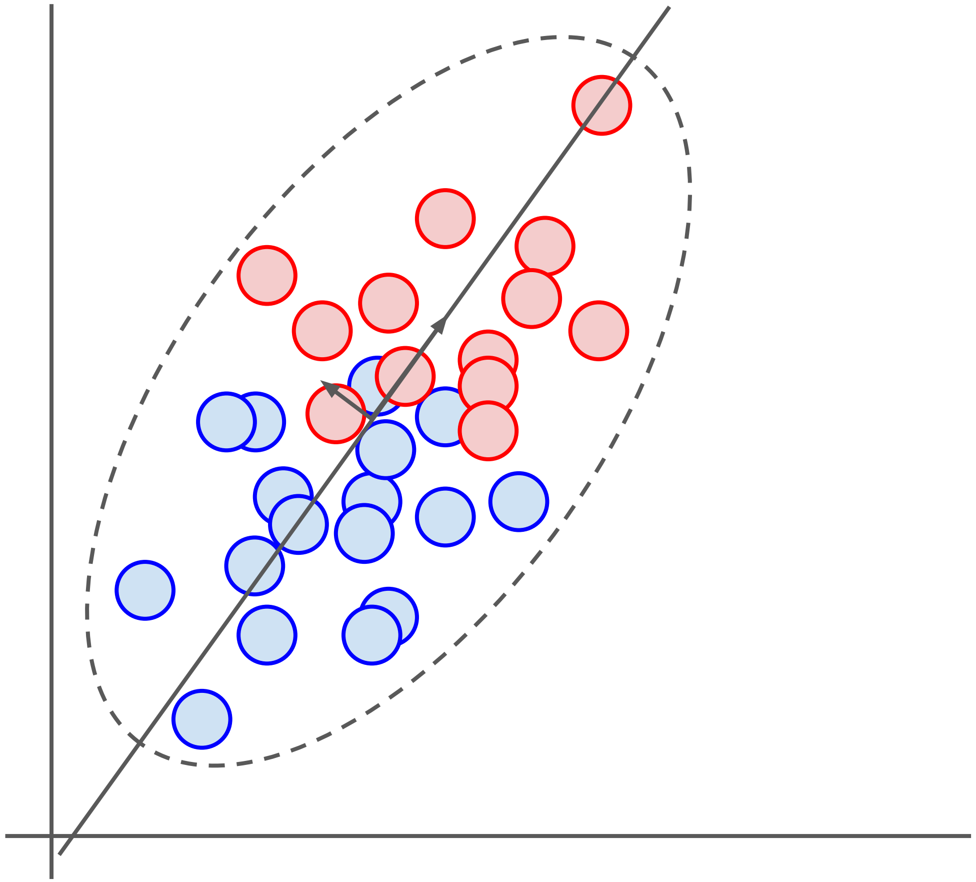


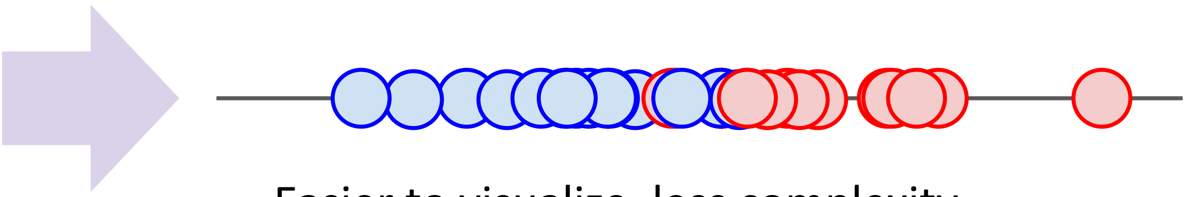
Image Segmentation



Dimensionality Reduction

Example: principal component analysis (PCA)



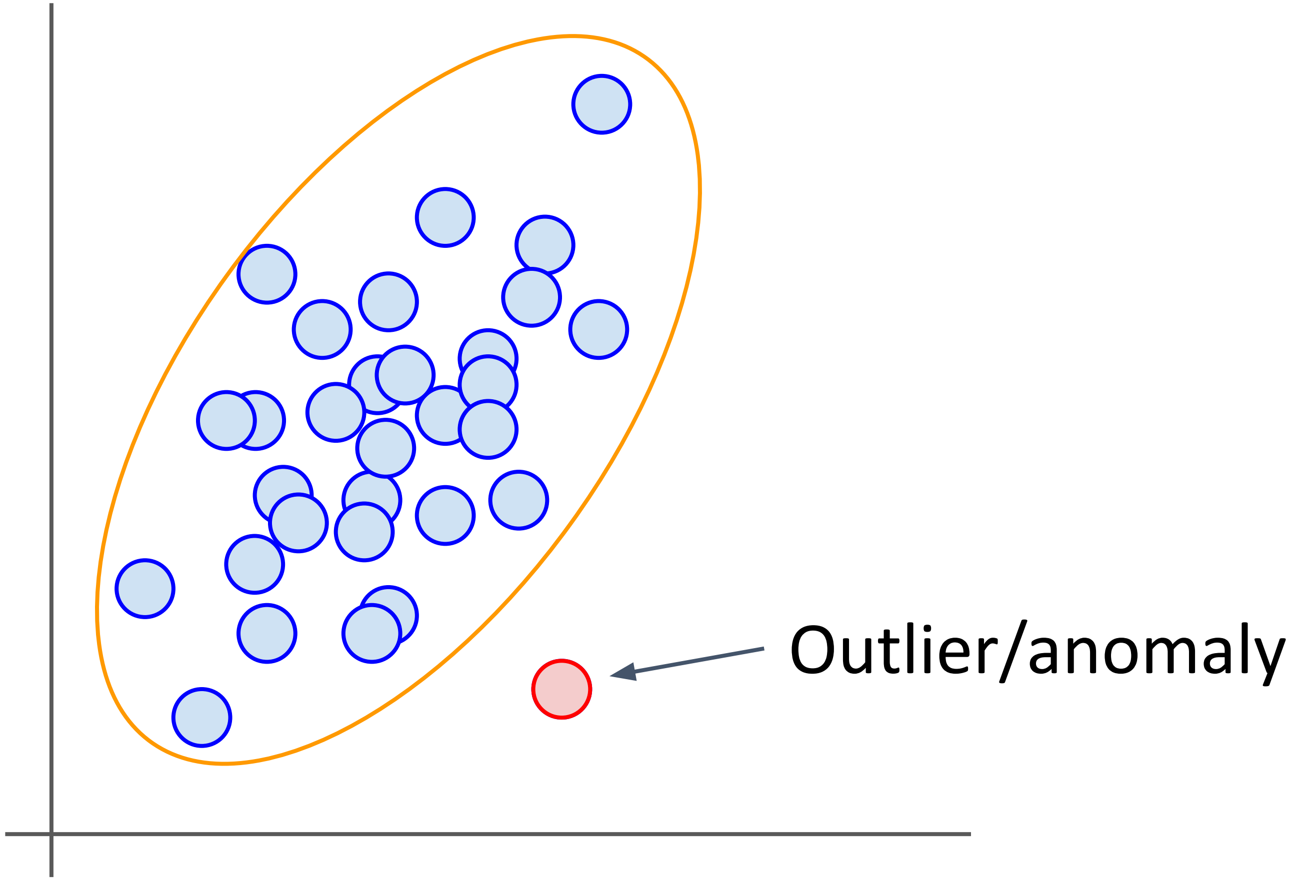


Easier to visualize, less complexity

Anomaly Detection

Examples:

- Email spam
- Credit card fraud
- Motion alarm
- Fault detection



K-means Clustering for Anomaly Detection

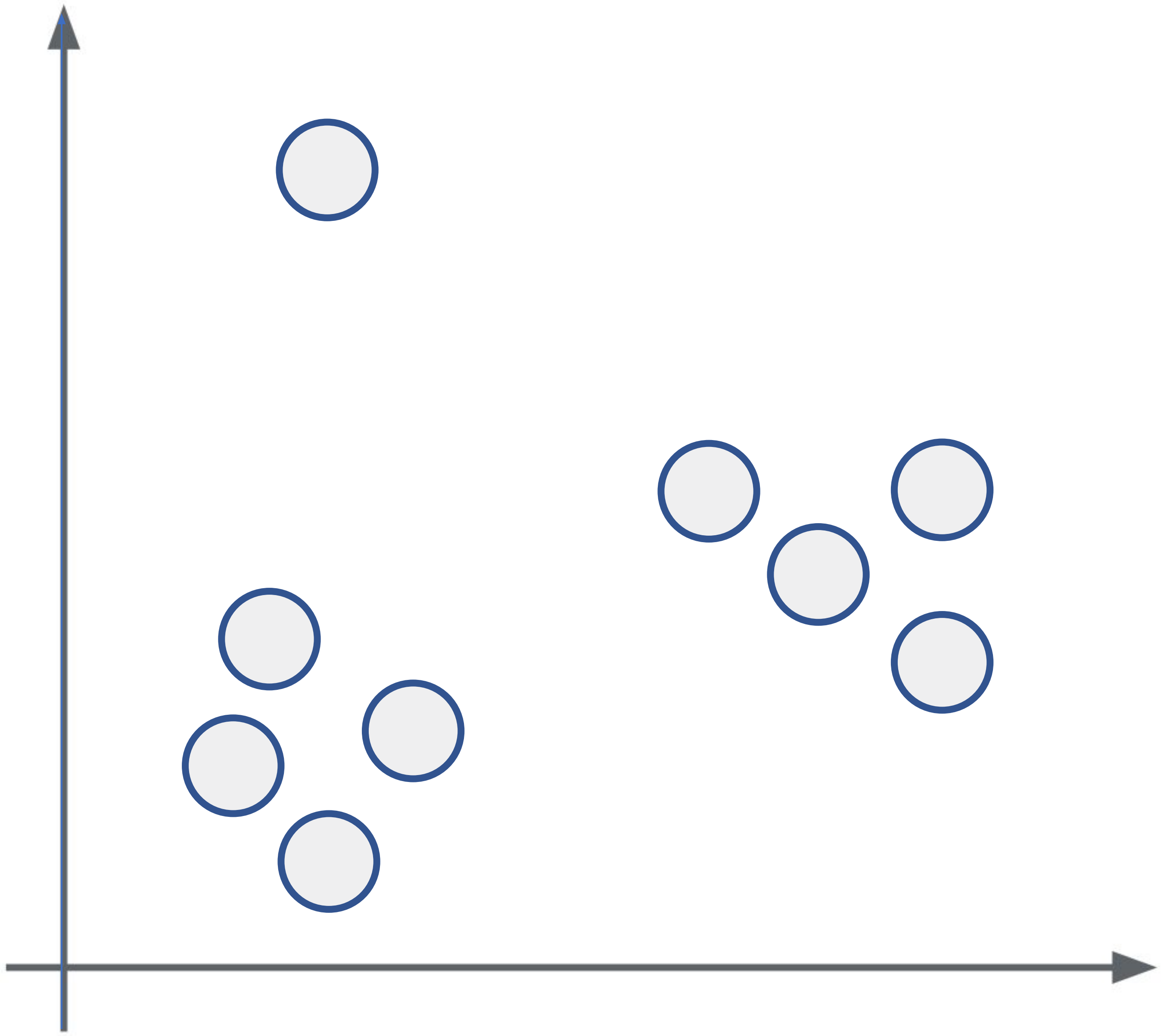
Code Time!

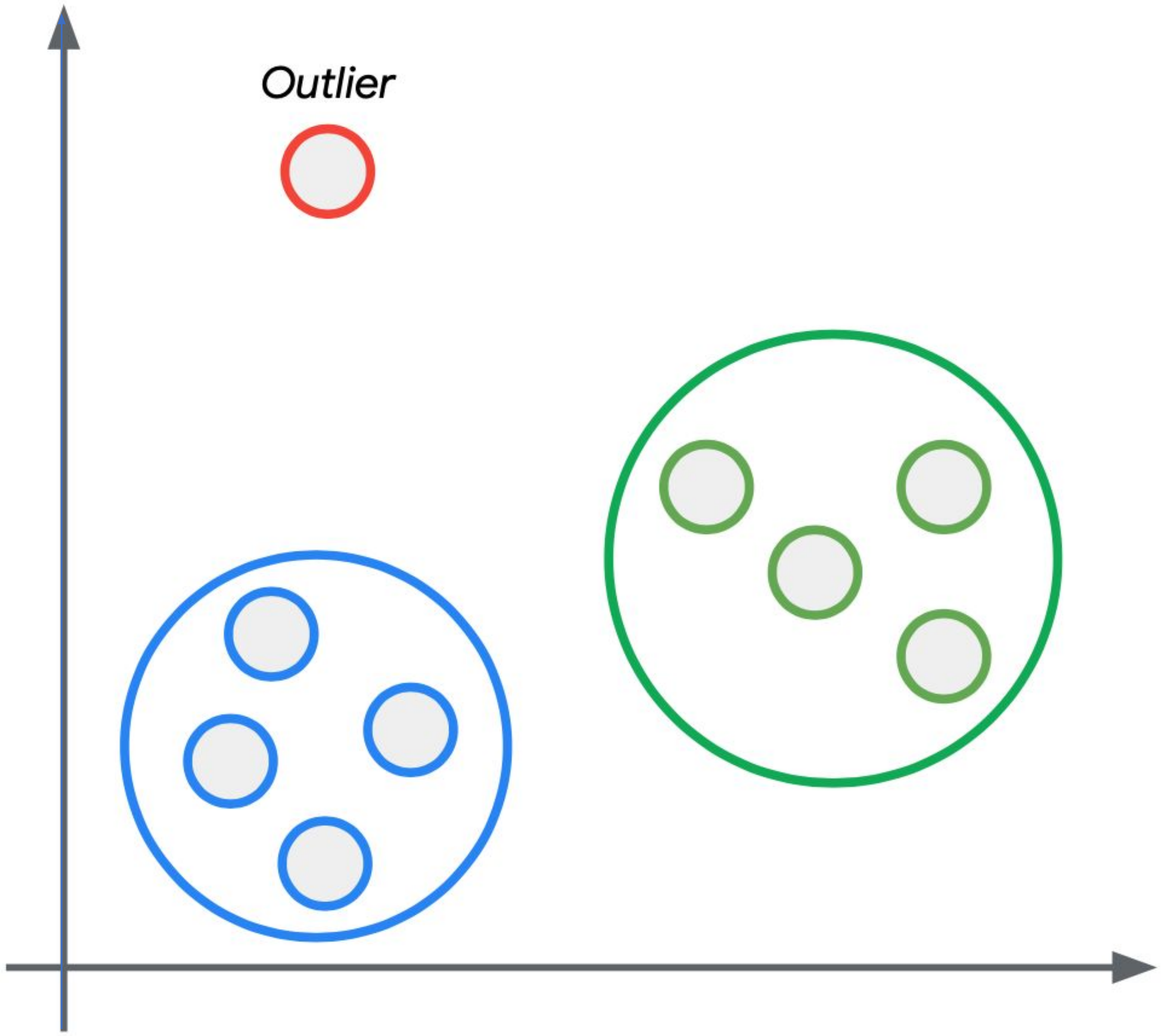
Anomaly_Detection_K_means.ipynb

Anomaly Detection

What is **Anomaly Detection**?

In **data analysis, anomaly detection** is the **identification of rare** items, events or observations which **raise suspicions** because they **differing significantly** from the **majority of the data**.




Outlier

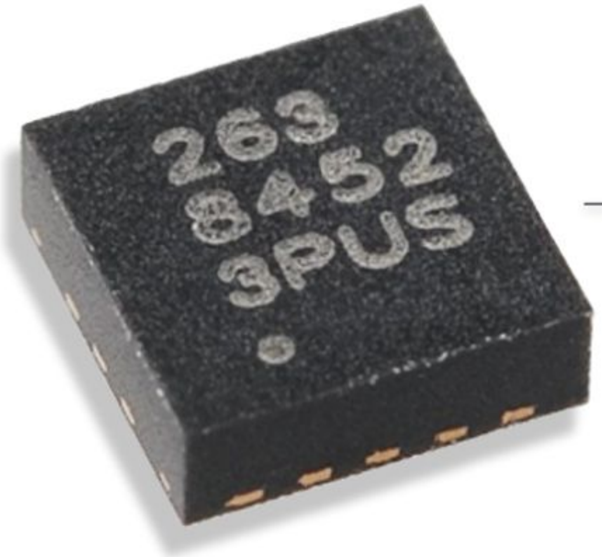
Application: Factory machinery

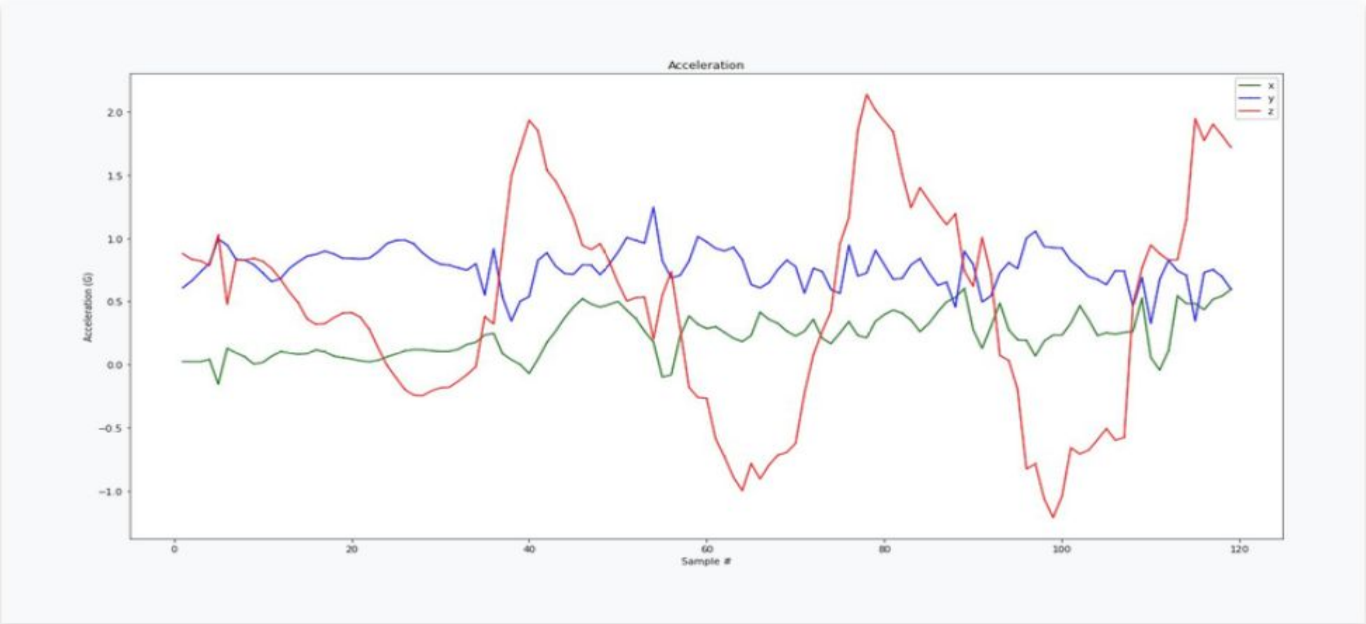
Application: Factory machinery

Ball Bearings

Accelerometer

Sensor: **Accelerometer**



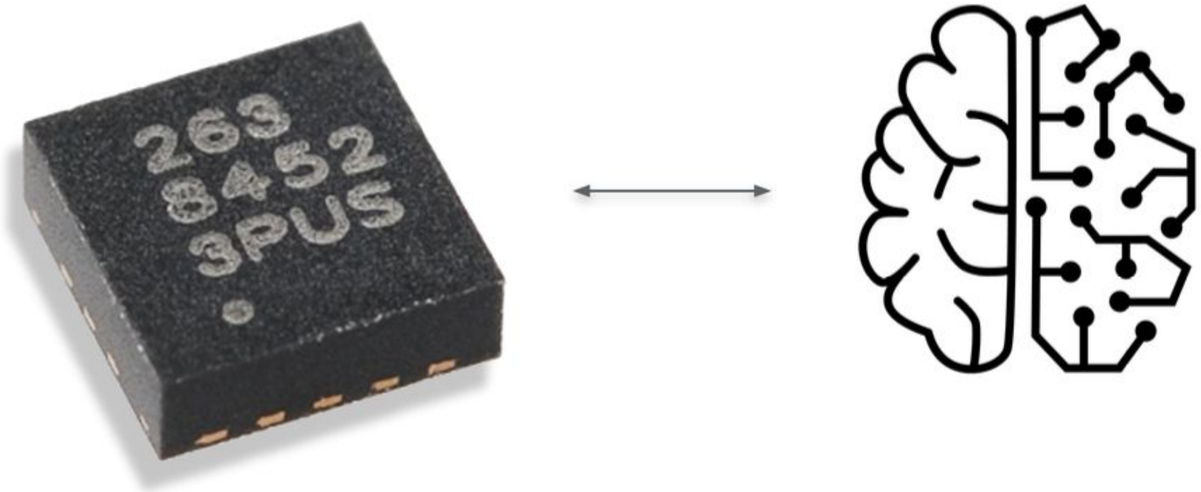


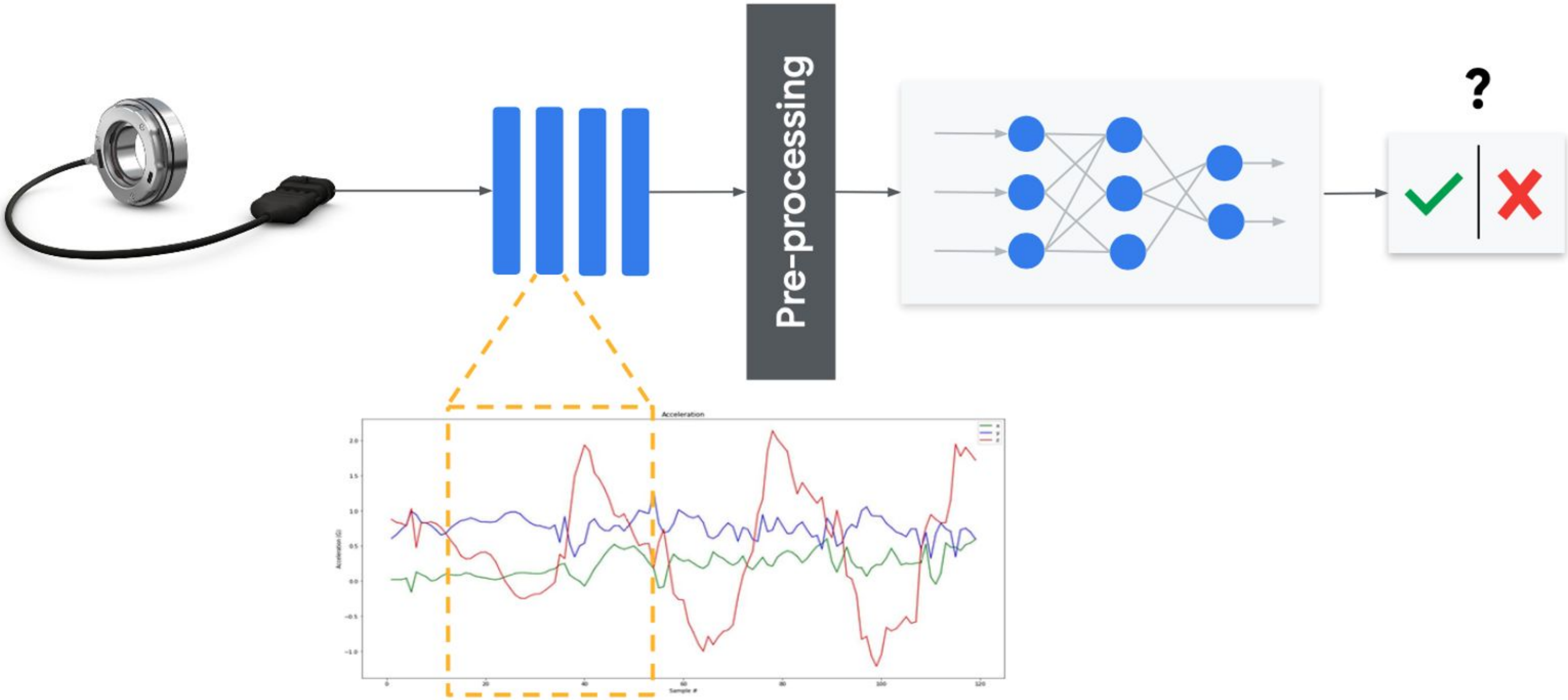
Sensor: **Accelerometer**

$$\underbrace{2\text{ bytes}}_{\text{Measurement}} \times \underbrace{8}_{\#\text{ Sensors}} \times \underbrace{20\text{kHz}}_{\text{Sample Rate}} = \mathbf{320}\text{ KB / sec}$$

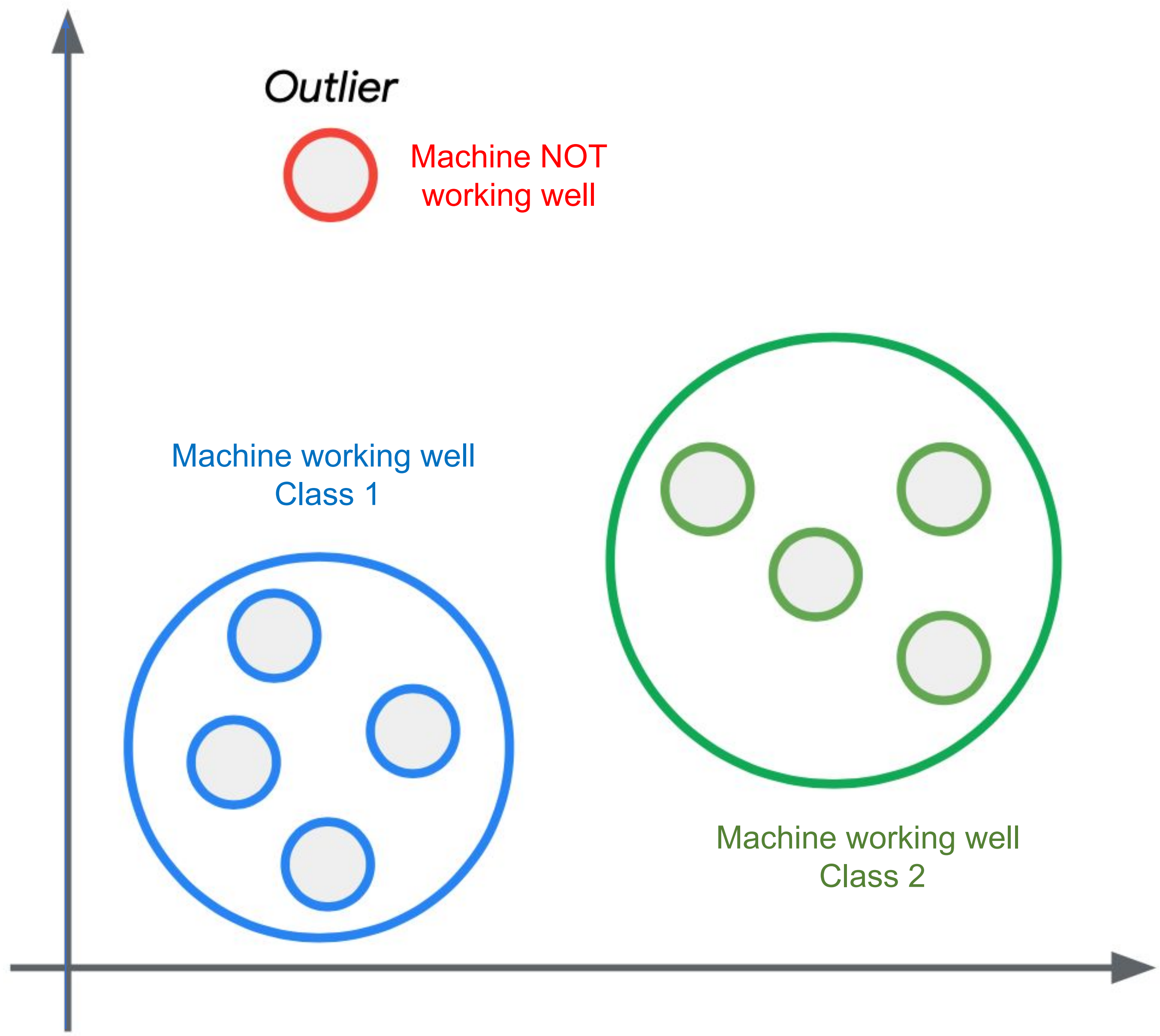
It’s too expensive to stream to the cloud

Need **"intelligence"** *close* **to sensors**





[IESTI01 2021.2 - Final Group Project: Bearing Failure Detection]


Outlier
Machine NOT working well
Machine working well Class 1
Machine working well Class 2

Reading Material

Main references

- Harvard School of Engineering and Applied Sciences - CS249r: Tiny Machine Learning
- Professional Certificate in Tiny Machine Learning (TinyML) – edX/Harvard
- Introduction to Embedded Machine Learning - Coursera/Edge Impulse
- Computer Vision with Embedded Machine Learning - Coursera/Edge Impulse
- Fundamentals textbook: "Deep Learning with Python" by François Chollet
- Applications & Deploy textbook: "TinyML" by Pete Warden, Daniel Situnayake
- Deploy textbook "TinyML Cookbook" by Gian Marco Iodice

I want to thank Shawn Hymel and Edge Impulse, Pete Warden and Laurence Moroney from Google, Professor Vijay Janapa Reddi and Brian Plancher from Harvard, and the rest of the TinyMLedu team for preparing the excellent material on TinyML that is the basis of this course at UNIFEI.

The IESTI01 course is part of the **TinyML4D**, an initiative to make TinyML education available to everyone globally.

Thanks

UNIFEI