



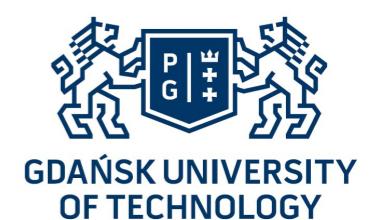
# Unsupervised Learning with Scikit-Learn

---

Jakub Powierza



Konferencja Inżynierii Oprogramowania



GDAŃSK UNIVERSITY  
OF TECHNOLOGY

# Jakub Powierza

---

- **Student** of Gdańsk University of Technology
- Co-founder & Board Member at **Gradient**
- ML & DL Enthusiast
- Fan of **Computer Vision**
- **MedTagger's** Team Leader
- **Graphics Software Engineer** at Intel Technology Poland



# Agenda

---

1. Introduction to Machine Learning
2. Clustering (~30 minutes)
3. Hierarchical Clustering (~30 minutes)
4. Anomaly Detection (~40 minutes)

Ask Questions anytime! 😊

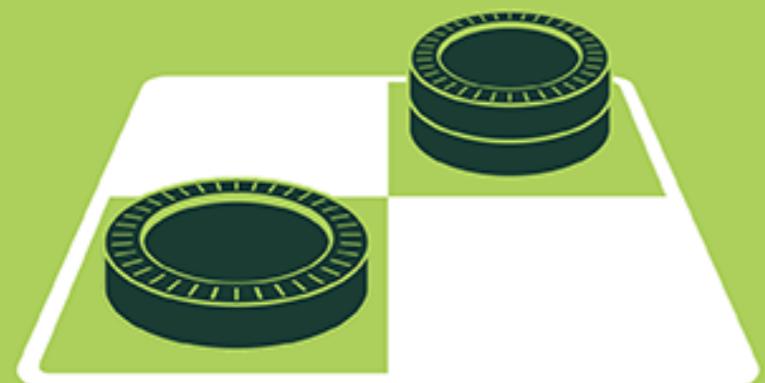
# Introduction to Machine Learning

---

Before we will dive deep into Unsupervised Learning...

# ARTIFICIAL INTELLIGENCE

Early artificial intelligence stirs excitement.



# MACHINE LEARNING

Machine learning begins to flourish.



# DEEP LEARNING

Deep learning breakthroughs drive AI boom.



1950's

1960's

1970's

1980's

1990's

2000's

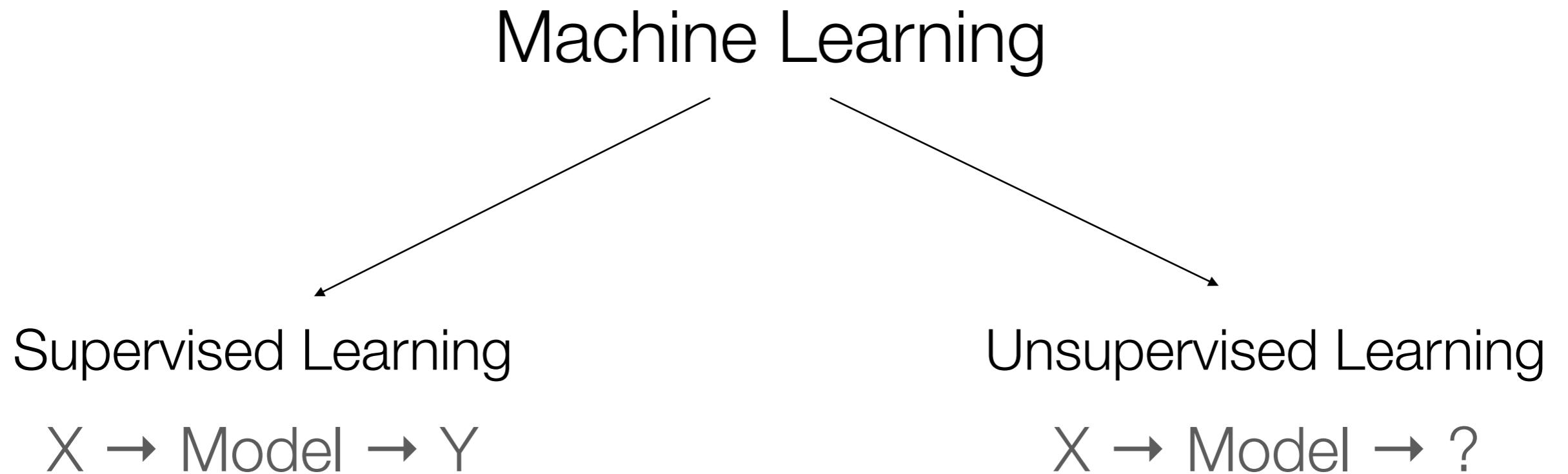
2010's

Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Source: Nvidia's blog

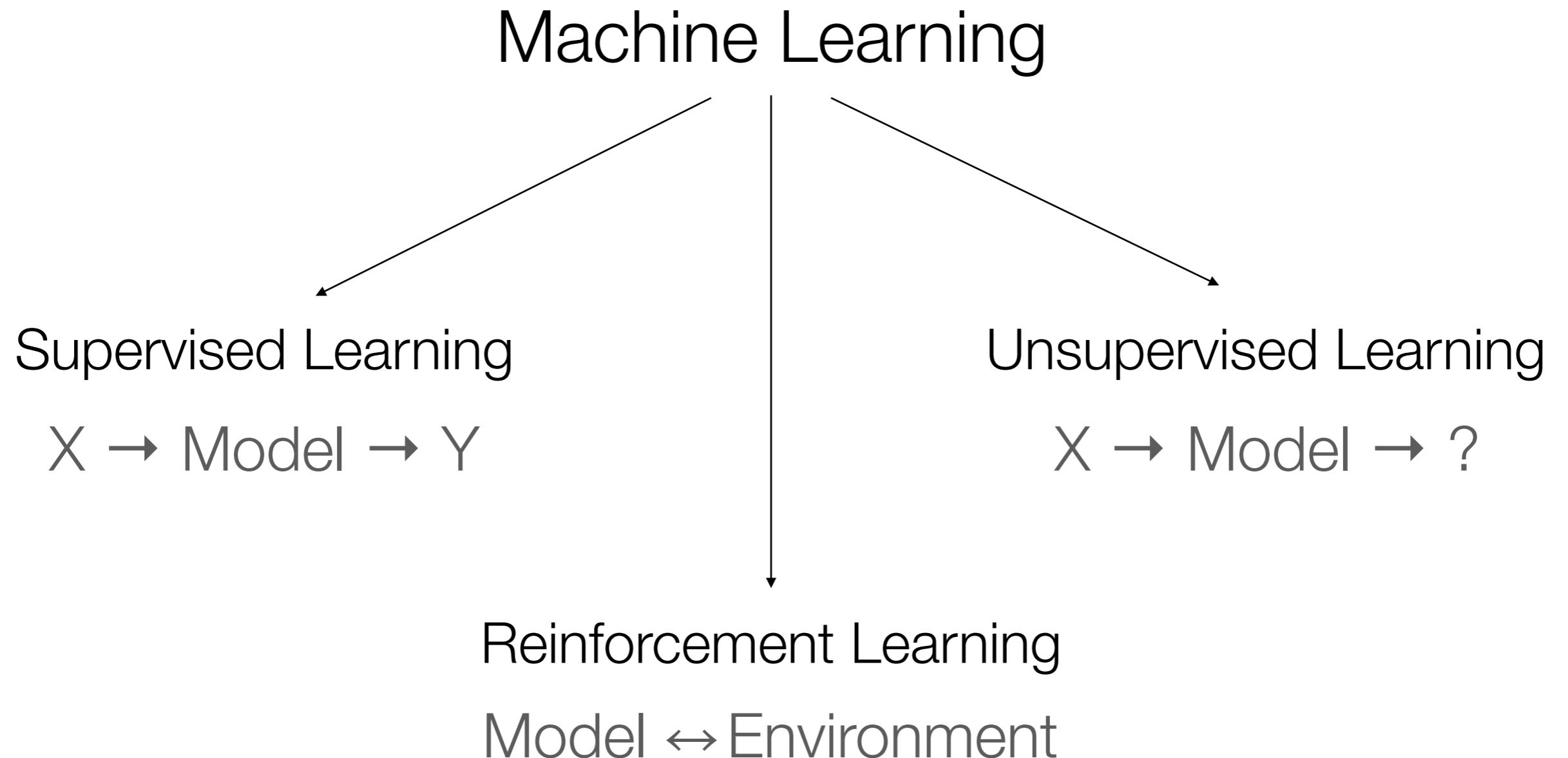
# Machine Learning Tasks

---



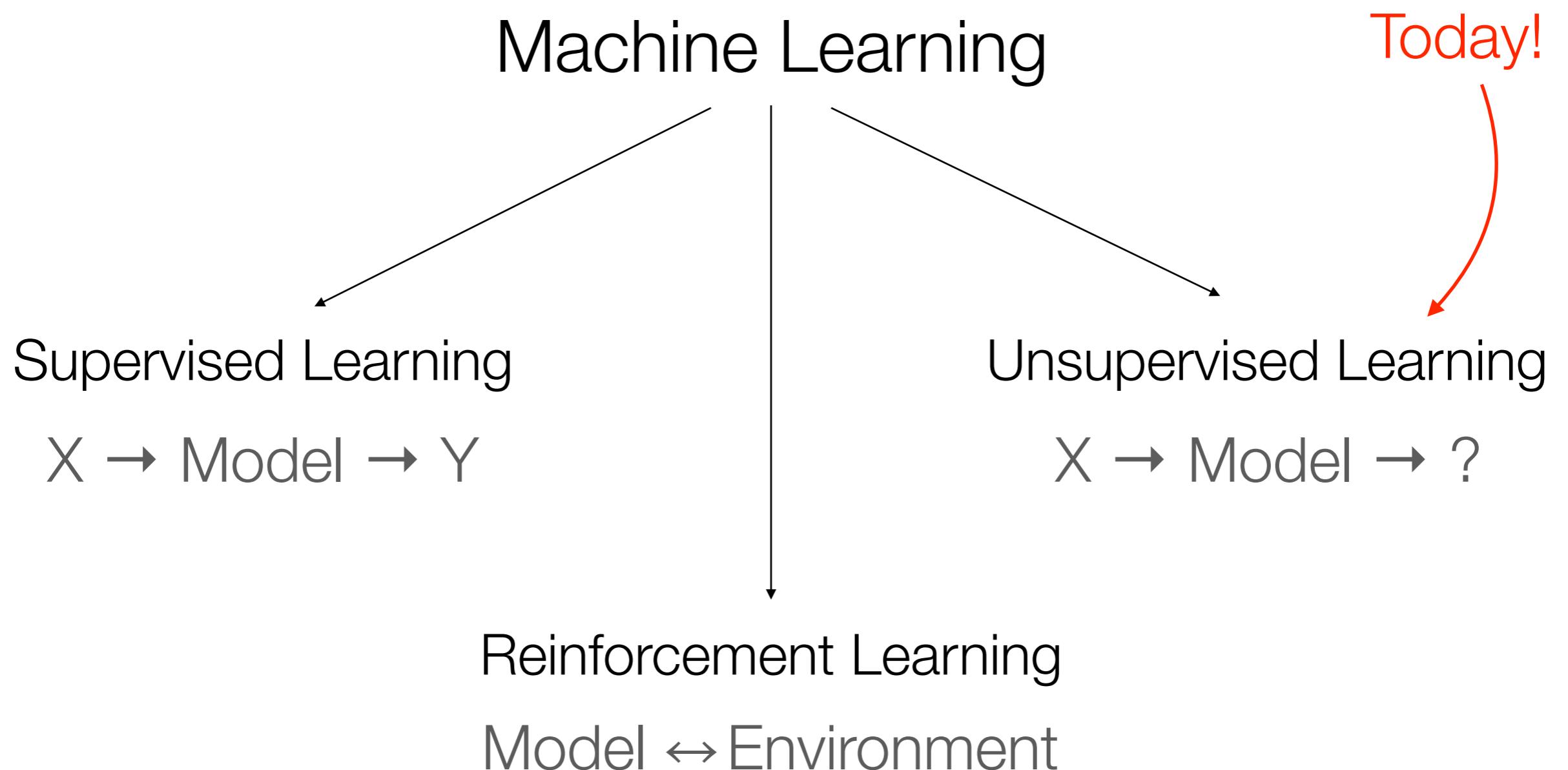
# Machine Learning Tasks

---



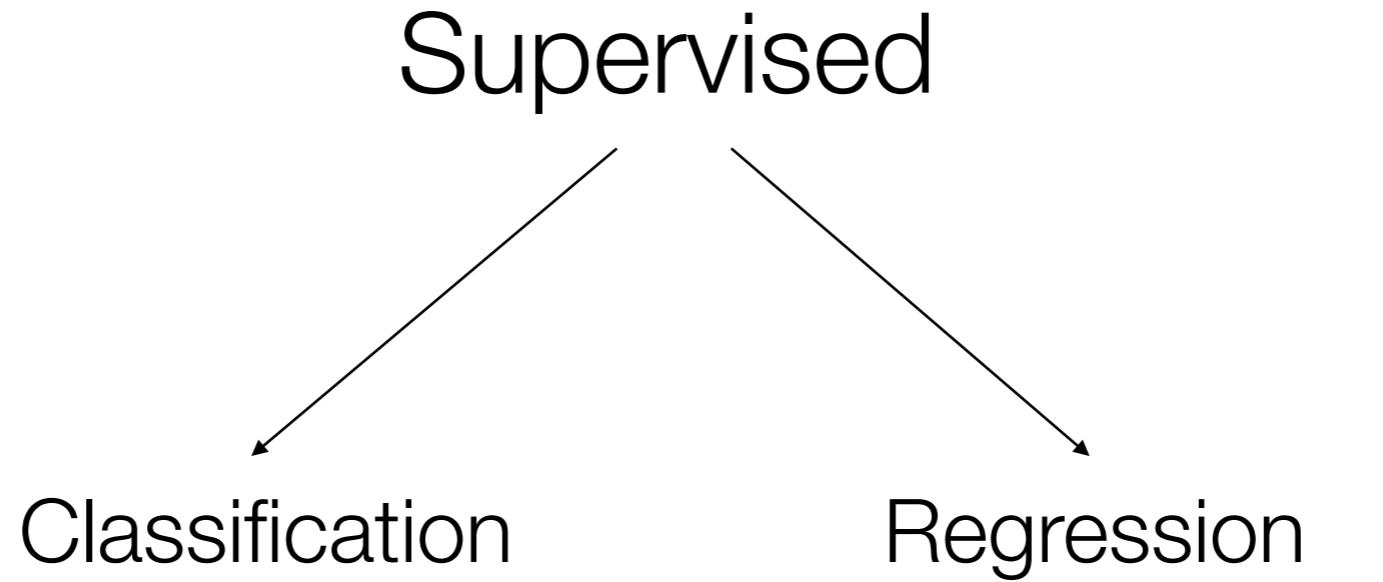
# Machine Learning Tasks

---

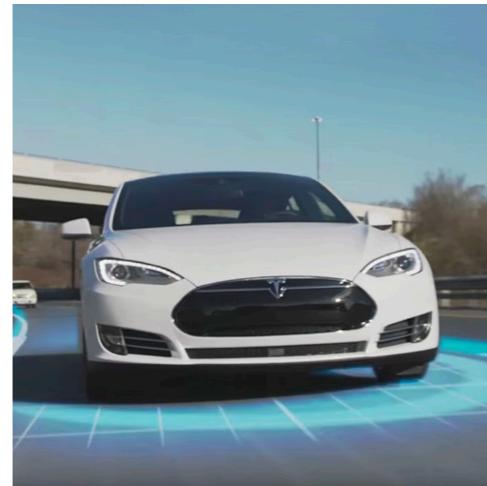


# Machine Learning Problems

---



“This is a cat!”



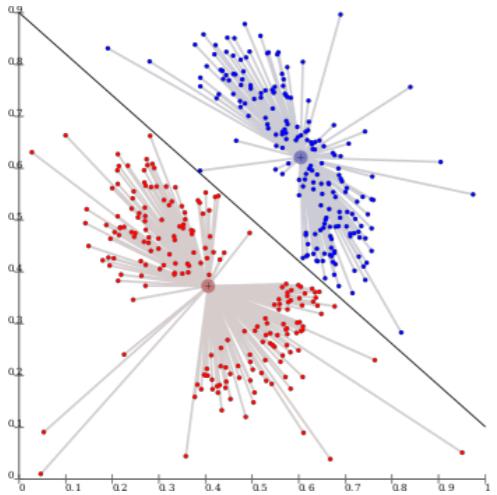
“5 degrees to the left”

# Machine Learning Problems

---

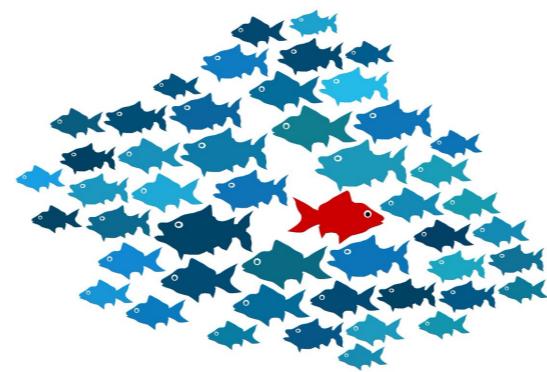
Unsupervised

Clustering



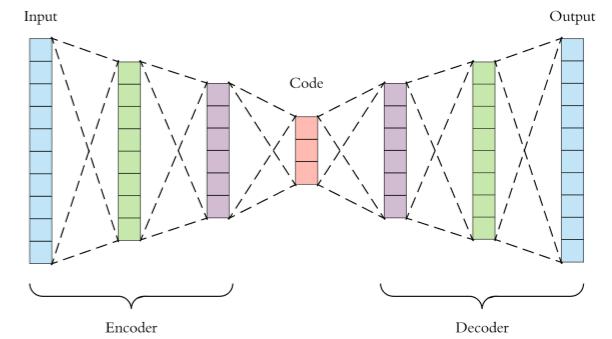
“Looks very similar  
to others”

Anomaly detection



“Looks abnormal...”

Autoencoders



“Here is the encoding  
for this example!”

Questions? 😊

# Clustering

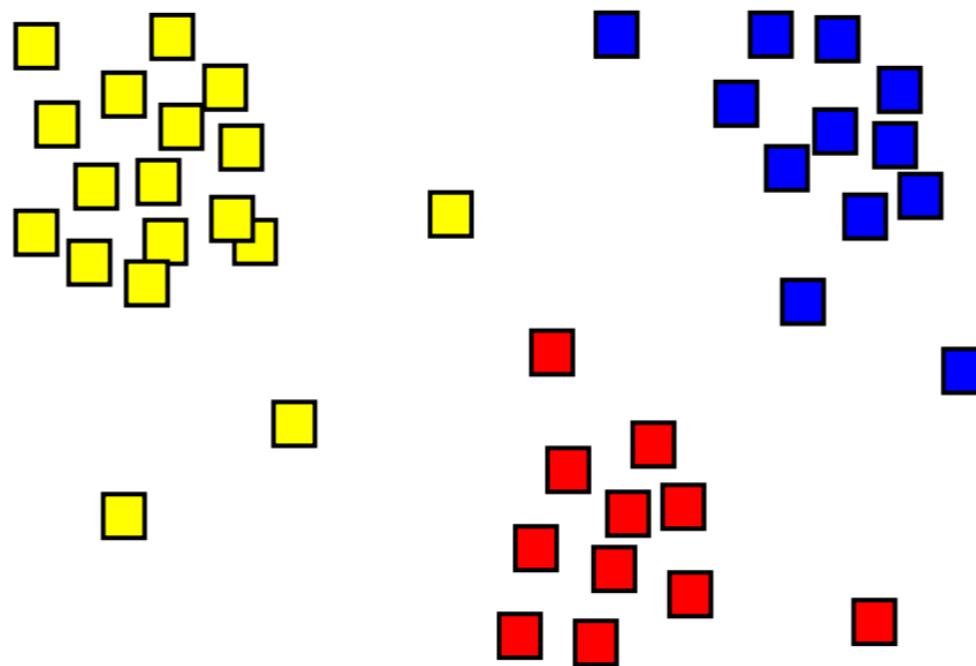
---

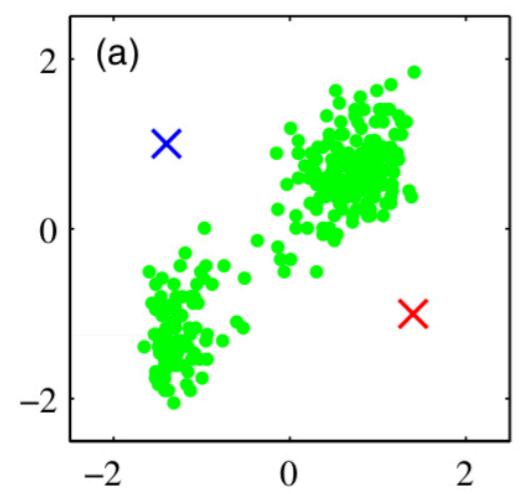
A few words about K-Means algorithm

# What is clustering?

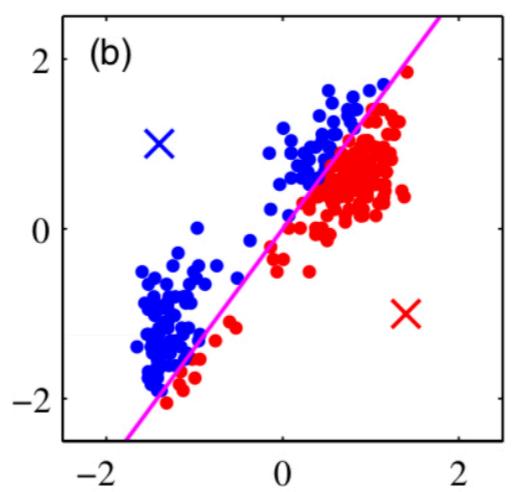
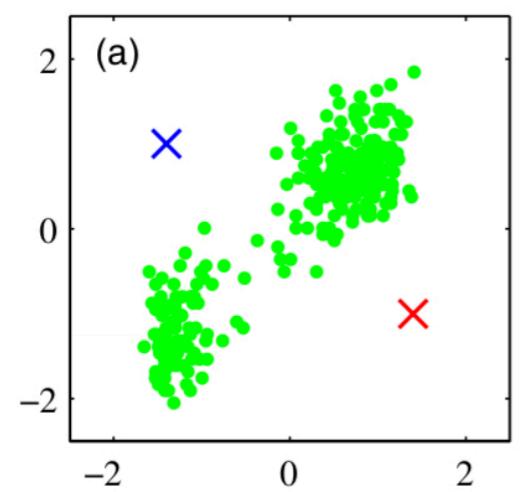
---

A clustering algorithm will help you **group the data** into distinct groups, guaranteeing that data points in each group are **similar to each other**.

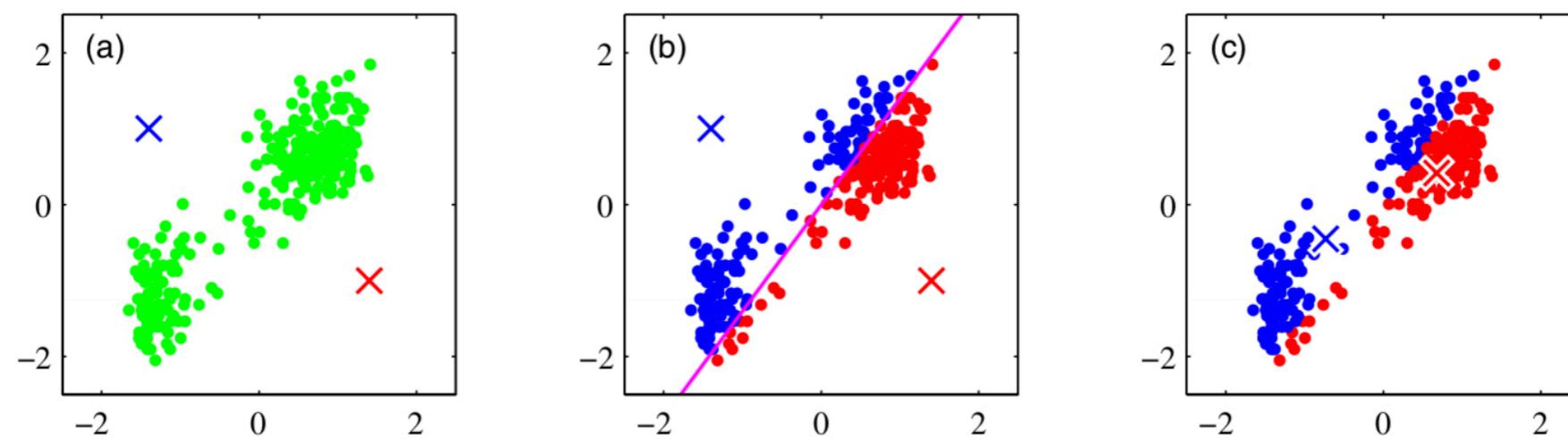




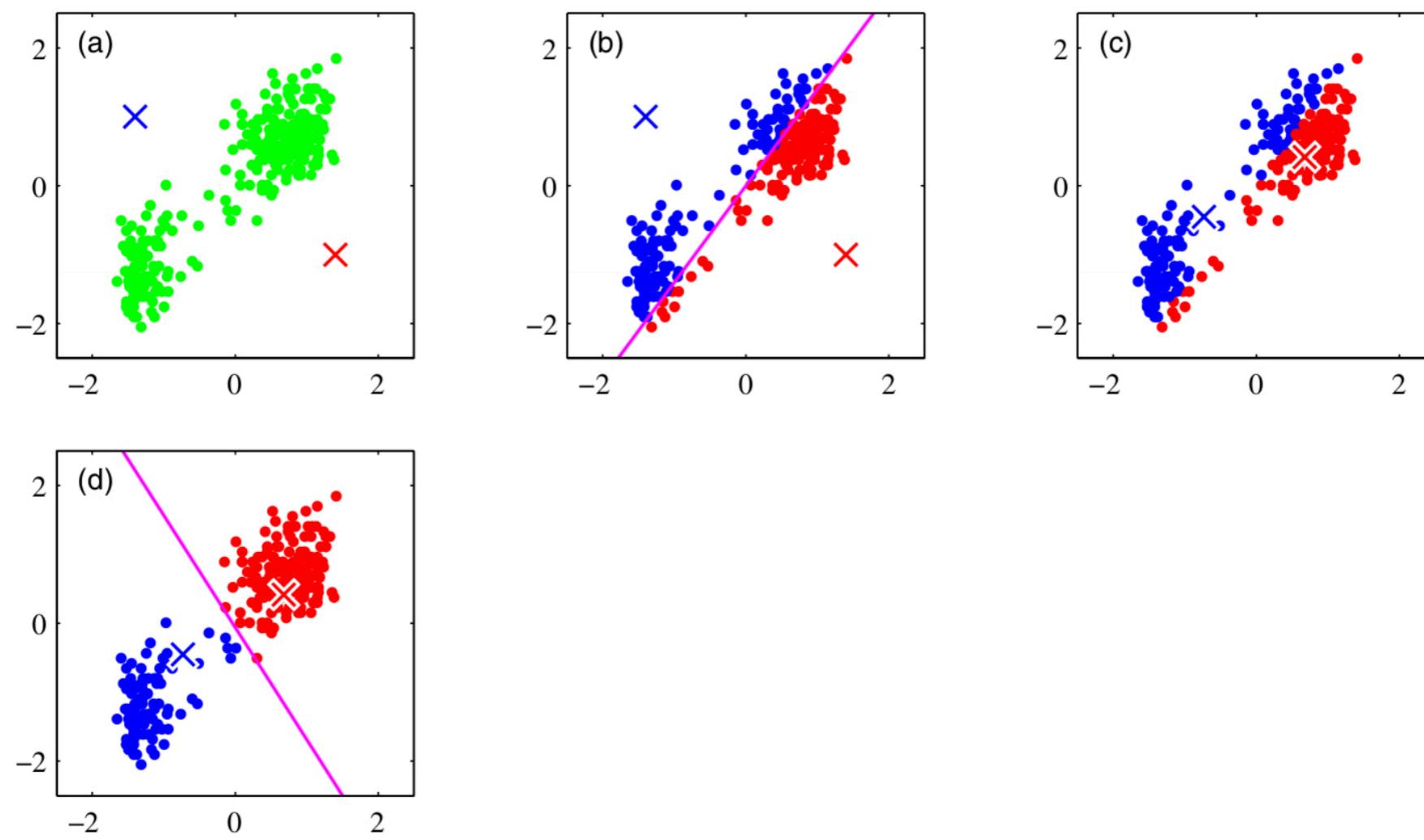
Source: *Pattern Recognition and Machine Learning*



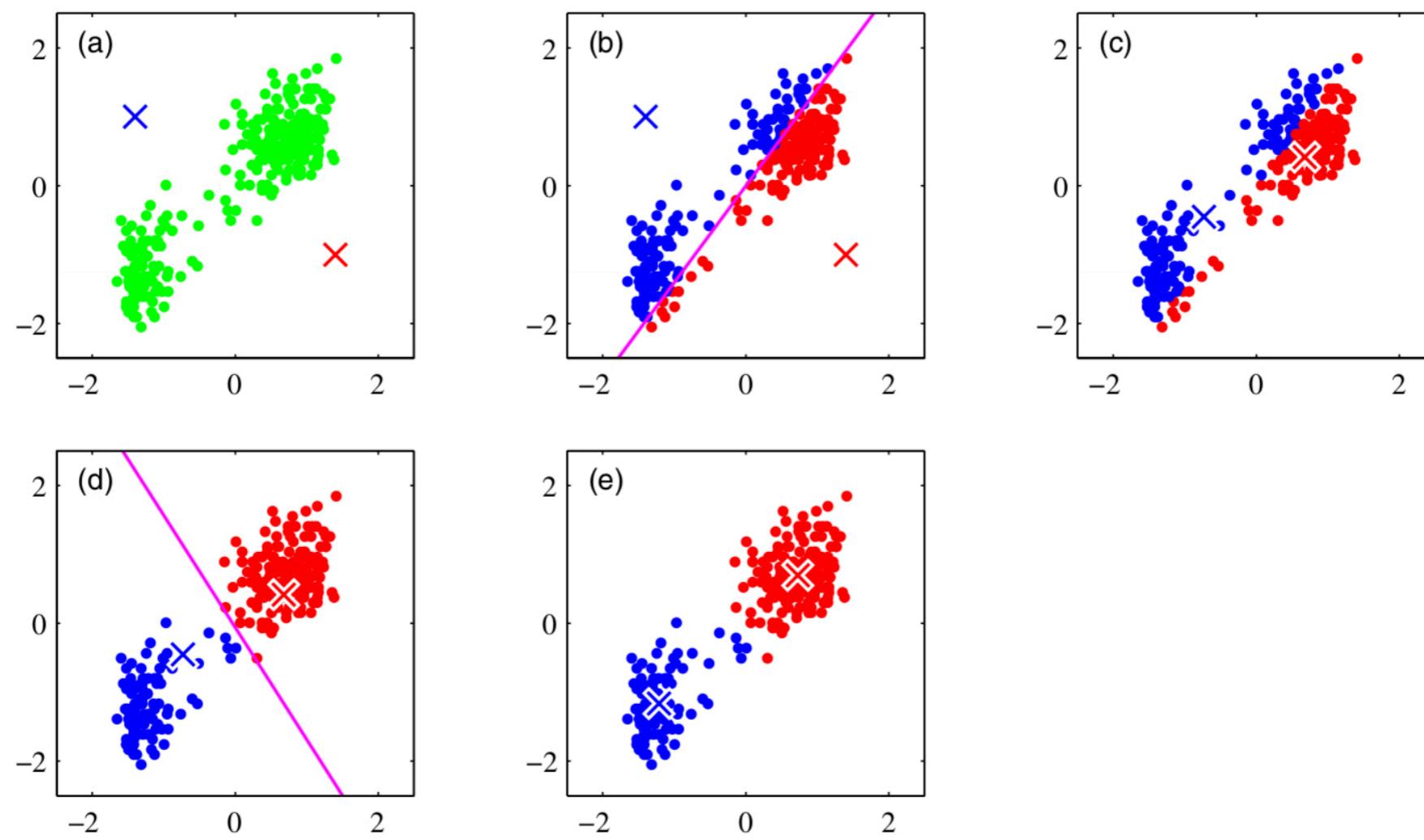
Source: *Pattern Recognition and Machine Learning*



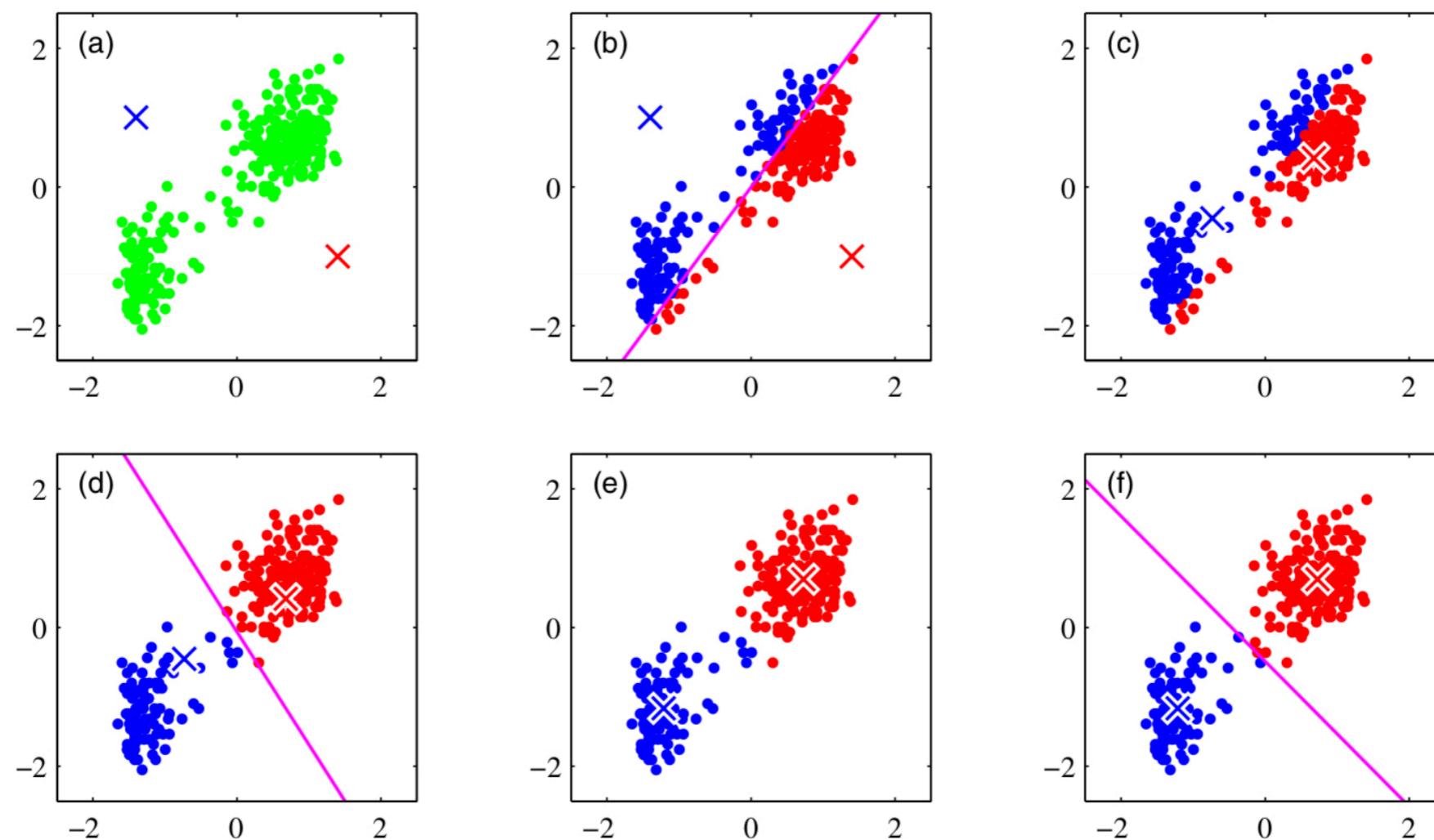
Source: *Pattern Recognition and Machine Learning*



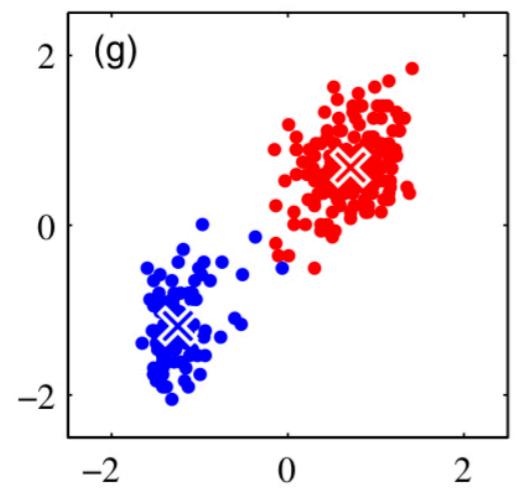
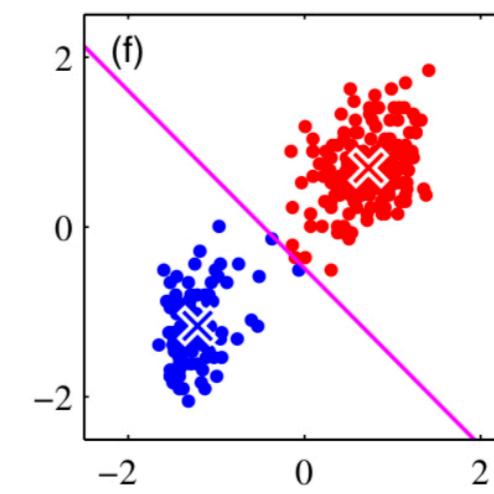
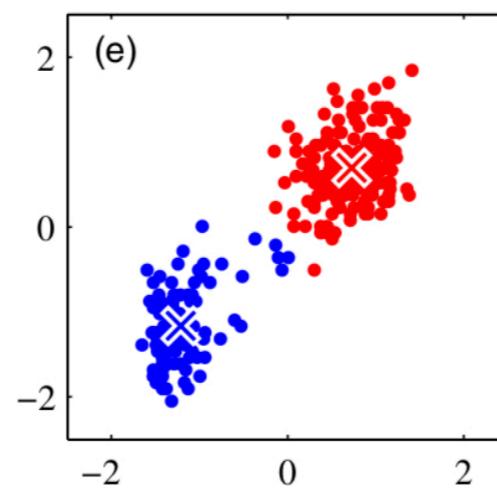
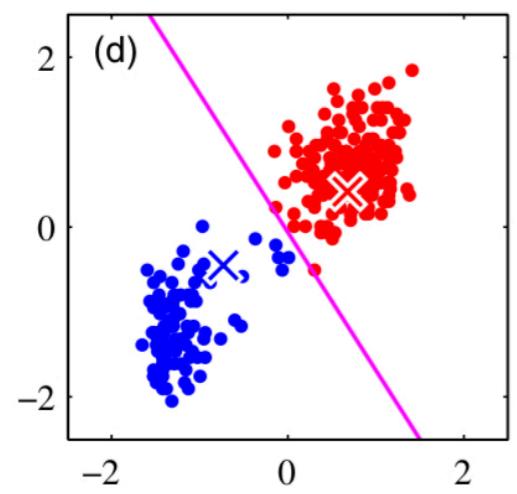
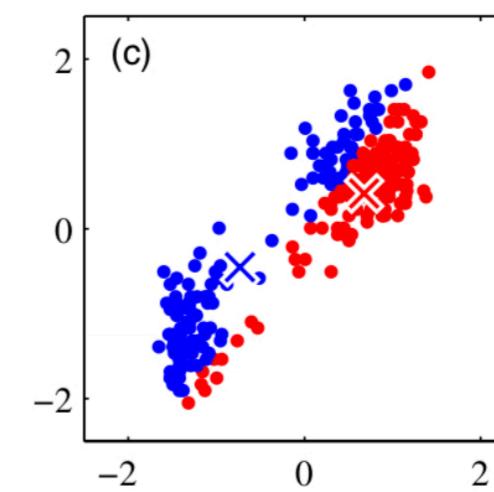
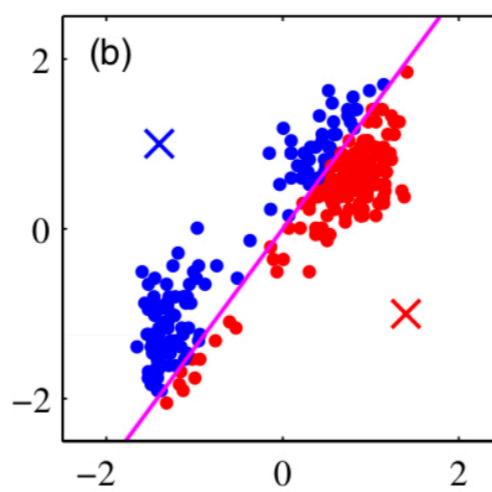
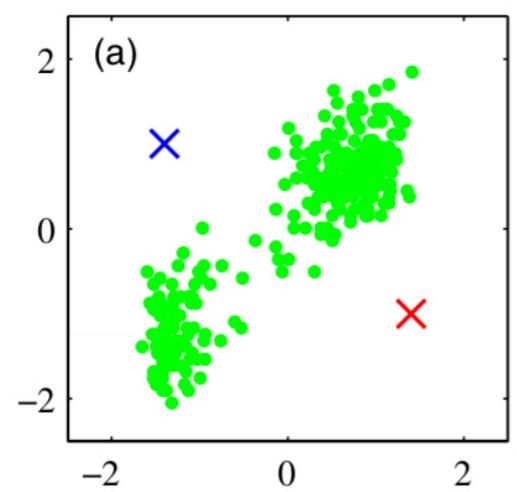
Source: *Pattern Recognition and Machine Learning*



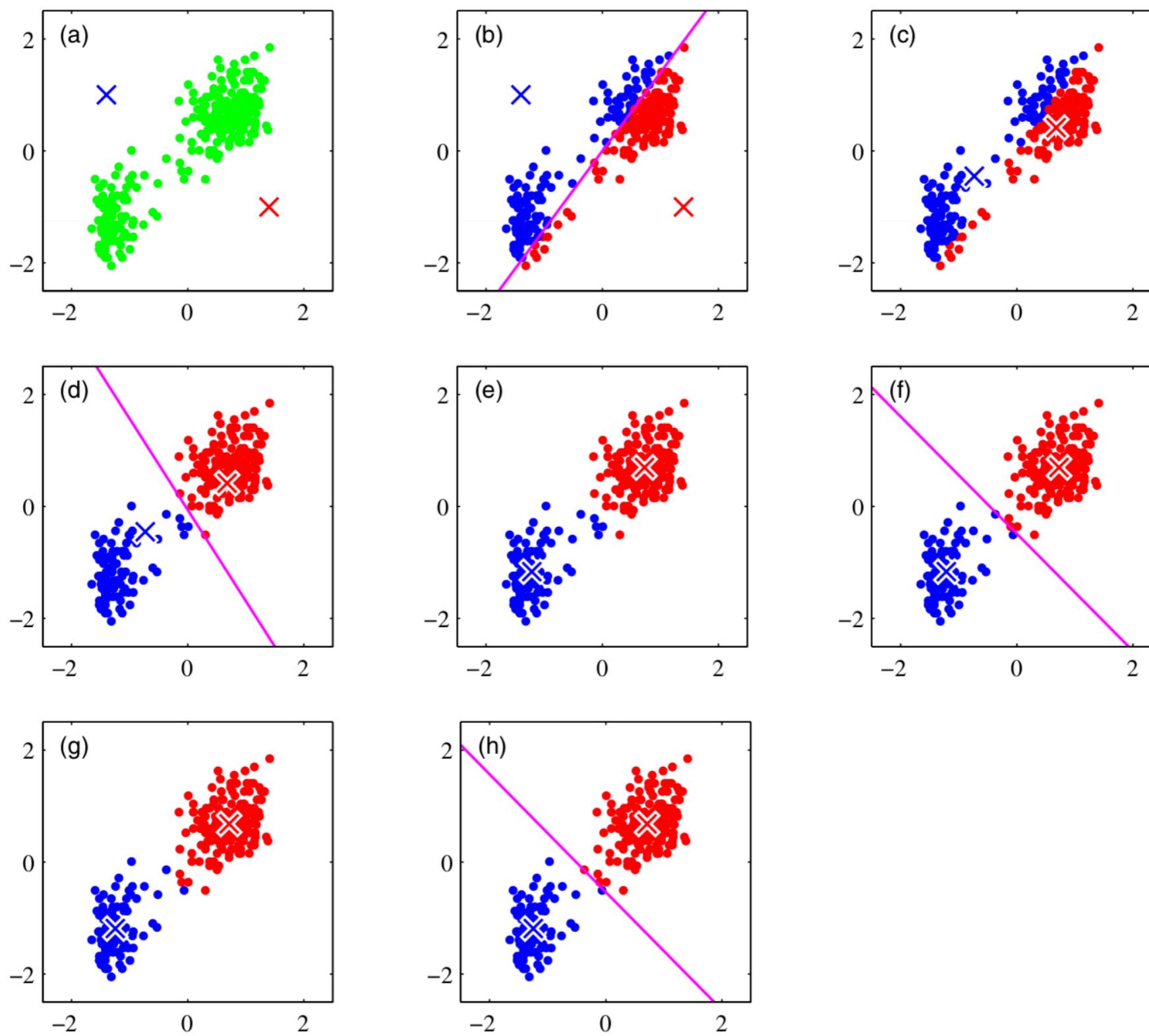
Source: *Pattern Recognition and Machine Learning*



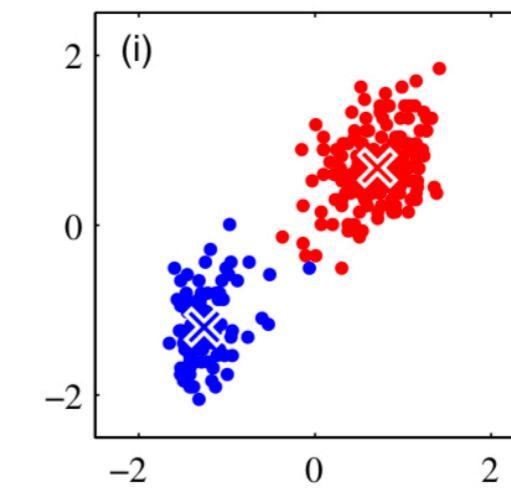
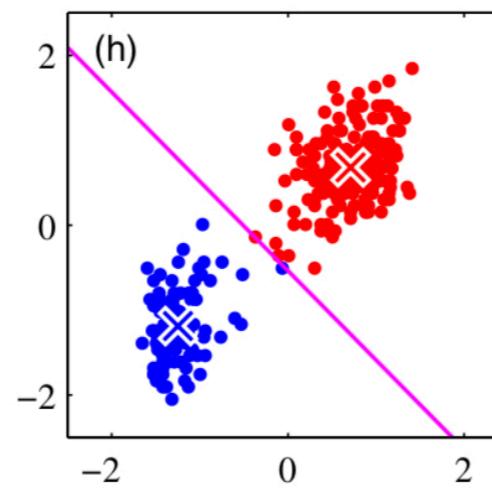
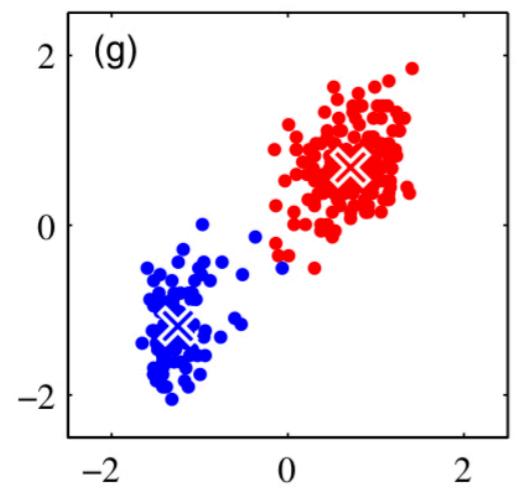
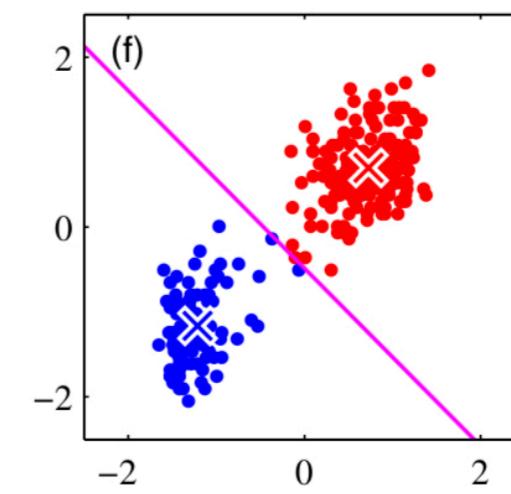
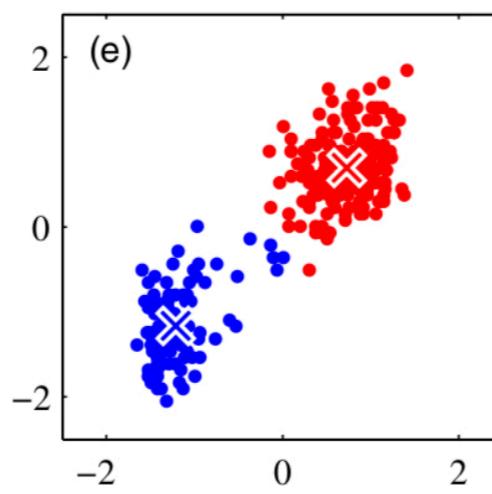
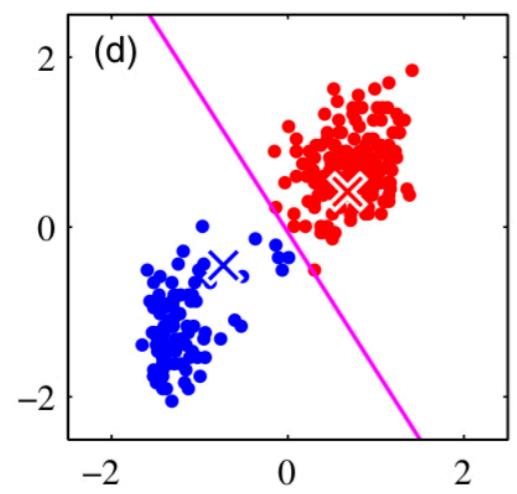
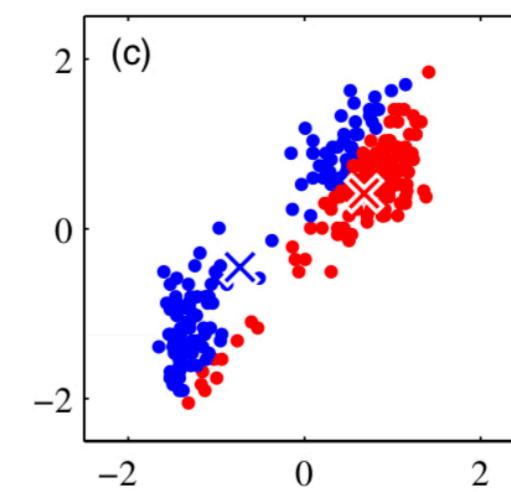
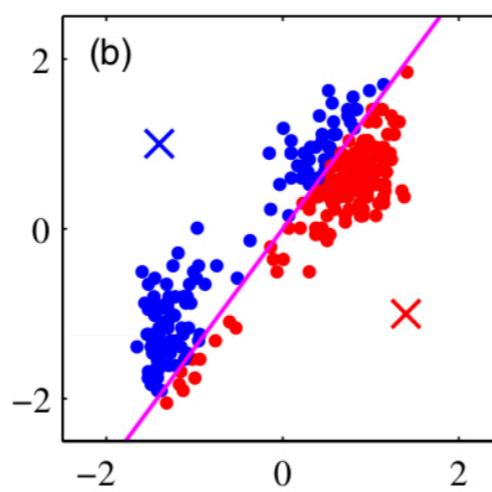
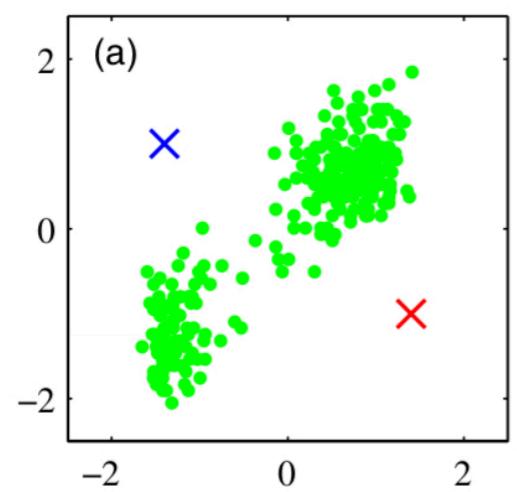
Source: *Pattern Recognition and Machine Learning*



Source: *Pattern Recognition and Machine Learning*



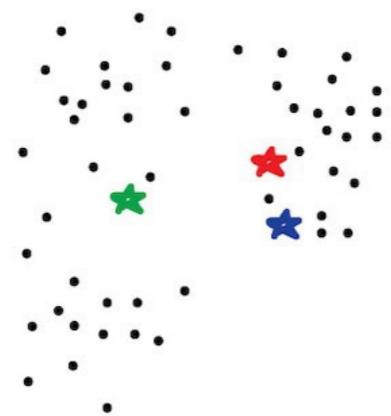
Source: *Pattern Recognition and Machine Learning*



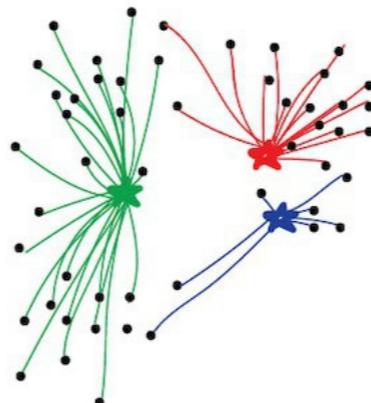
Source: Pattern Recognition and Machine Learning

## PUT KEBAB KIOSKS IN THE OPTIMAL WAY

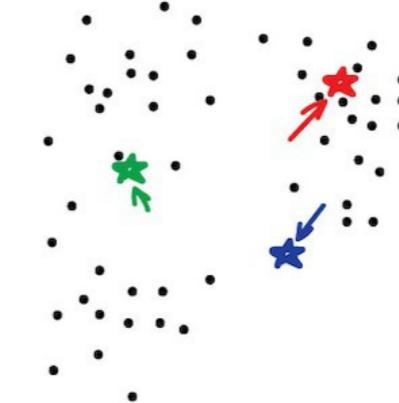
(also illustrating the K-means method)



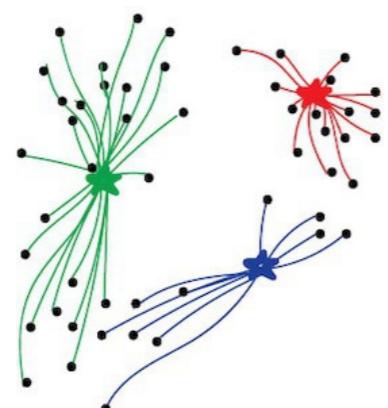
1. Put kebab kiosks in random places in city



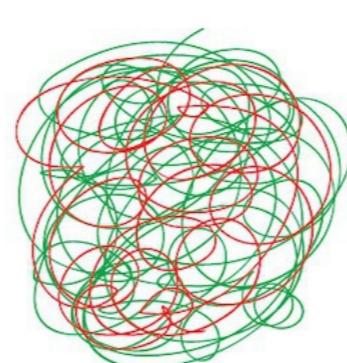
2. Watch how buyers choose the nearest one



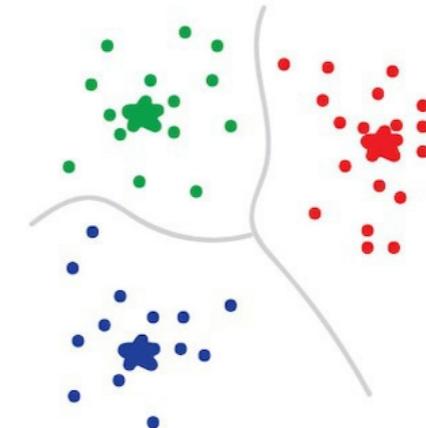
3. Move kiosks closer to the centers of their popularity



4. Watch and move again



5. Repeat a million times



6. Done!  
You're god of kebabs!

Source: Vas3k's blog

# K-Means Algorithm

---

1. Select K clusters randomly,
2. For each point select the closest cluster,
3. Update your clusters by taking mean value of all your data points,
4. Check for convergence and move to Step 2 if needed.

# How to measure the distance?

---

Euclidean distance

$$\|a - b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$$

Squared Euclidean distance

$$\|a - b\|_2^2 = \sum_i (a_i - b_i)^2$$

Manhattan distance

$$\|a - b\|_1 = \sum_i |a_i - b_i|$$

# Why/Why not to use K-Means Clustering?

---

## **Advantages:**

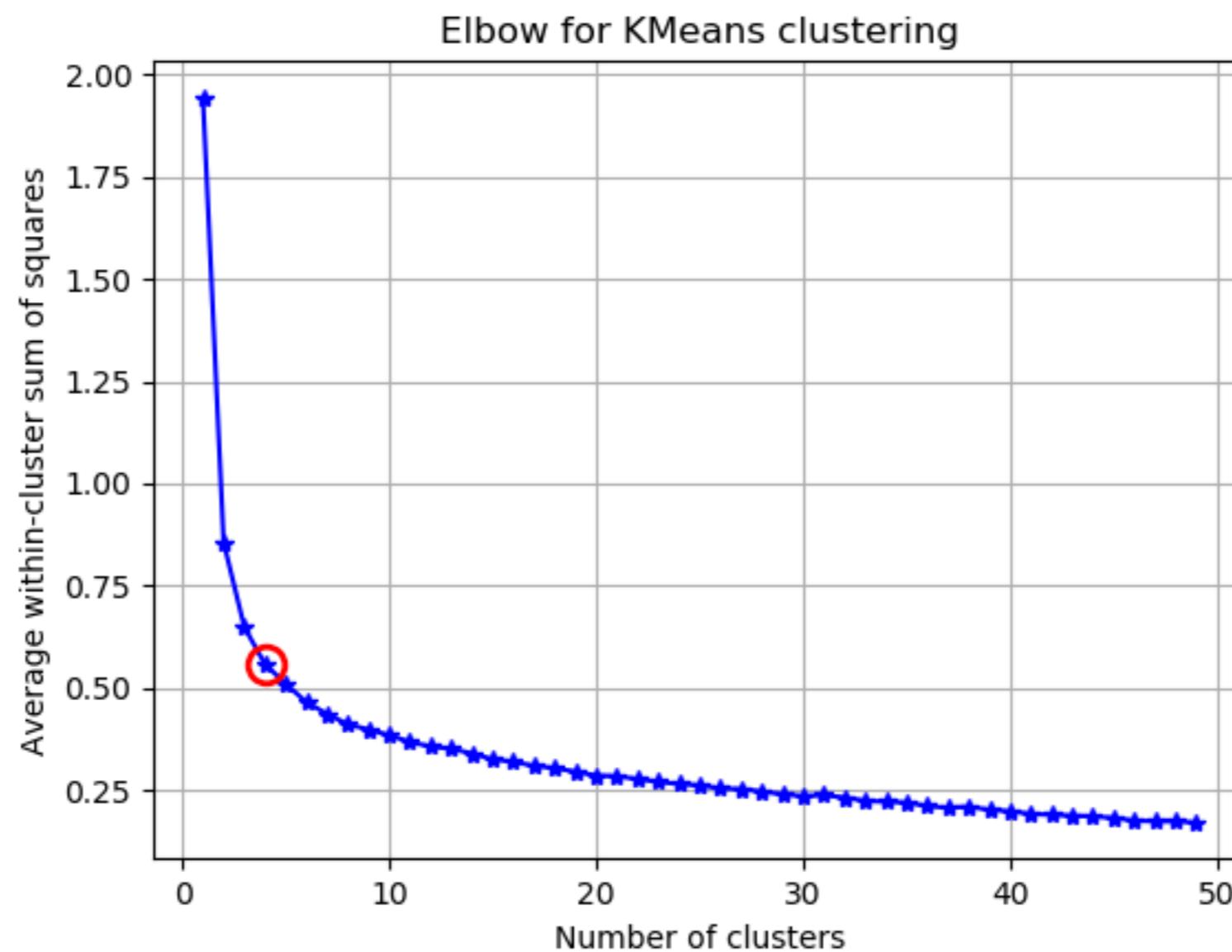
- Easy to implement,
- Relatively fast and efficient,
- Only one hyperparameter - K.

## **Disdvantages:**

- Hyperparameter for number of clusters,
- Random points at the beginning.

# How to choose the number of clusters?

---



Let's use it in practice! 

---

[TODO] LINK TO GOOGLE COLAB

Questions? 😊

# Hierarchical Clustering

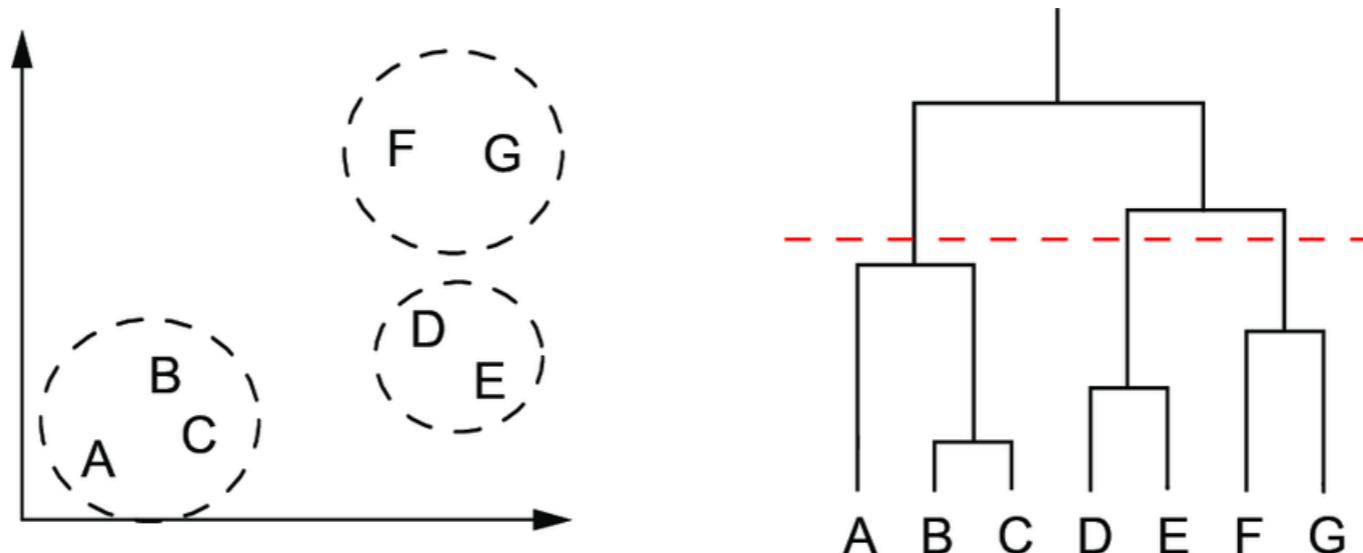
---

Is it really different than K-Means?

# Hierarchical Clustering

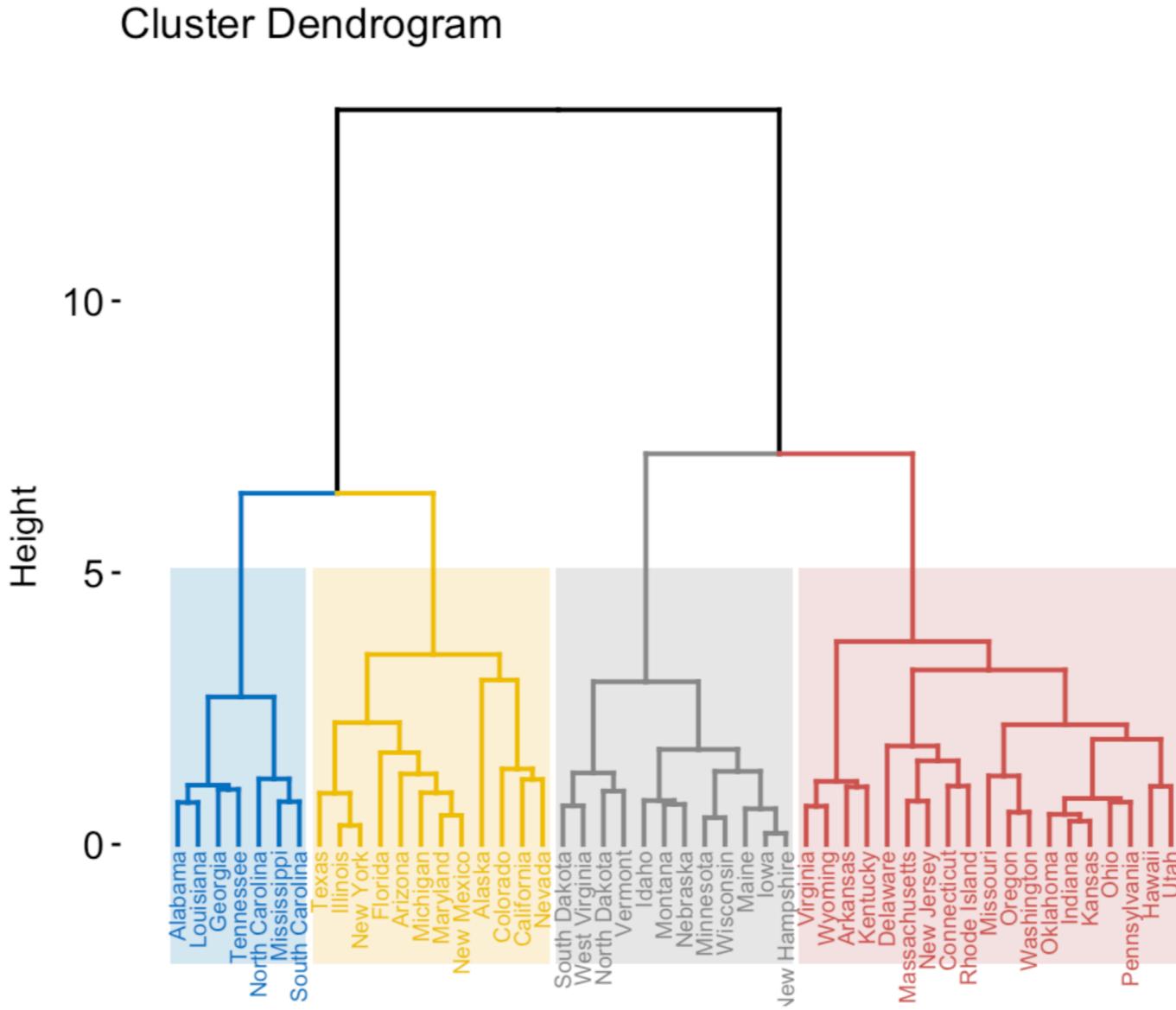
---

- Hierarchy of clusters is represented as a **tree**,
- **Root** is a unique cluster that gathers all samples,
- **Leaves** are clusters with only one sample.

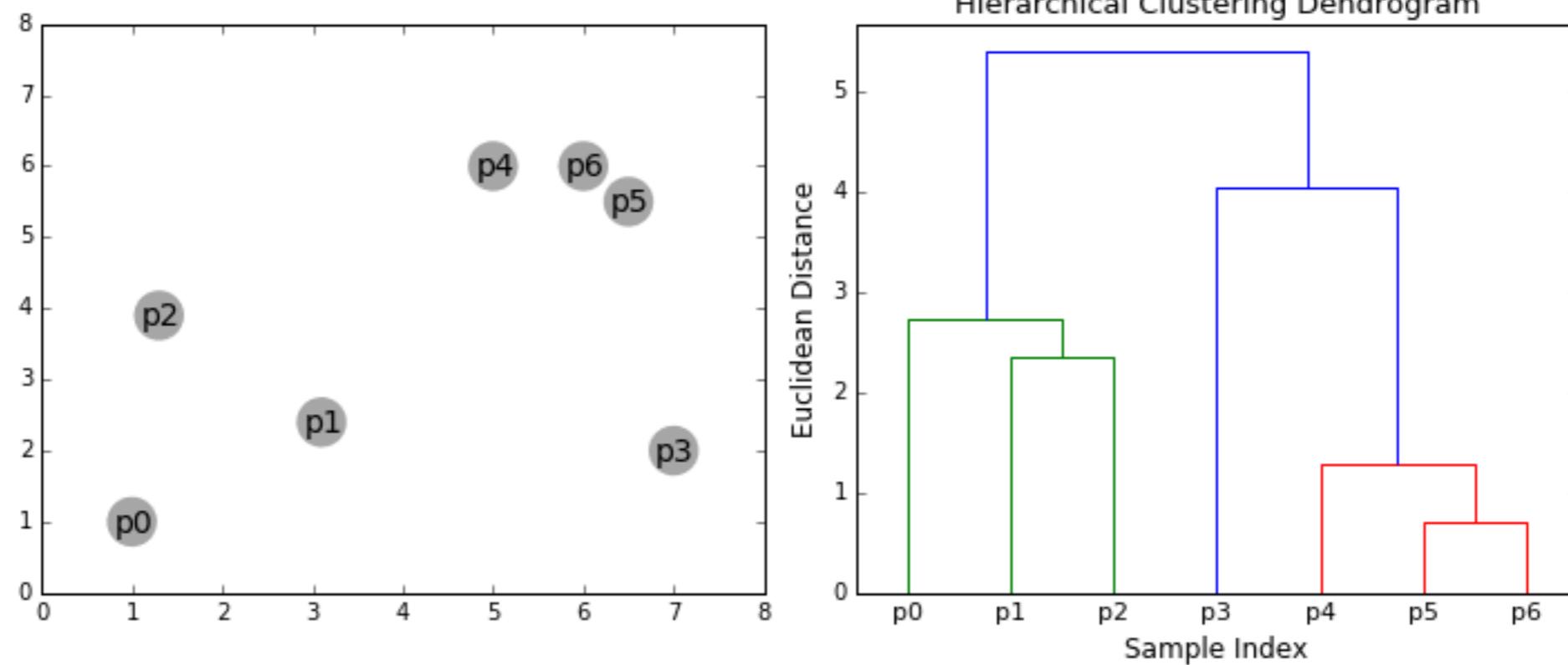


# Dendrogram is your friend

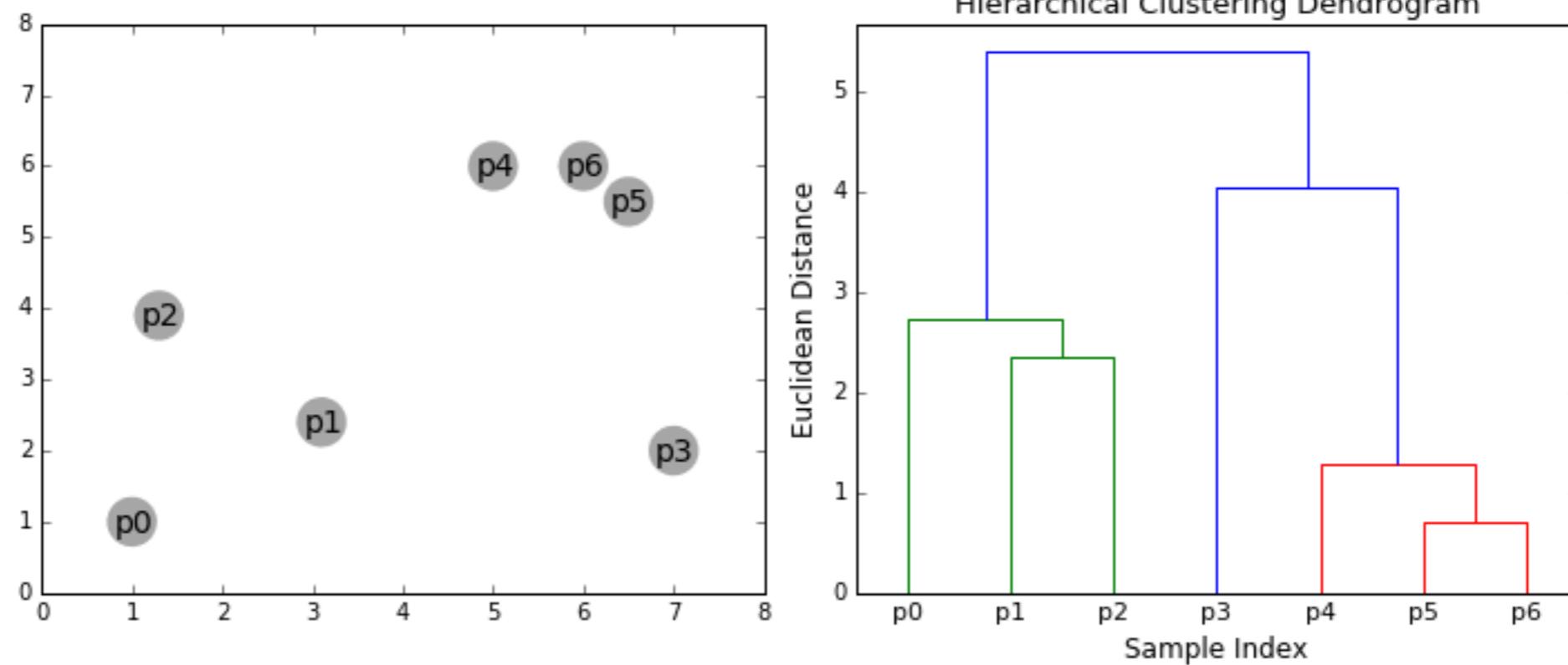
---



Source: Practical Guide to Cluster Analysis in R



Source: George Seif's blogpost on Towards Data Science



Source: George Seif's blogpost on Towards Data Science

## **Agglomerative**

Each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

## **Divisive**

All observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

# Agglomerative Clustering Algorithm

---

1. Create separate cluster for each example in the dataset,
2. Find two closest clusters.
3. Merge them into one cluster.
4. Move to Step 2 if you have more than one cluster.

# Why/Why not to use Hierarchical Clustering?

---

## **Advantages:**

- No need to specify number of clusters a'priori,
- Works well with different metrics,
- The best for hierarchical data structures.

## **Disdvantages:**

- Very low efficiency...

Let's use it in practice! 

---

[TODO] LINK TO GOOGLE COLAB

Questions? 😊

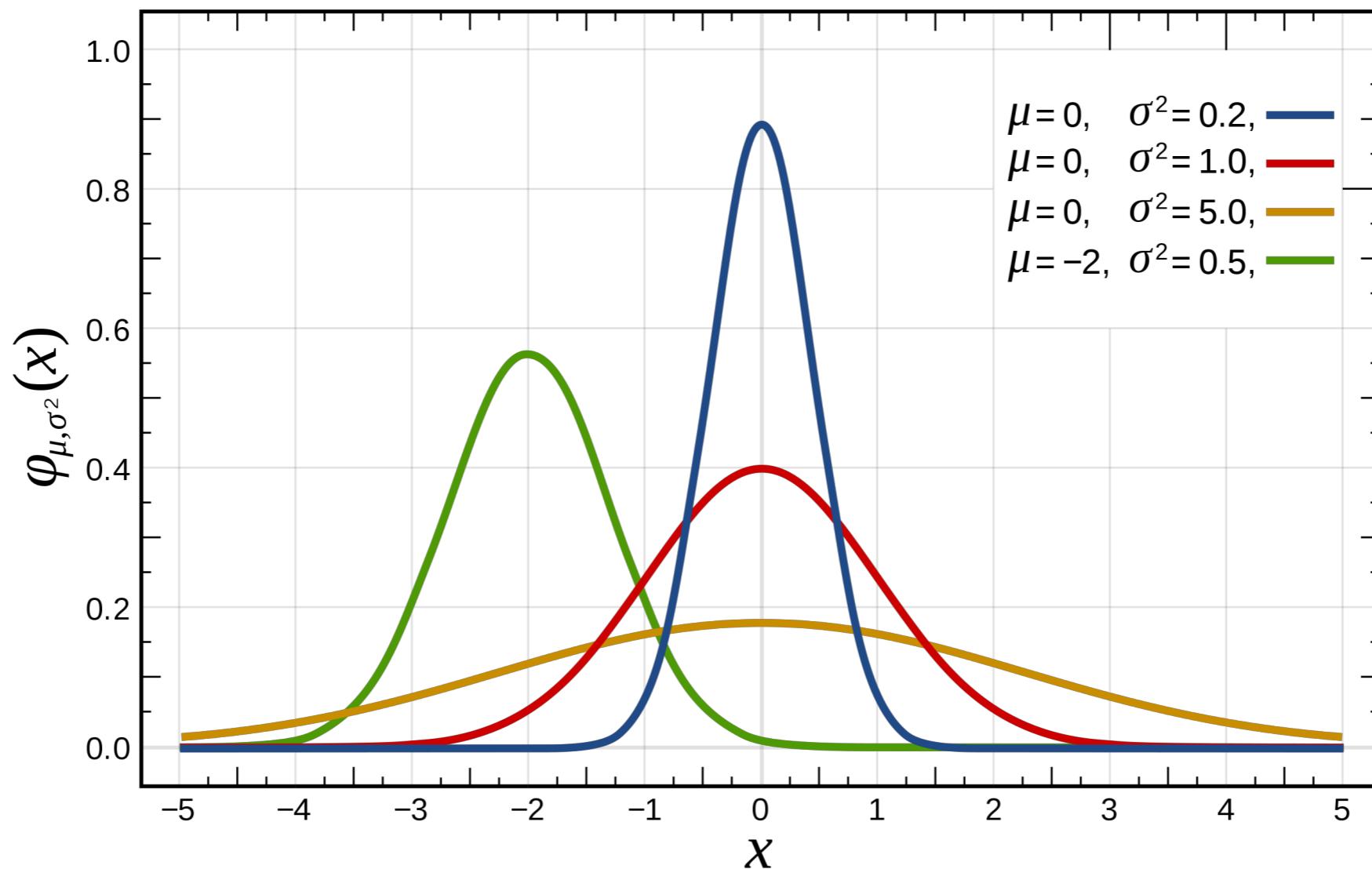
# Anomaly Detection

---

Gaussian Mixture Models

# Gaussian Distribution as a base

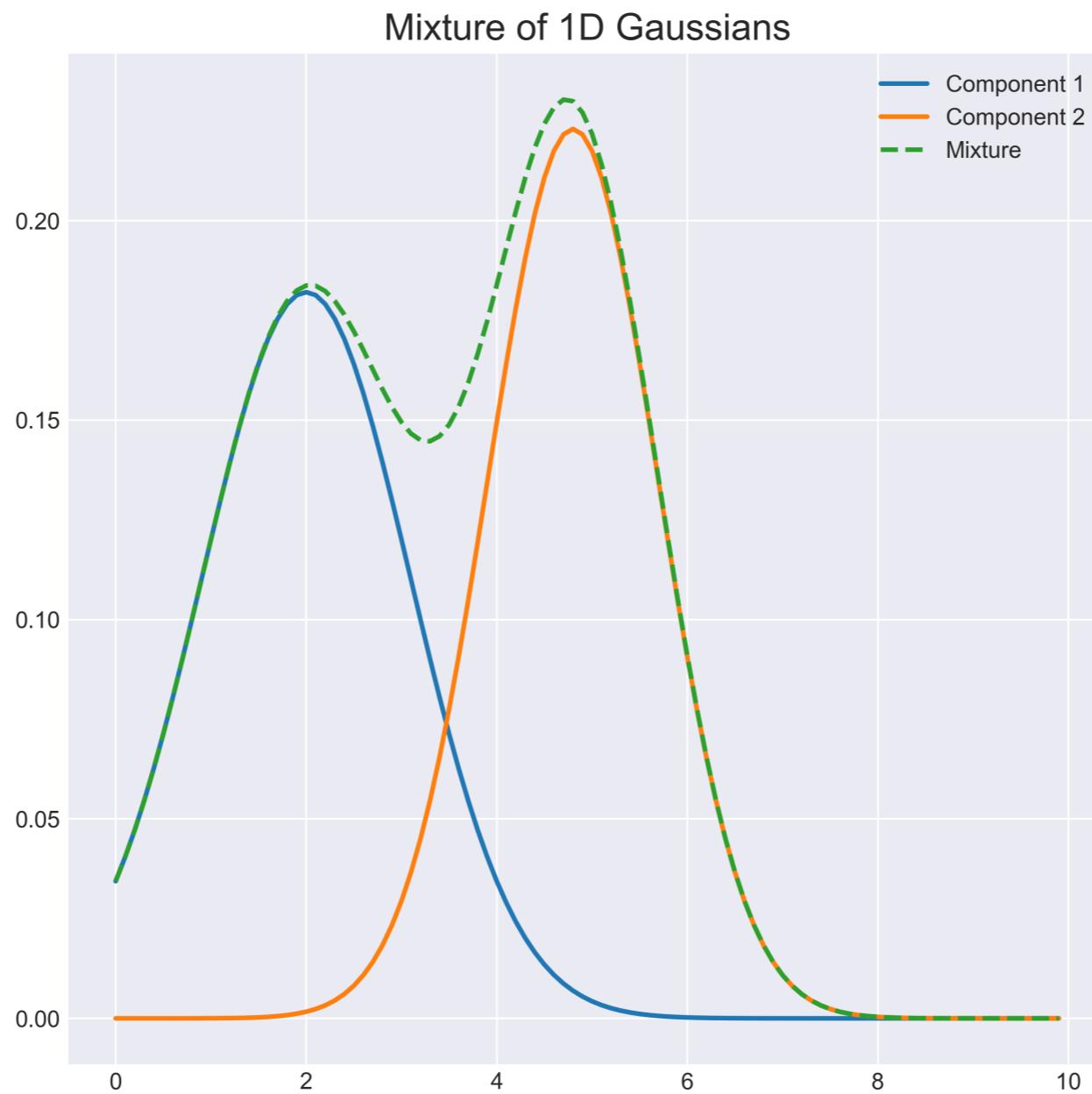
---



Source: Wikipedia

# Mixture of Gaussians

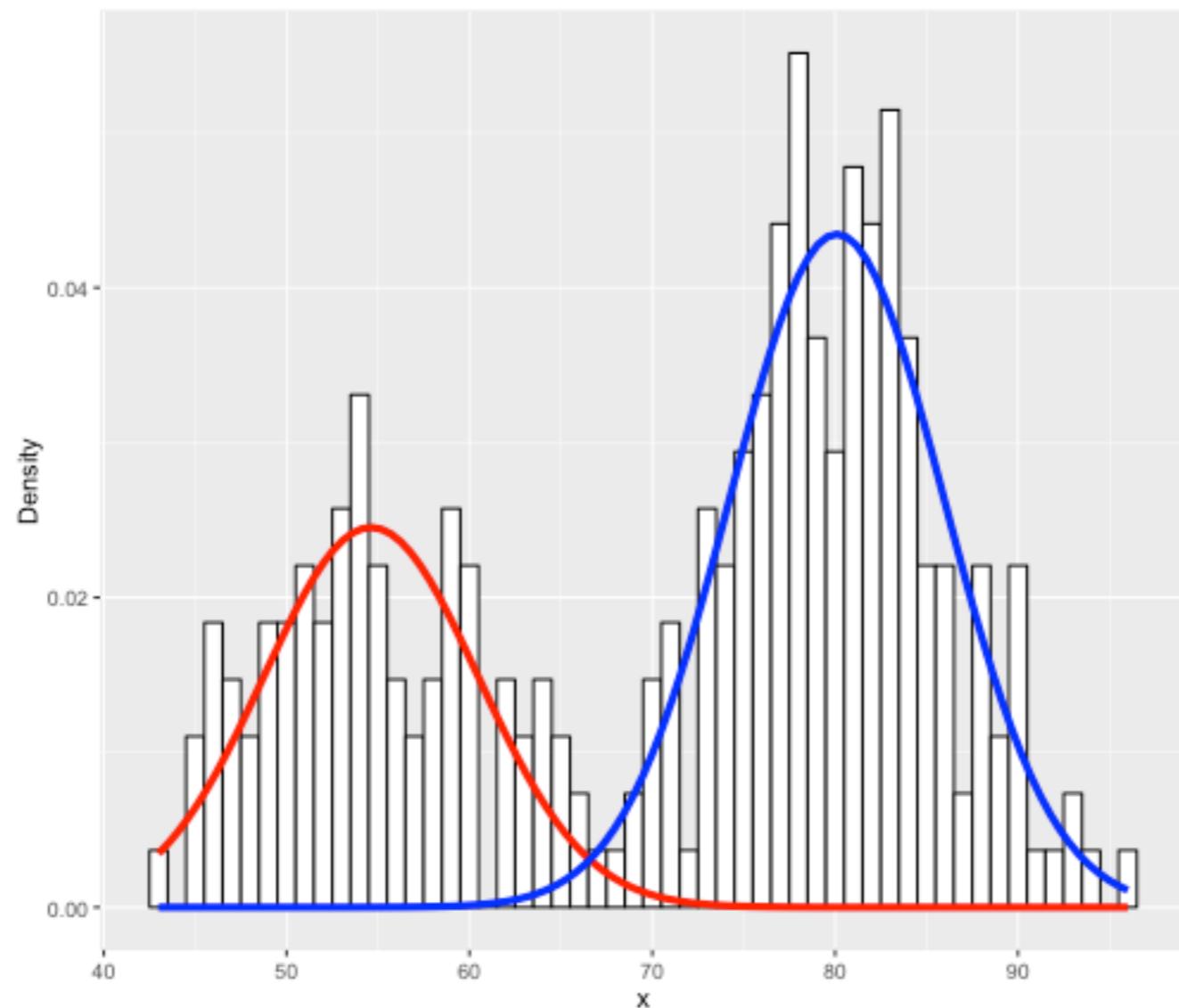
---



Source: Angus Turner's blog

# Mixture of Gaussians

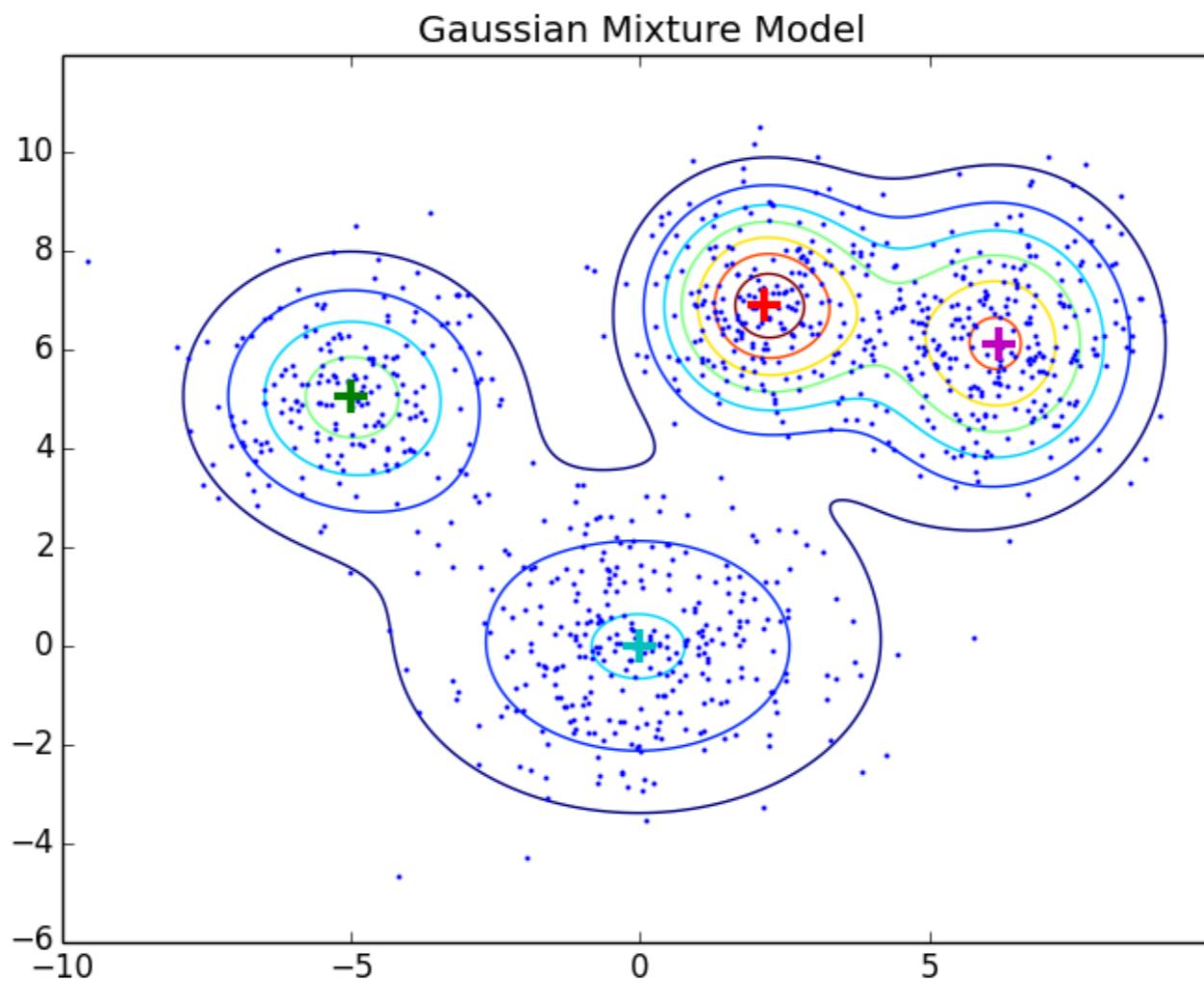
---



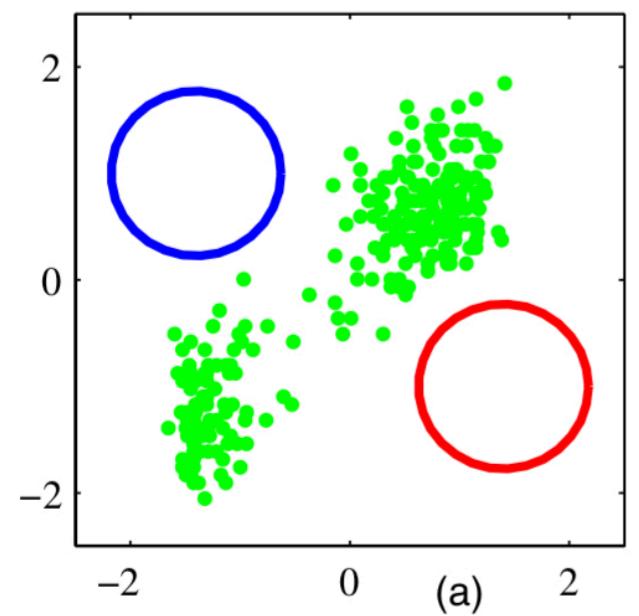
Source: Fong Chun Chan's blog

# Mixture of Gaussians

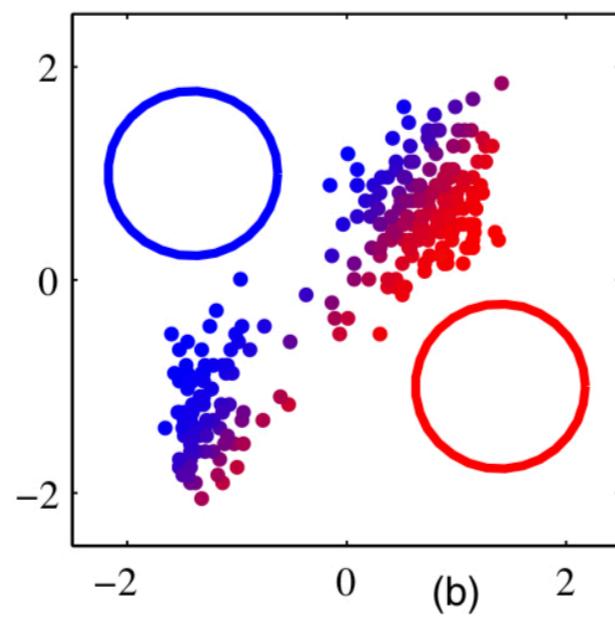
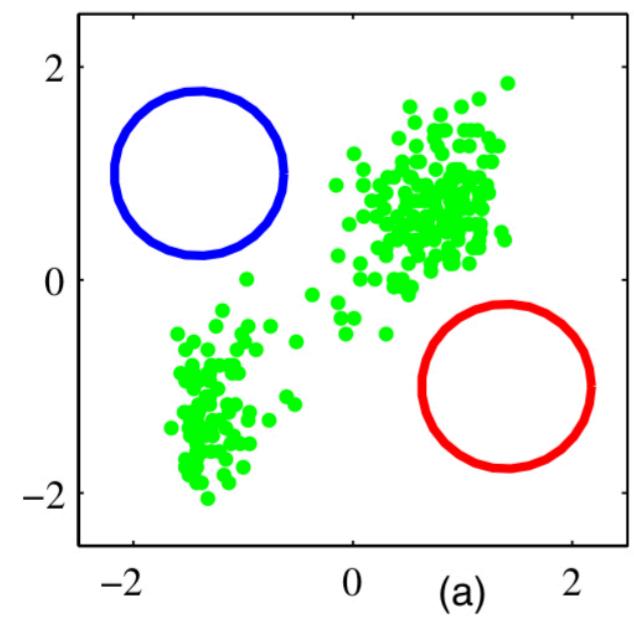
---



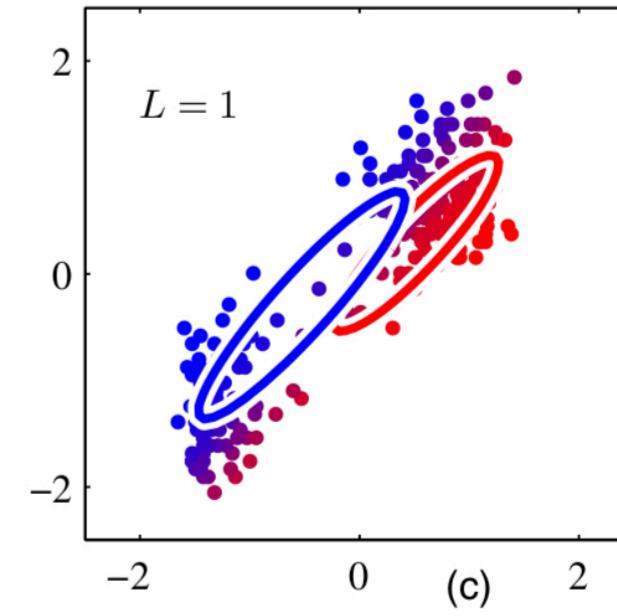
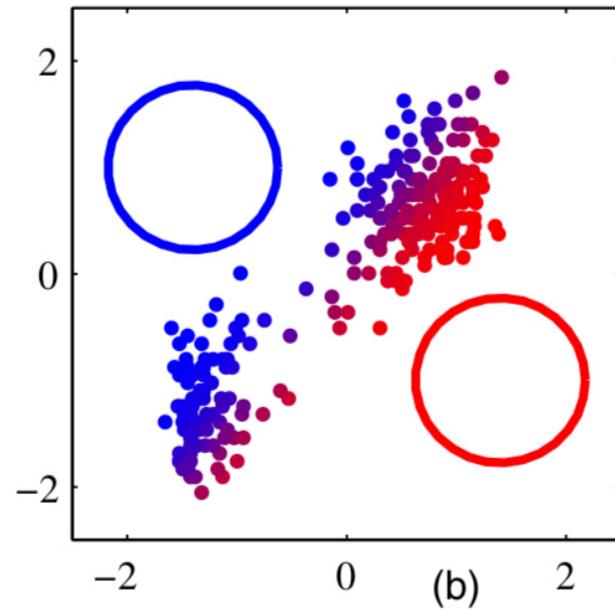
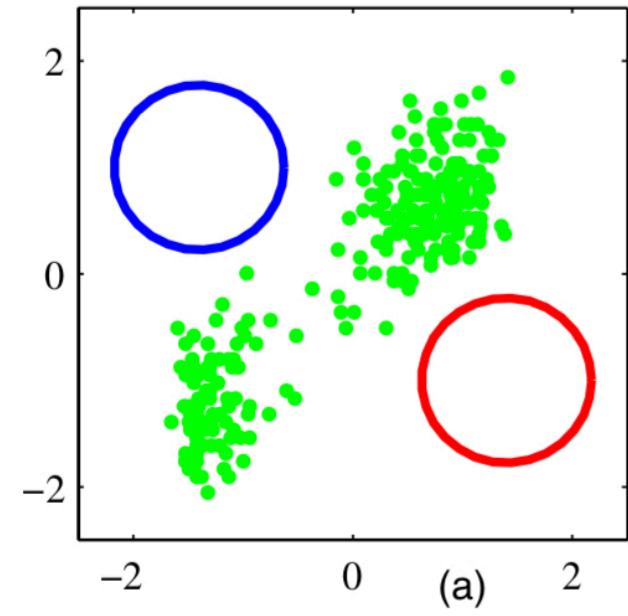
Source: Yu's Machine Learning Garden blog



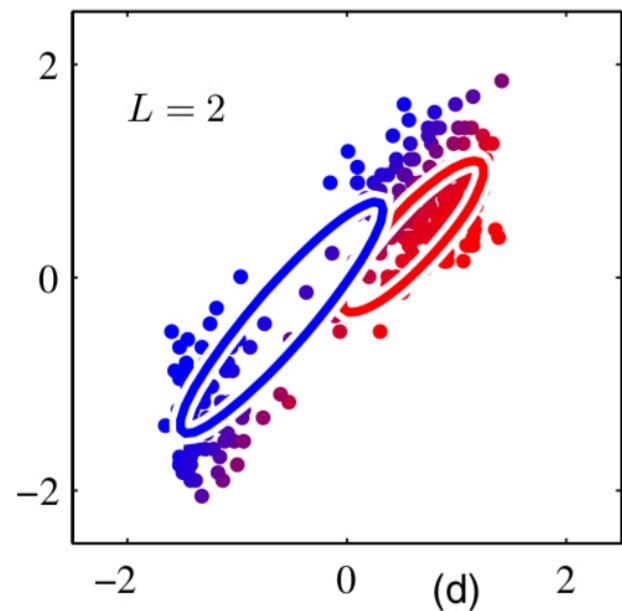
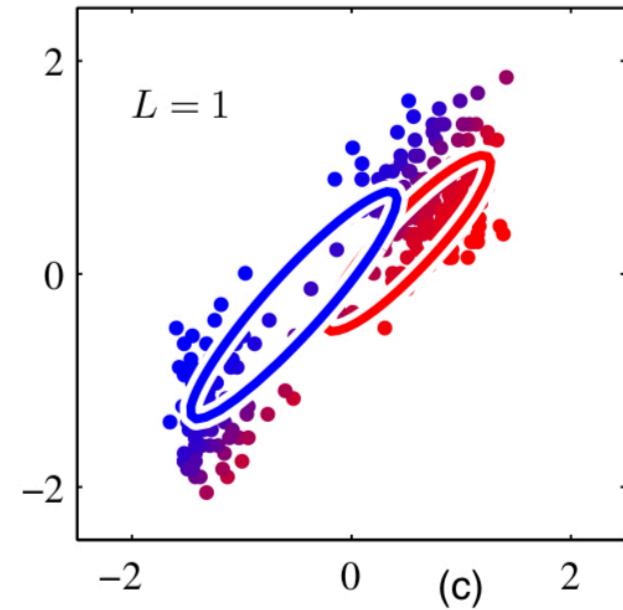
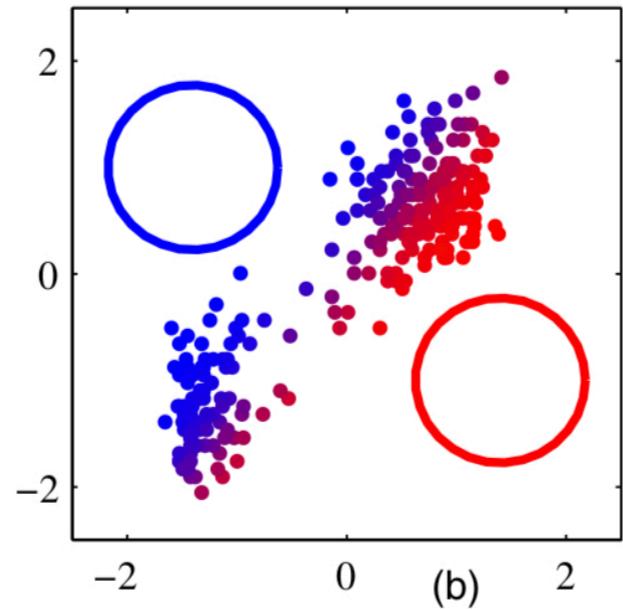
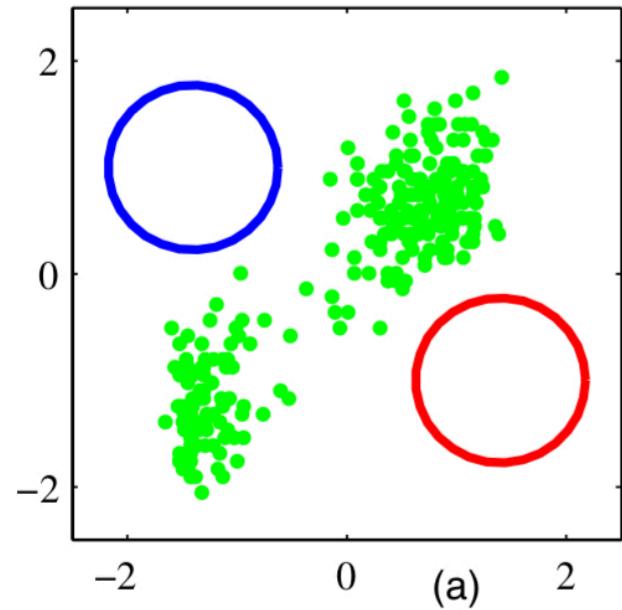
Source: *Pattern Recognition and Machine Learning*



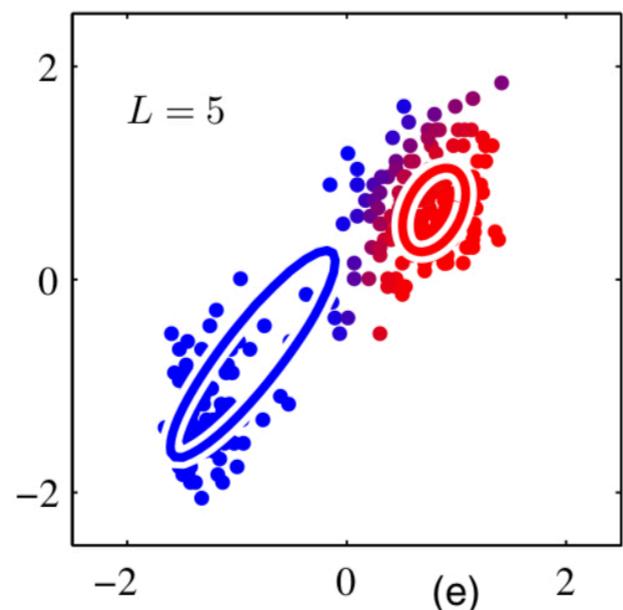
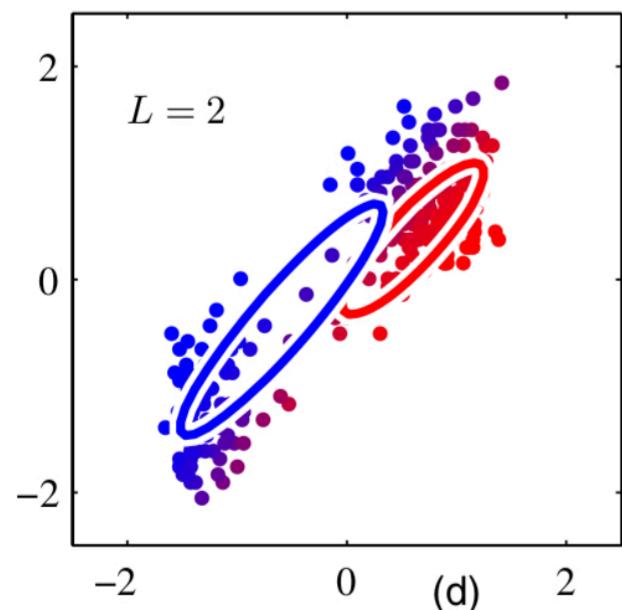
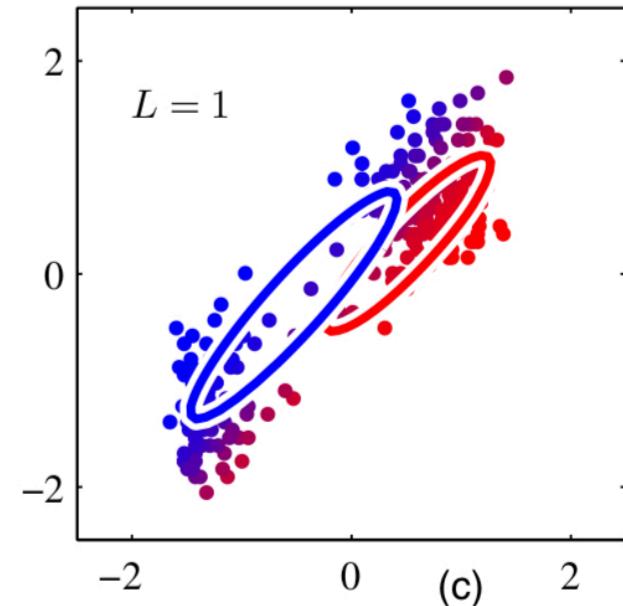
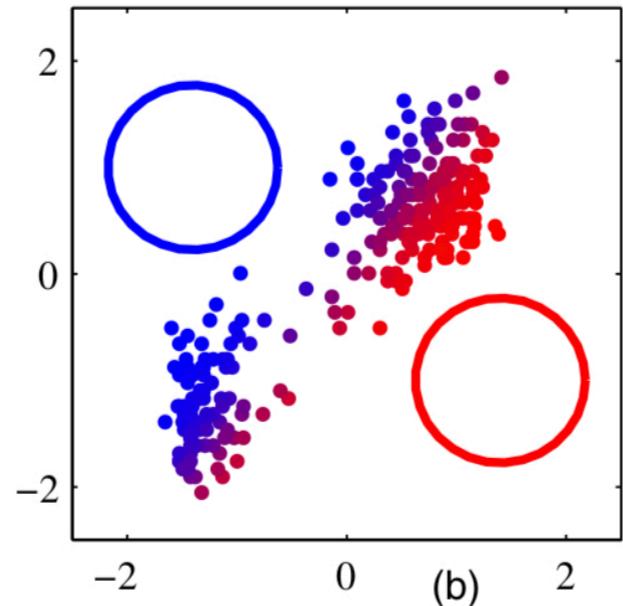
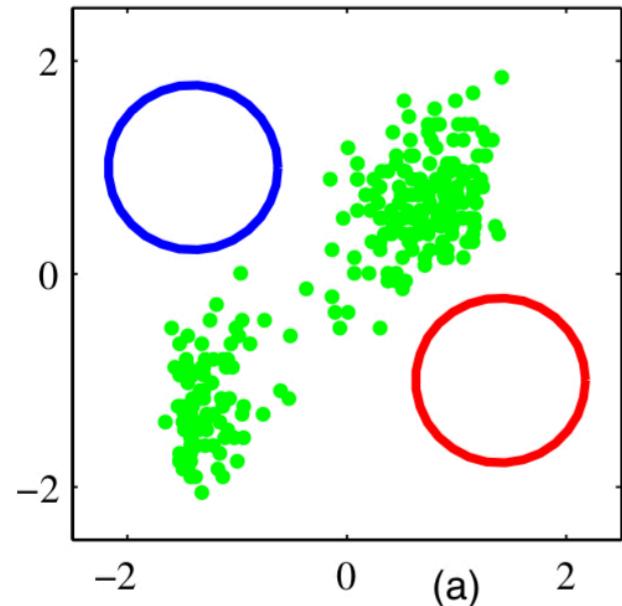
Source: *Pattern Recognition and Machine Learning*



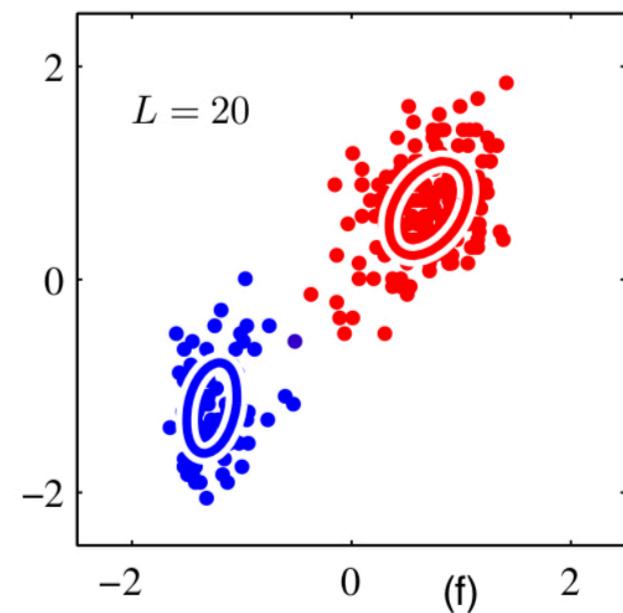
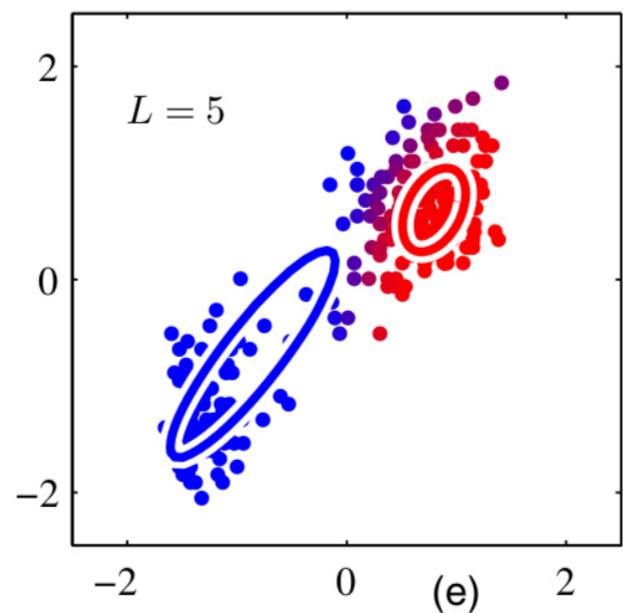
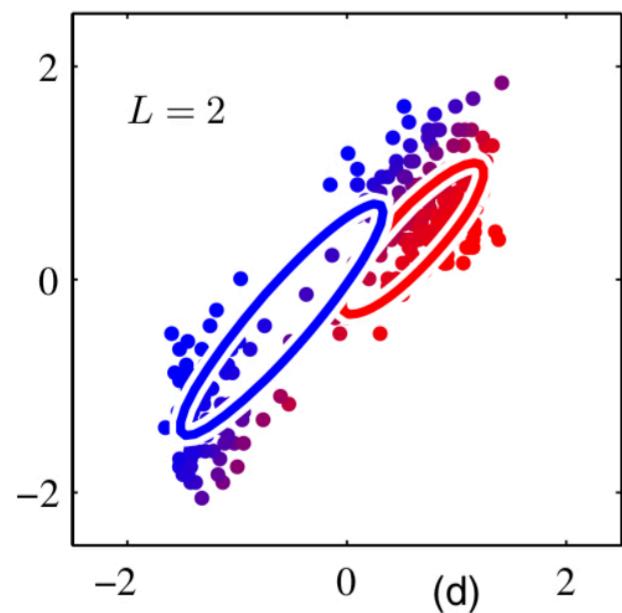
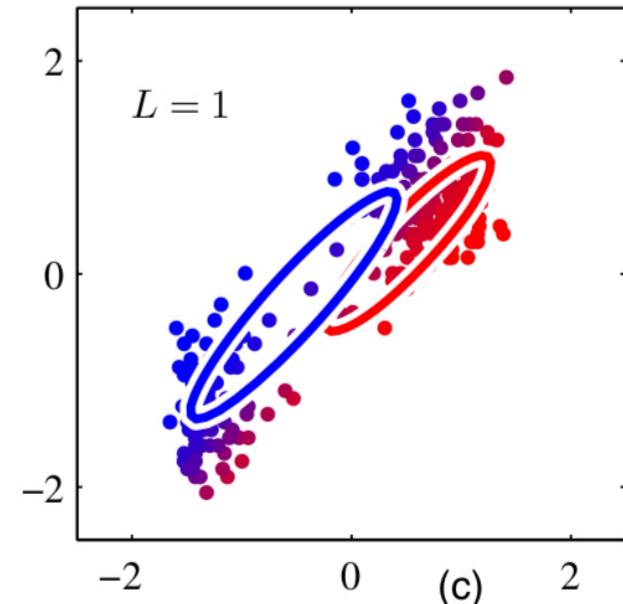
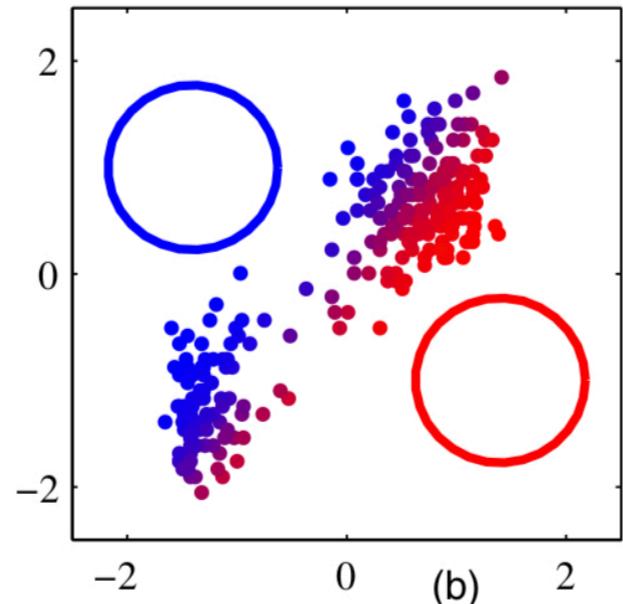
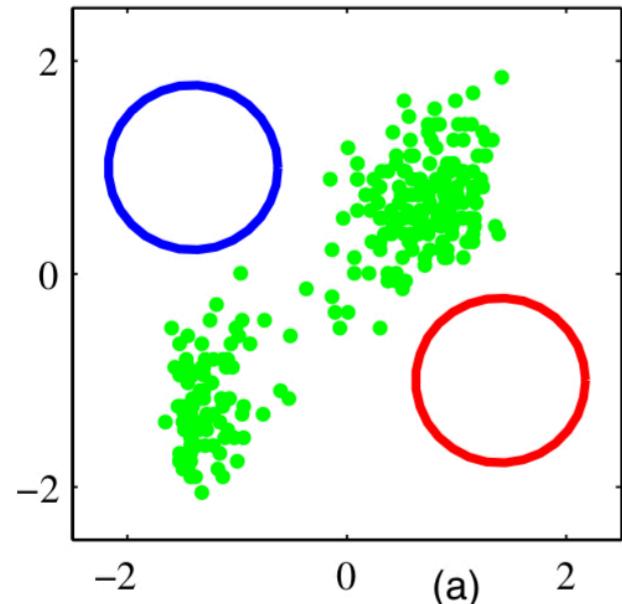
Source: *Pattern Recognition and Machine Learning*



Source: *Pattern Recognition and Machine Learning*



Source: *Pattern Recognition and Machine Learning*



Source: *Pattern Recognition and Machine Learning*

# Gaussian Mixture Models Algorithm

---

## E Step

$$w_{ik} = \frac{\omega_k N(x_i | \mu_k, \Sigma_k^2)}{\sum_{m=1}^K \omega_m N(x_i | \mu_m, \Sigma_m^2)}$$

# Gaussian Mixture Models Algorithm

---

## E Step

$$w_{ik} = \frac{\omega_k N(x_i | \mu_k, \Sigma_k^2)}{\sum_{m=1}^K \omega_m N(x_i | \mu_m, \Sigma_m^2)}$$

## M Step

$$N_k = \sum_{n=1}^N w_{ik}$$

# Gaussian Mixture Models Algorithm

---

## E Step

$$w_{ik} = \frac{\omega_k N(x_i | \mu_k, \Sigma_k^2)}{\sum_{m=1}^K \omega_m N(x_i | \mu_m, \Sigma_m^2)}$$

## M Step

$$N_k = \sum_{n=1}^N w_{ik}$$

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^N w_{ik} x_i$$

# Gaussian Mixture Models Algorithm

---

## E Step

$$w_{ik} = \frac{\omega_k N(x_i | \mu_k, \Sigma_k^2)}{\sum_{m=1}^K \omega_m N(x_i | \mu_m, \Sigma_m^2)}$$

## M Step

$$N_k = \sum_{n=1}^N w_{ik}$$

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^N w_{ik} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^N w_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

# Gaussian Mixture Models Algorithm

---

## E Step

$$w_{ik} = \frac{\omega_k N(x_i | \mu_k, \Sigma_k^2)}{\sum_{m=1}^K \omega_m N(x_i | \mu_m, \Sigma_m^2)}$$

## M Step

$$N_k = \sum_{n=1}^N w_{ik}$$

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^N w_{ik} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^N w_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

$$\omega_k = \frac{N_k}{N}$$

# Gaussian Mixture Models Algorithm

---

Repeat till  
convergence

## E Step

$$w_{ik} = \frac{\omega_k N(x_i | \mu_k, \Sigma_k^2)}{\sum_{m=1}^K \omega_m N(x_i | \mu_m, \Sigma_m^2)}$$

## M Step

$$N_k = \sum_{n=1}^N w_{ik}$$

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^N w_{ik} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^N w_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

$$\omega_k = \frac{N_k}{N}$$

# Why/Why not to use GMMs?

---

## **Advantages:**

- Mixed membership based on probability,
- More flexible in terms of covariance.

## **Disdvantages:**

- Optimization is not so trivial...
- May not work for datasets that doesn't have “hidden” Gaussian distribution.

Let's use it in practice! 

---

[TODO] LINK TO GOOGLE COLAB

Questions? 😊

# Resources

---

- “Pattern Recognition and Machine Learning”, Christopher M. Bishop,
- Credit Card Fraud Detection - [Kaggle Dataset](#),
- Gaussian Mixture Models - [Scikit-Learn User Guide](#),
- Gaussian Mixture Models - [Scikit-Learn Documentation](#),
- Andrew Ng - [Anomaly Detection Problem](#),
- [https://vas3k.com/blog/machine learning/](https://vas3k.com/blog/machine_learning/)
- <http://tinyheero.github.io/2015/10/13/mixture-model.html>
- <http://yulearning.blogspot.com/2014/11/einsteins-most-famous-equation-is-emc2.html>
- [https://angusturner.github.io/generative\\_models/2017/11/03/pytorch-gaussian-mixture-model.html](https://angusturner.github.io/generative_models/2017/11/03/pytorch-gaussian-mixture-model.html)

# Thanks

---

Jakub Powierza

[jakub.powierza@icloud.com](mailto:jakub.powierza@icloud.com)  
<https://linkedin.com/in/jakub-powierza/>