

# APS1070

Foundations of Data Analytics and  
Machine Learning

Winter 2022

## Week 4:

- *Clustering*
- *Probability Theory*
- *Summary Statistics*
- *Gaussian Distribution*
- *Performance Metrics*



# Slide Attribution

These slides contain materials from various sources. Special thanks to the following authors:

- Sinisa Colic
- William Fleshman
- Scott Sanner
- Ali Hadi Zadeh
- Jason Riordon

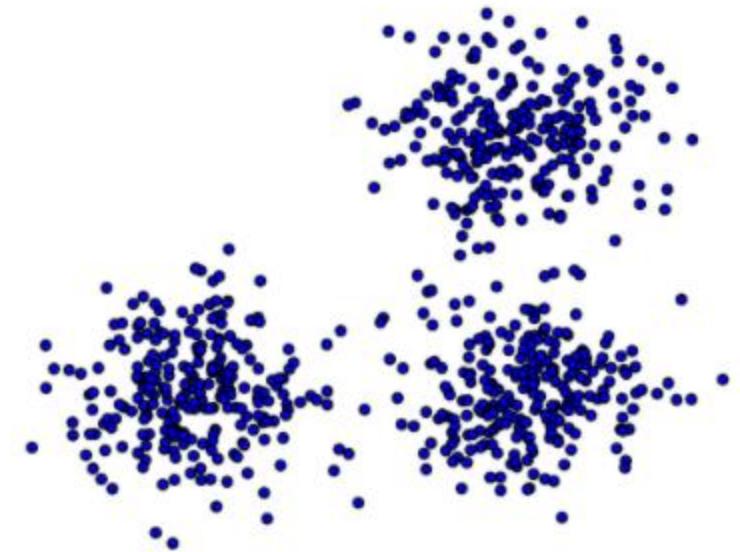
# Part 1

## K-Means Clustering

### (Unsupervised Learning)

# Clustering

- Clustering algorithms group samples/instances based on similarities in features
  - **Input:** set of samples/instances described by features
  - **Output:** assigned cluster (group) for each sample/instance
- Clustering are **unsupervised techniques** and do not have access to the sample/instance labels



# Clustering Strategies

- **k-Means Clustering**

- Assigns each point to the nearest cluster center

- **Agglomerative clustering**

- Assumes each point is a cluster and iteratively merges the closest clusters

# K-Means

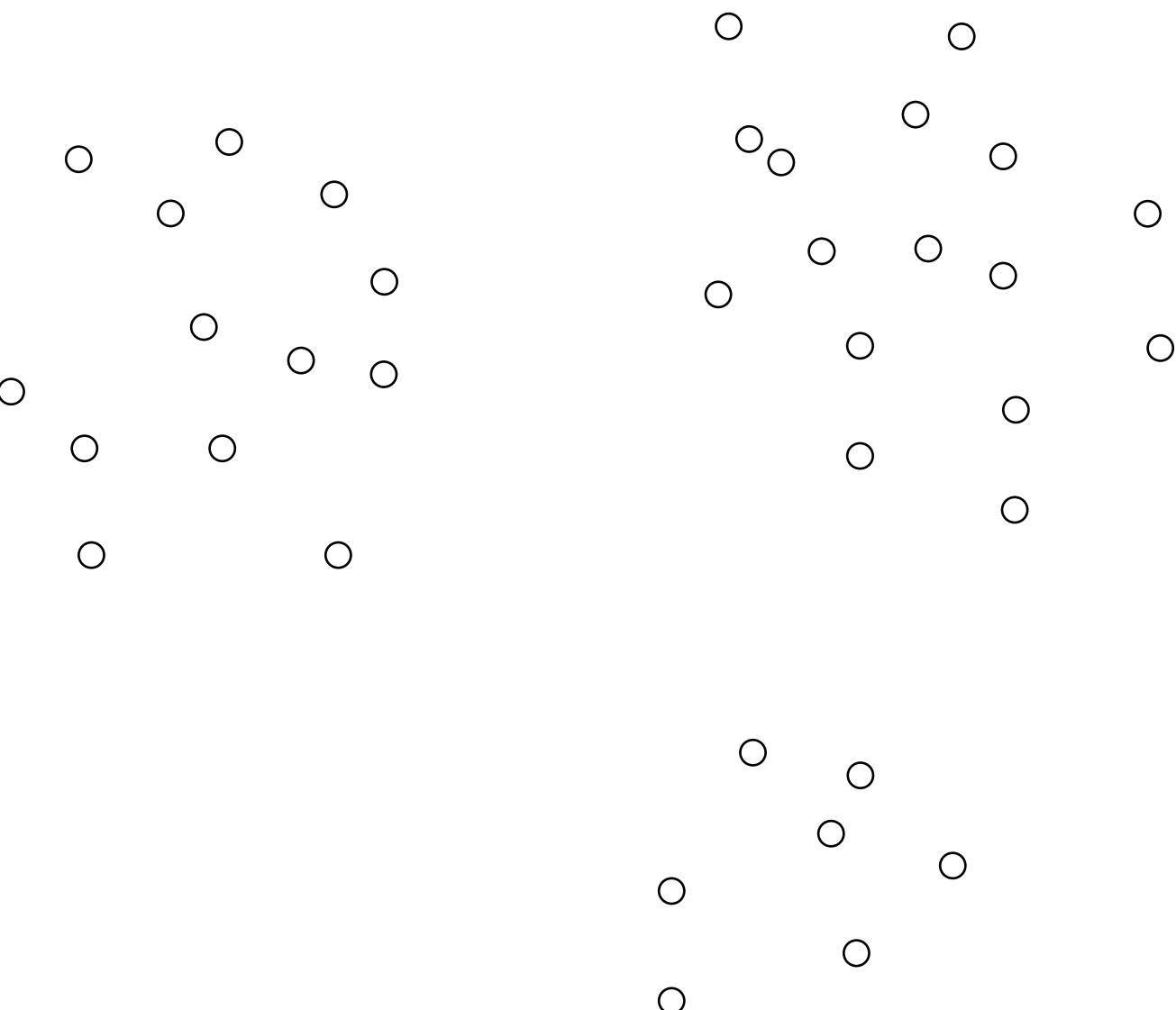
- Most well-known clustering method
- Distance-based **unsupervised** learning algorithm,  
NOT to be confused with k-NN.

- **Algorithm:**
  1. Assign each sample/instance to its closest mean
  2. Update the means based on the assignment
  3. Repeat until convergence

- **Requires:**
  - Selection of the number of clusters ‘k’ (hyperparameter)
  - Centre of each cluster is randomly initialized at the start

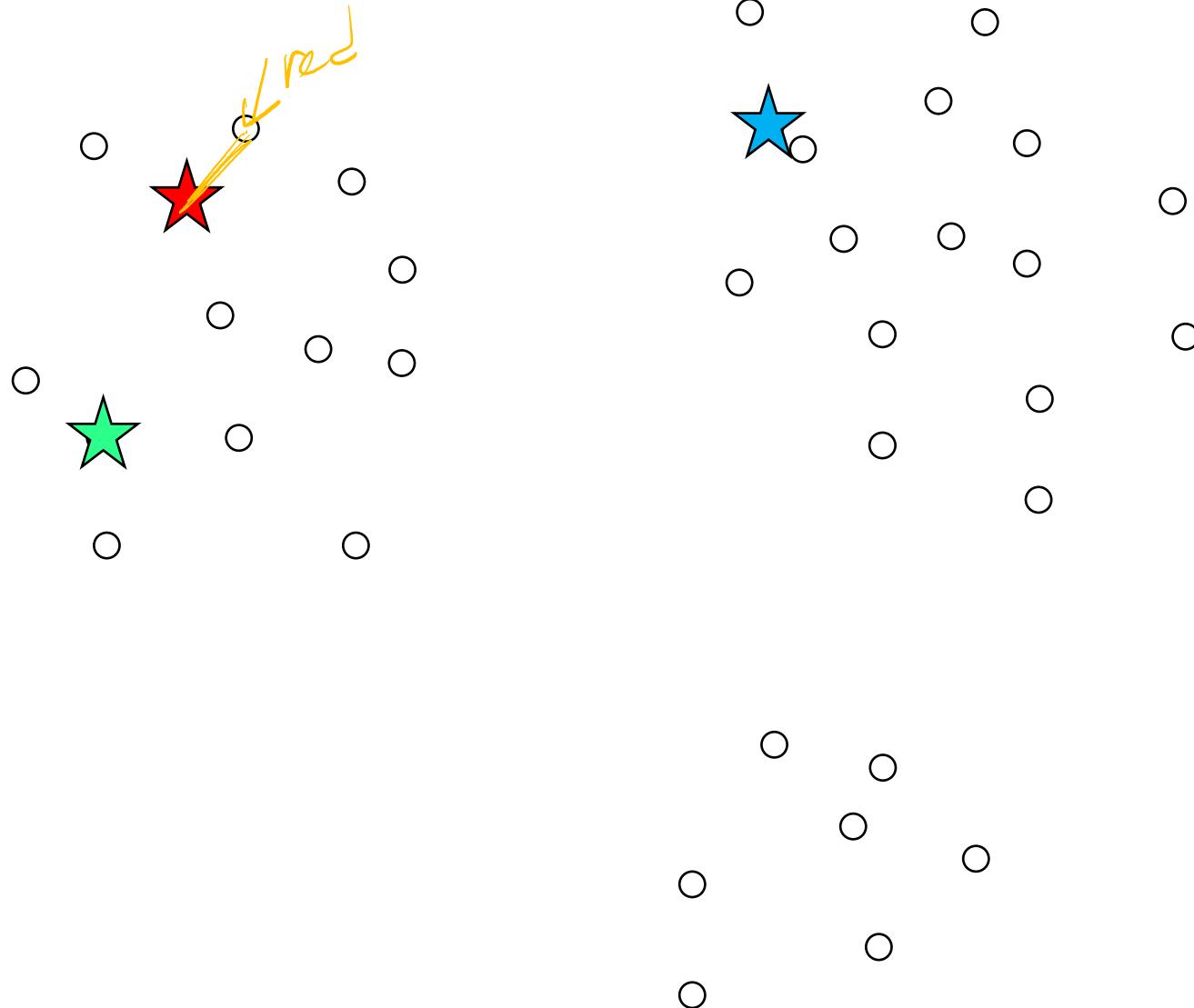
# K-Means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



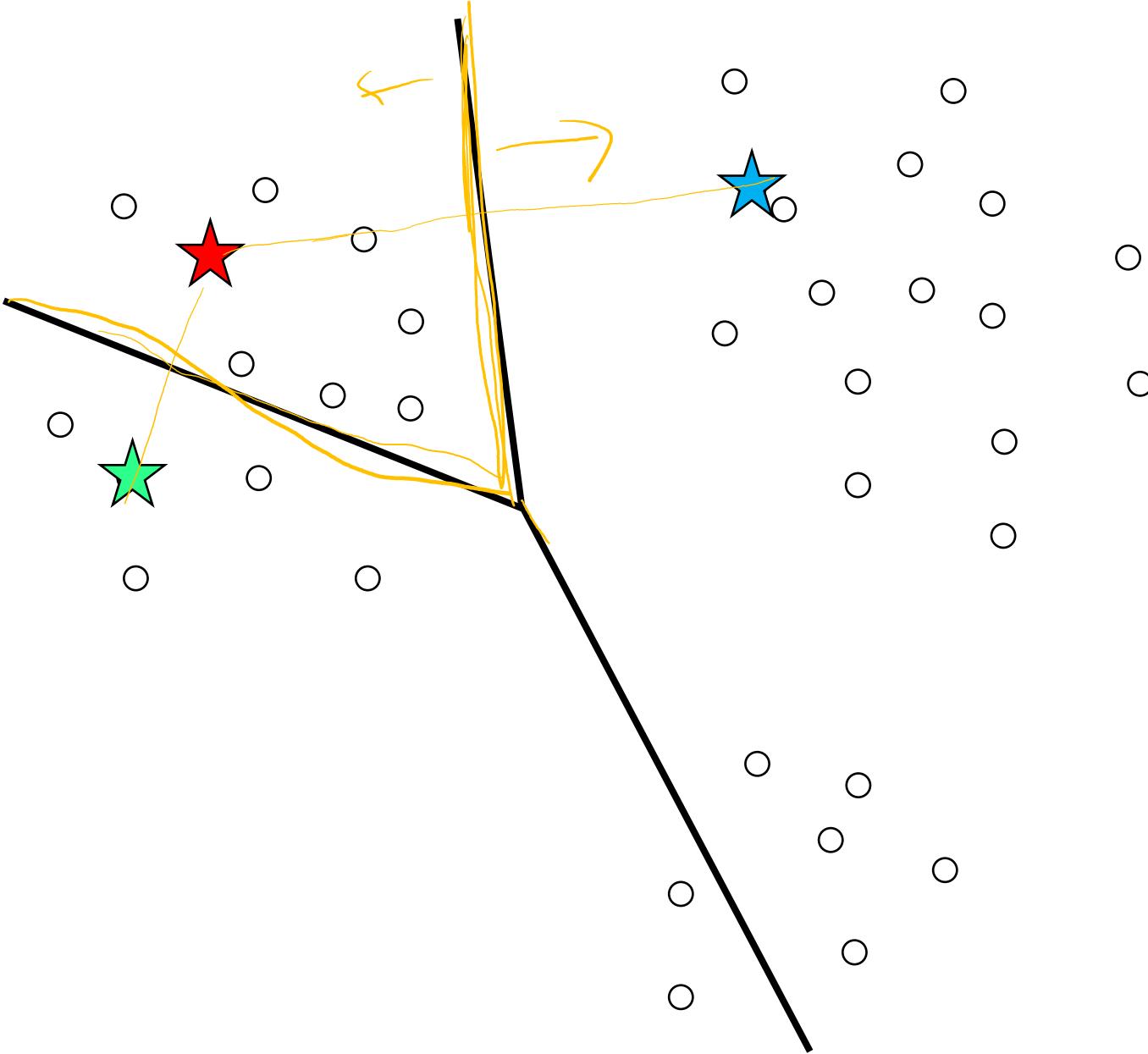
# K-Means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



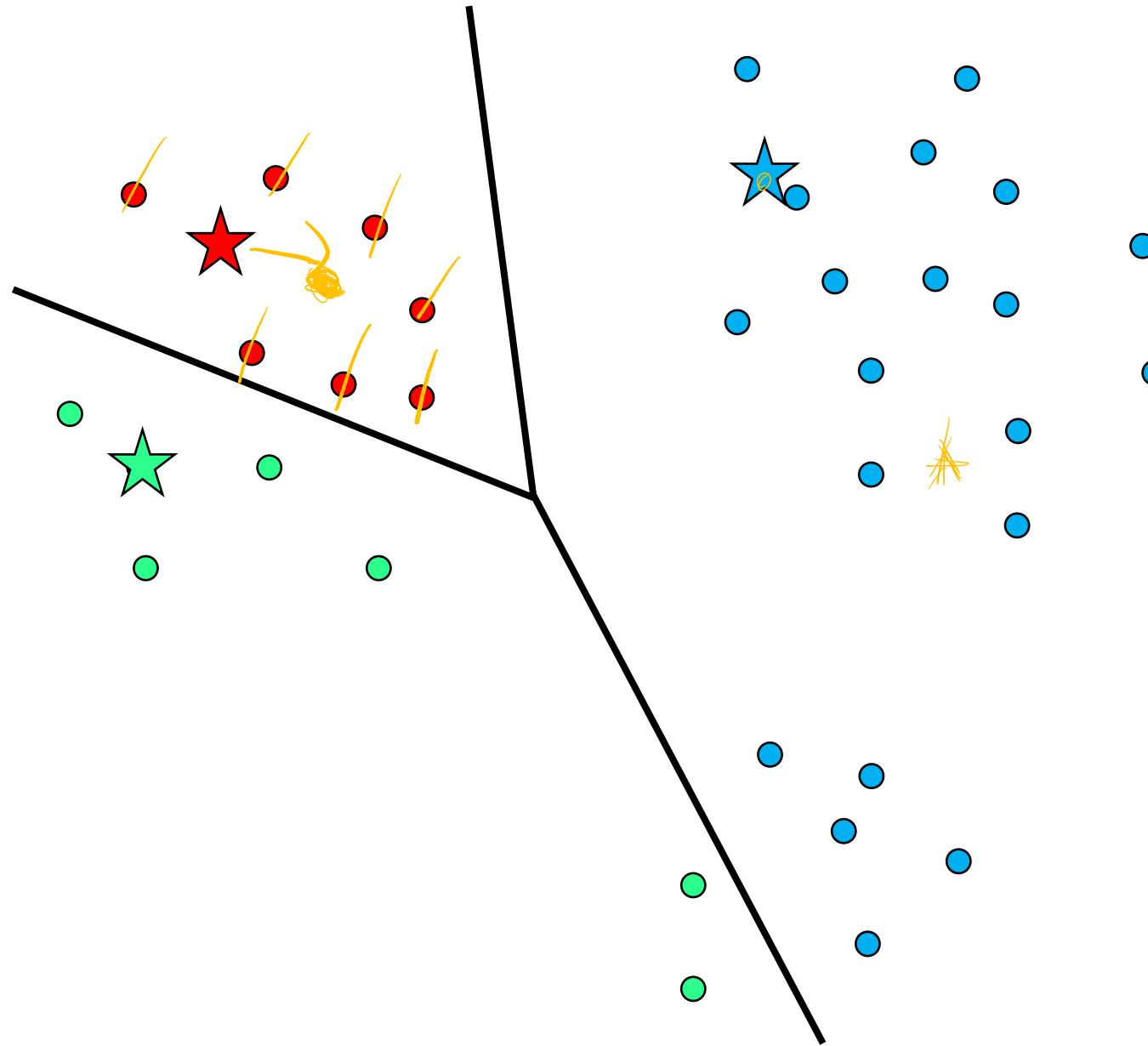
# K-Means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



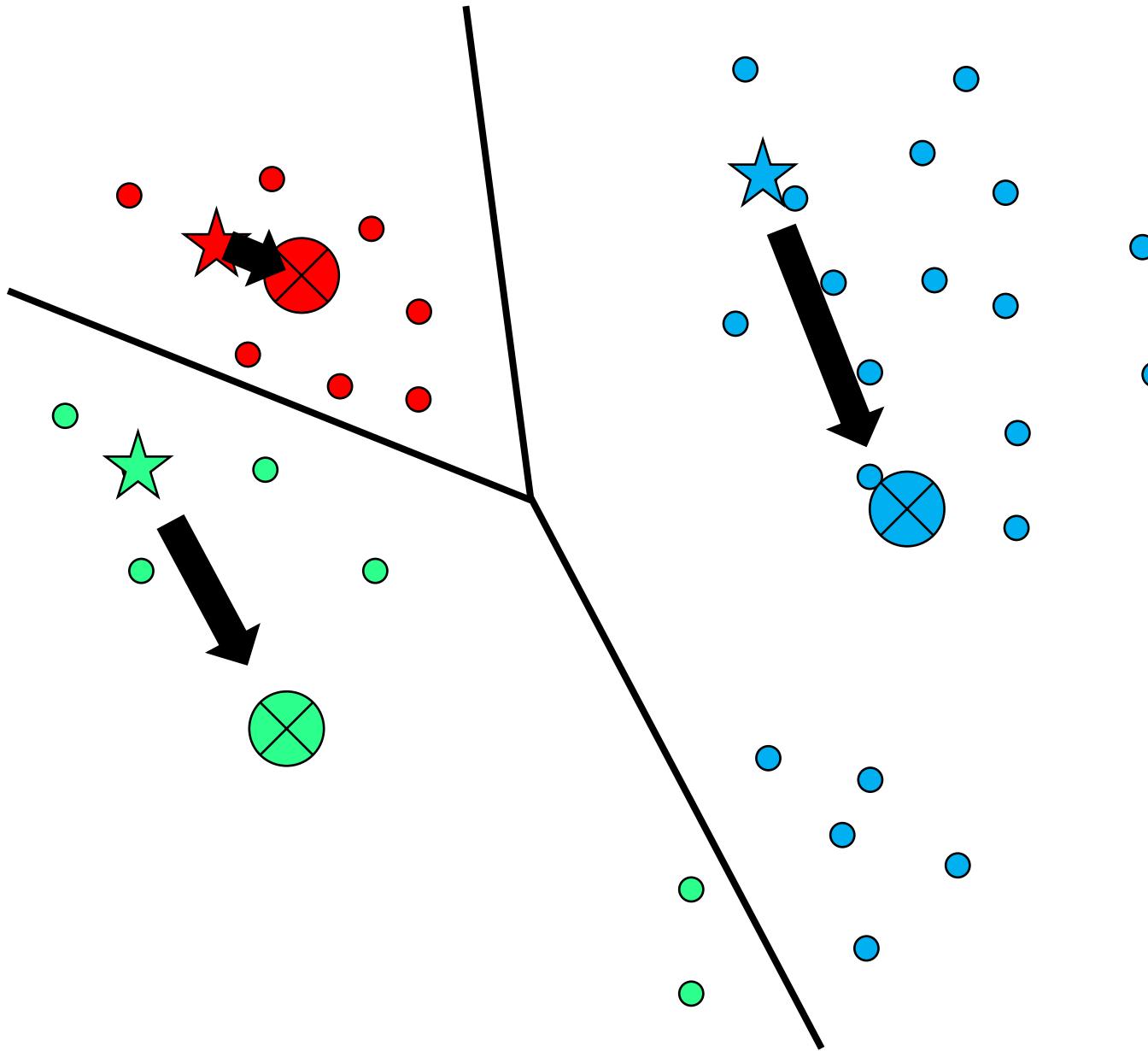
# K-Means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



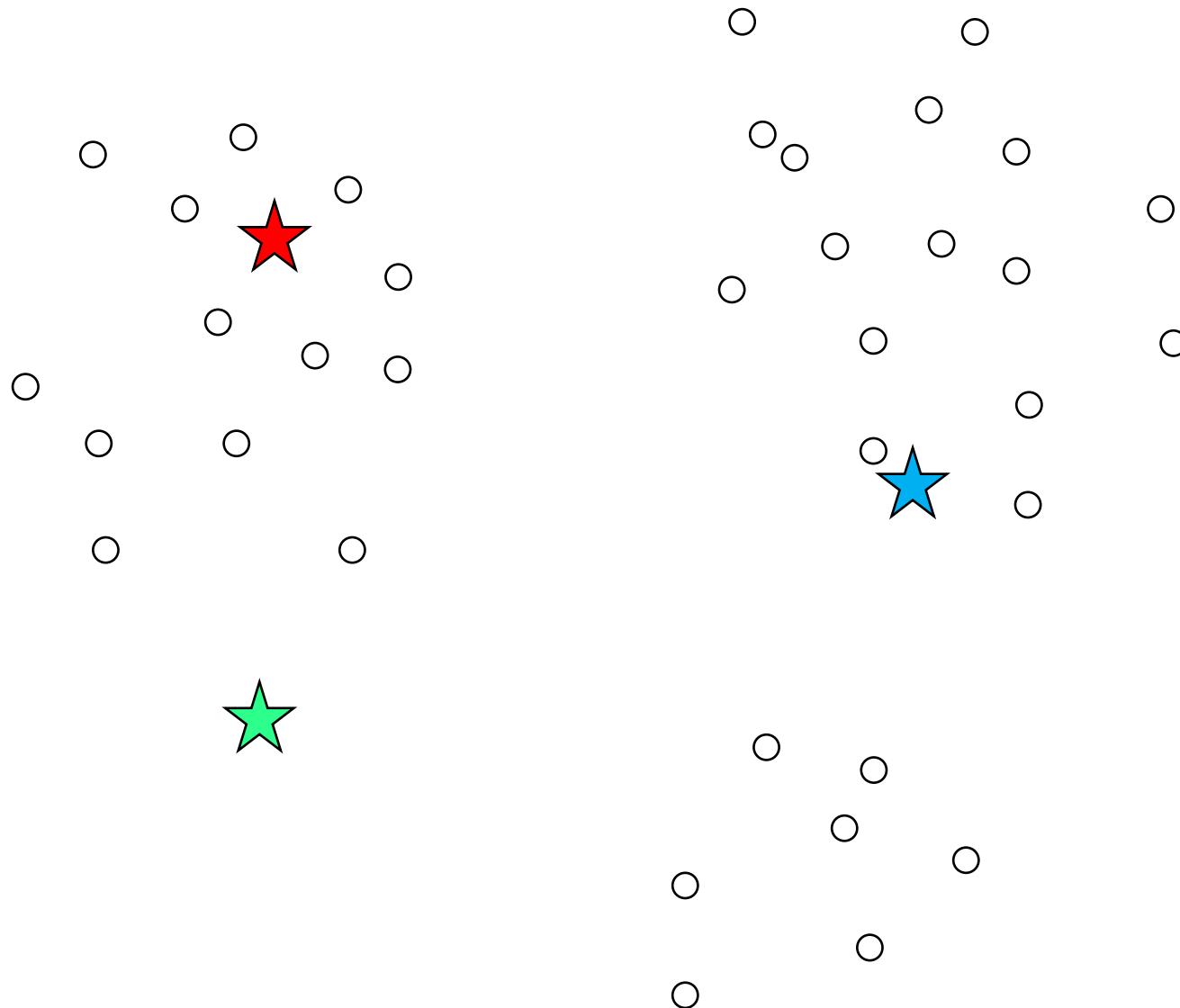
# K-Means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



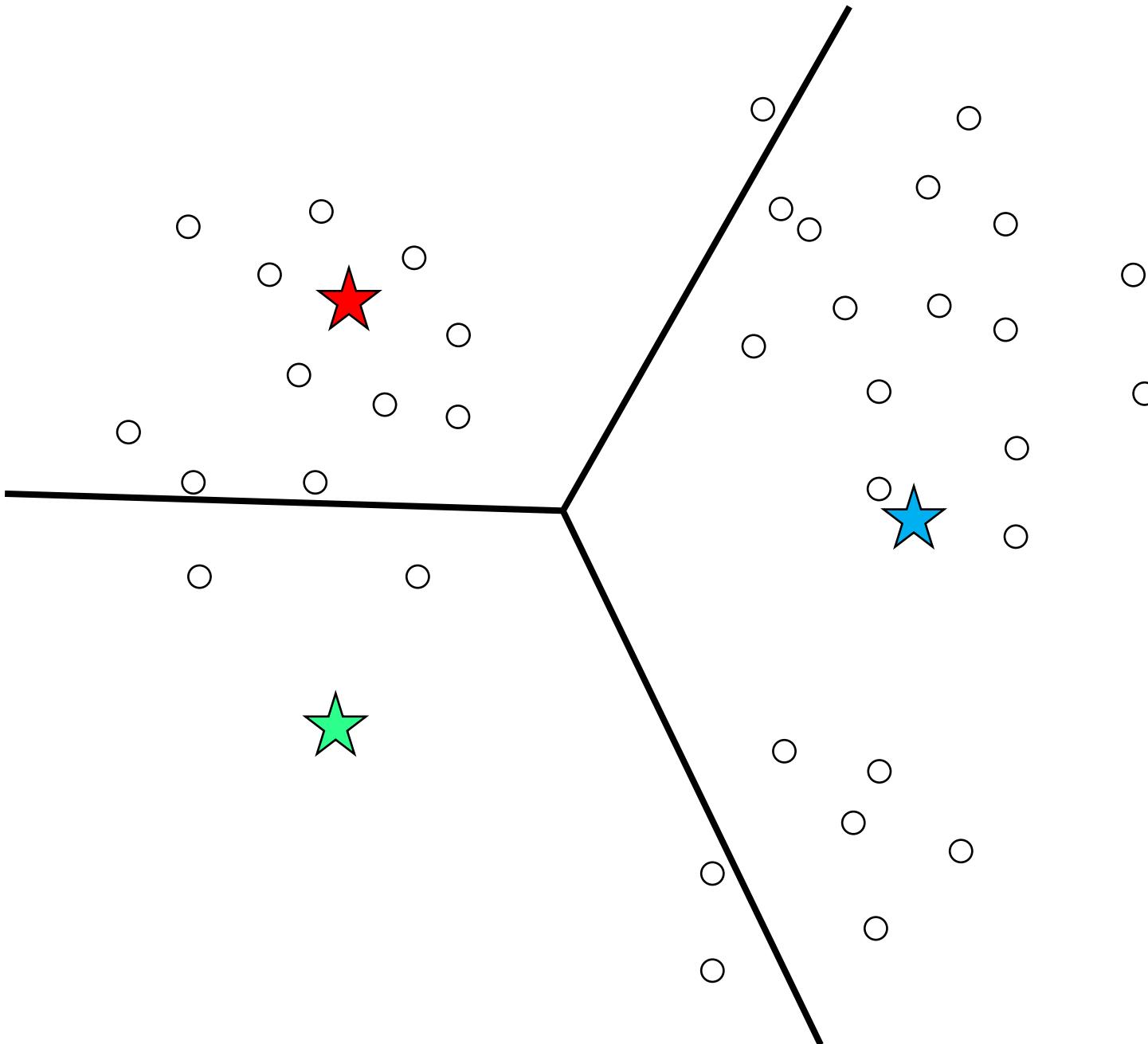
# K-means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



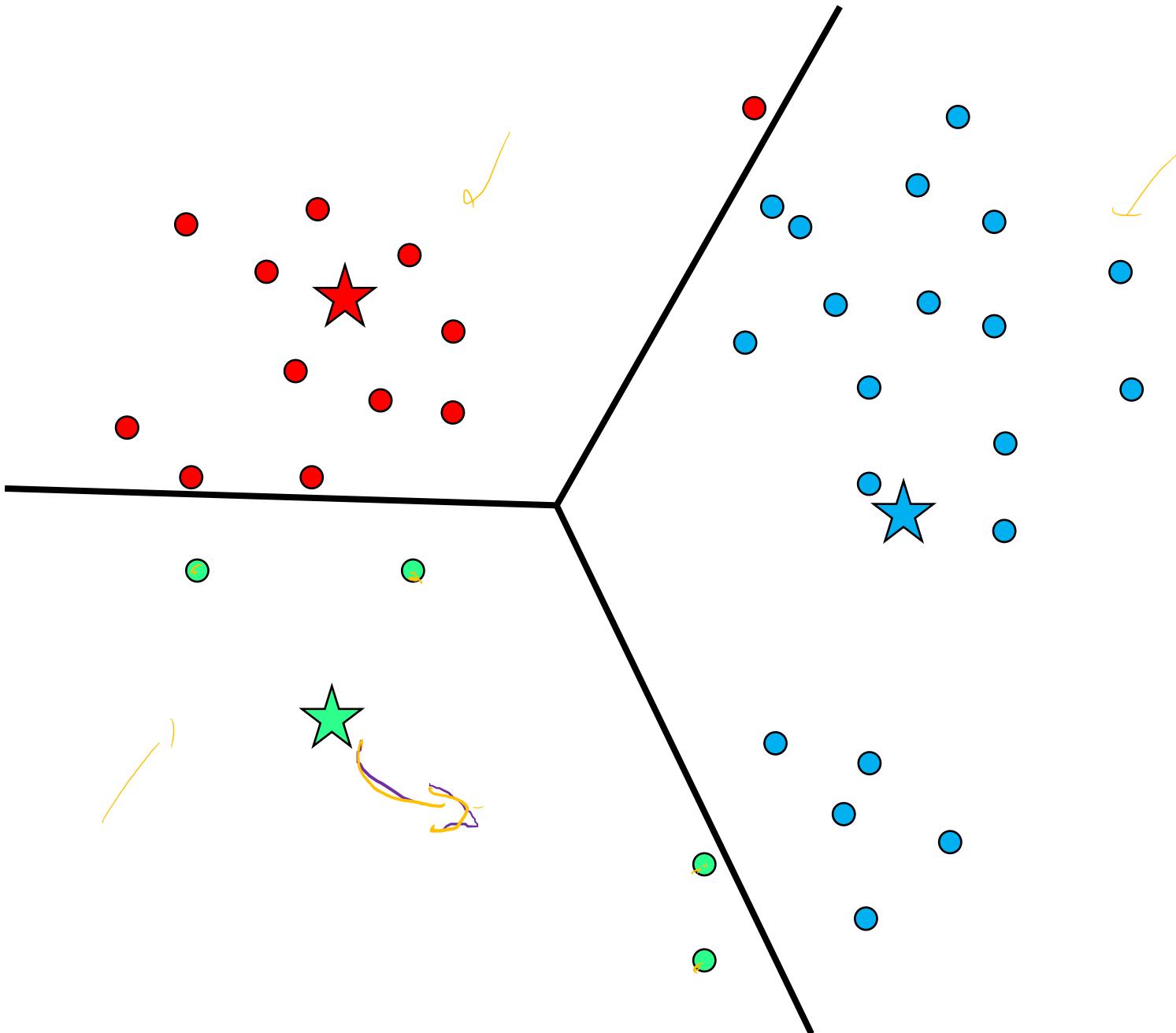
# K-means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



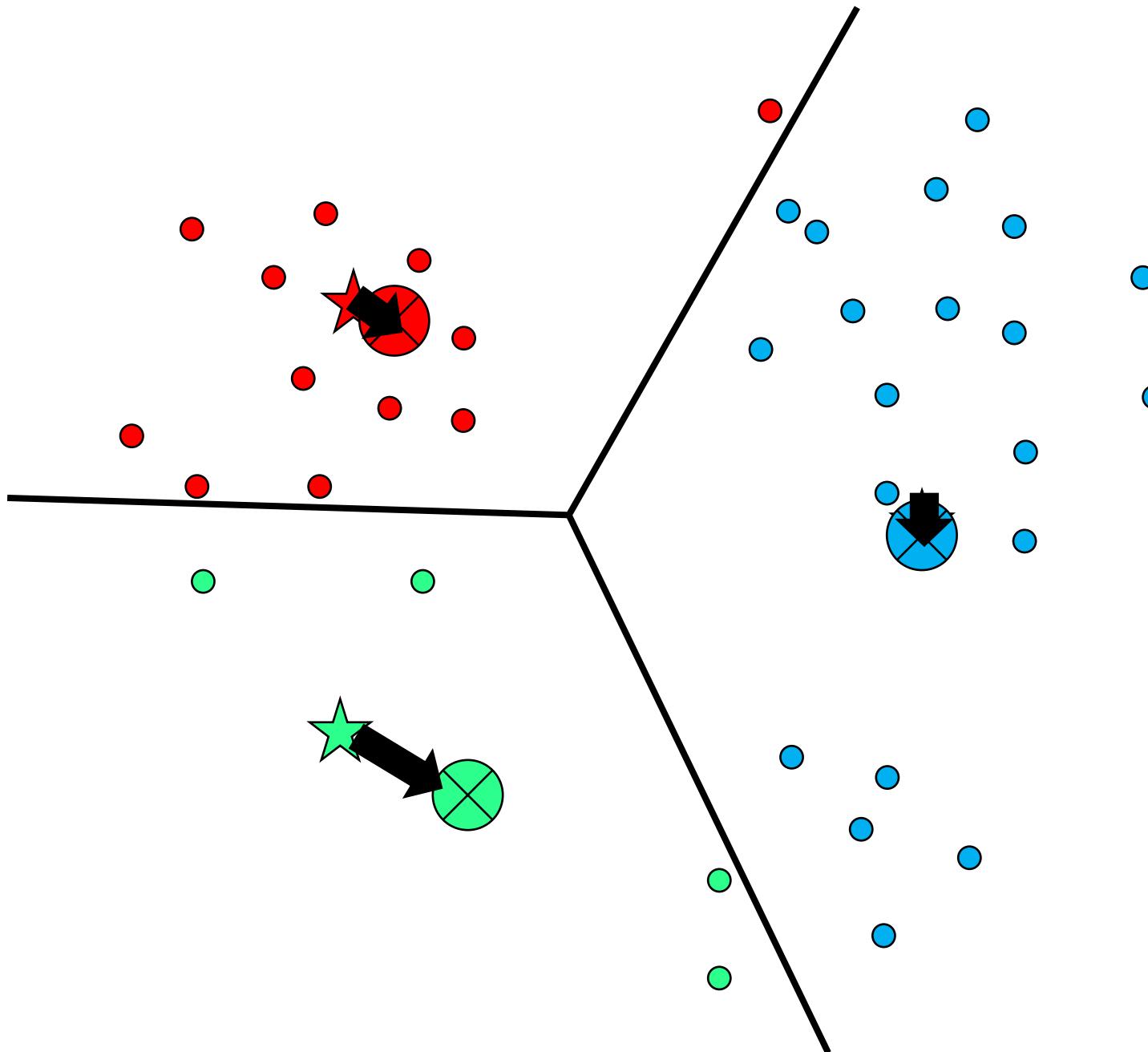
# K-means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



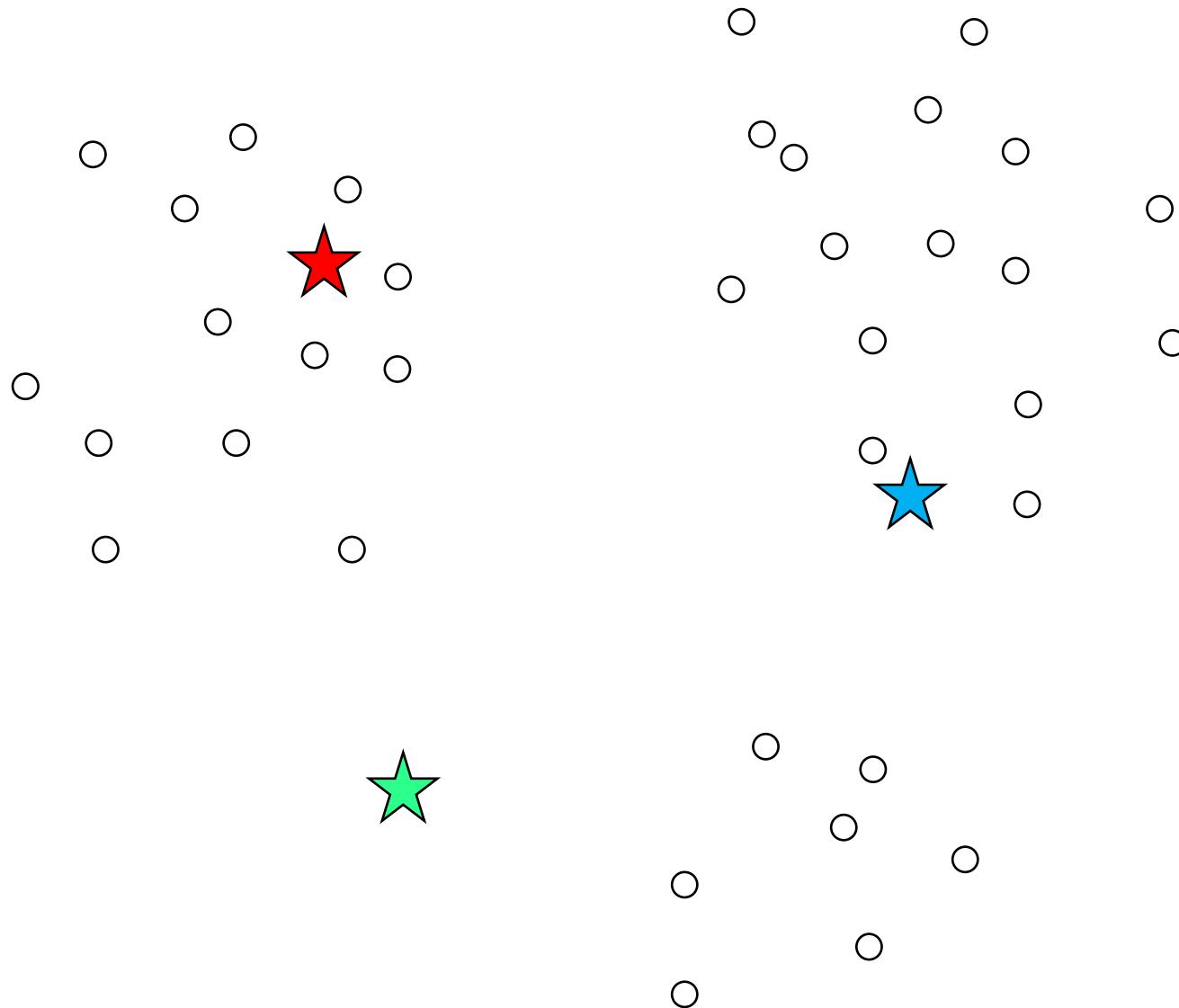
# K-means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



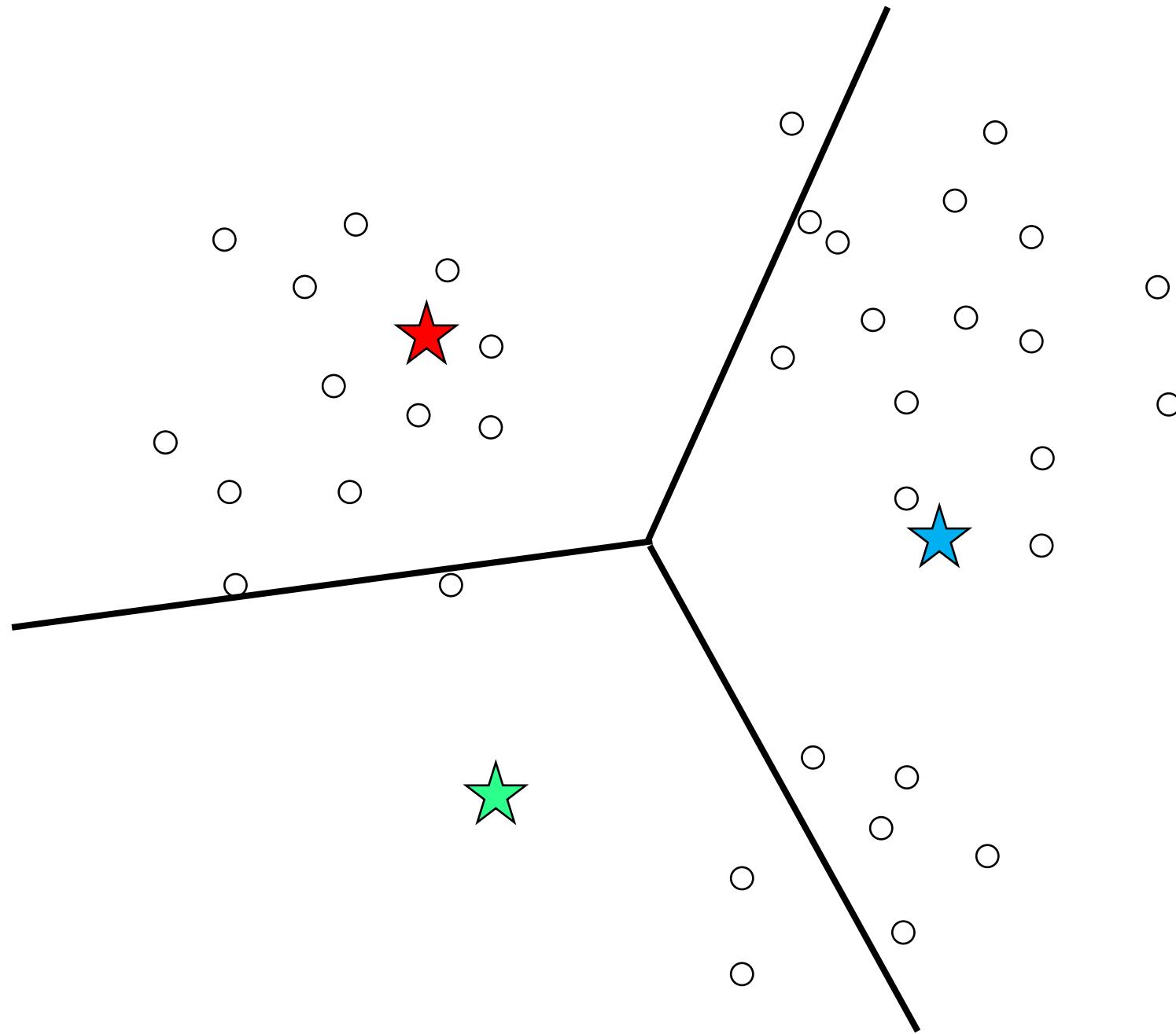
# K-means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



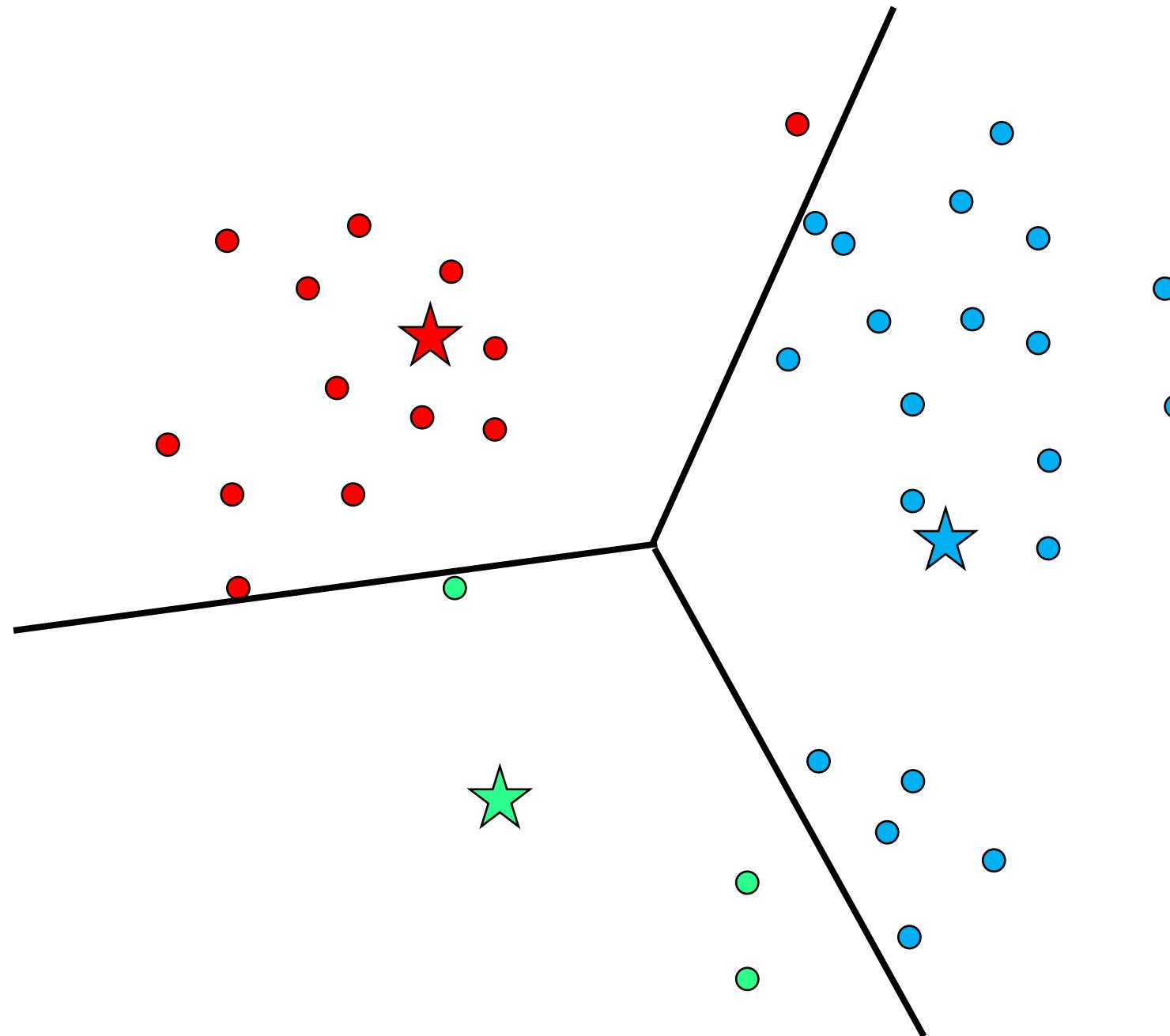
# K-means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



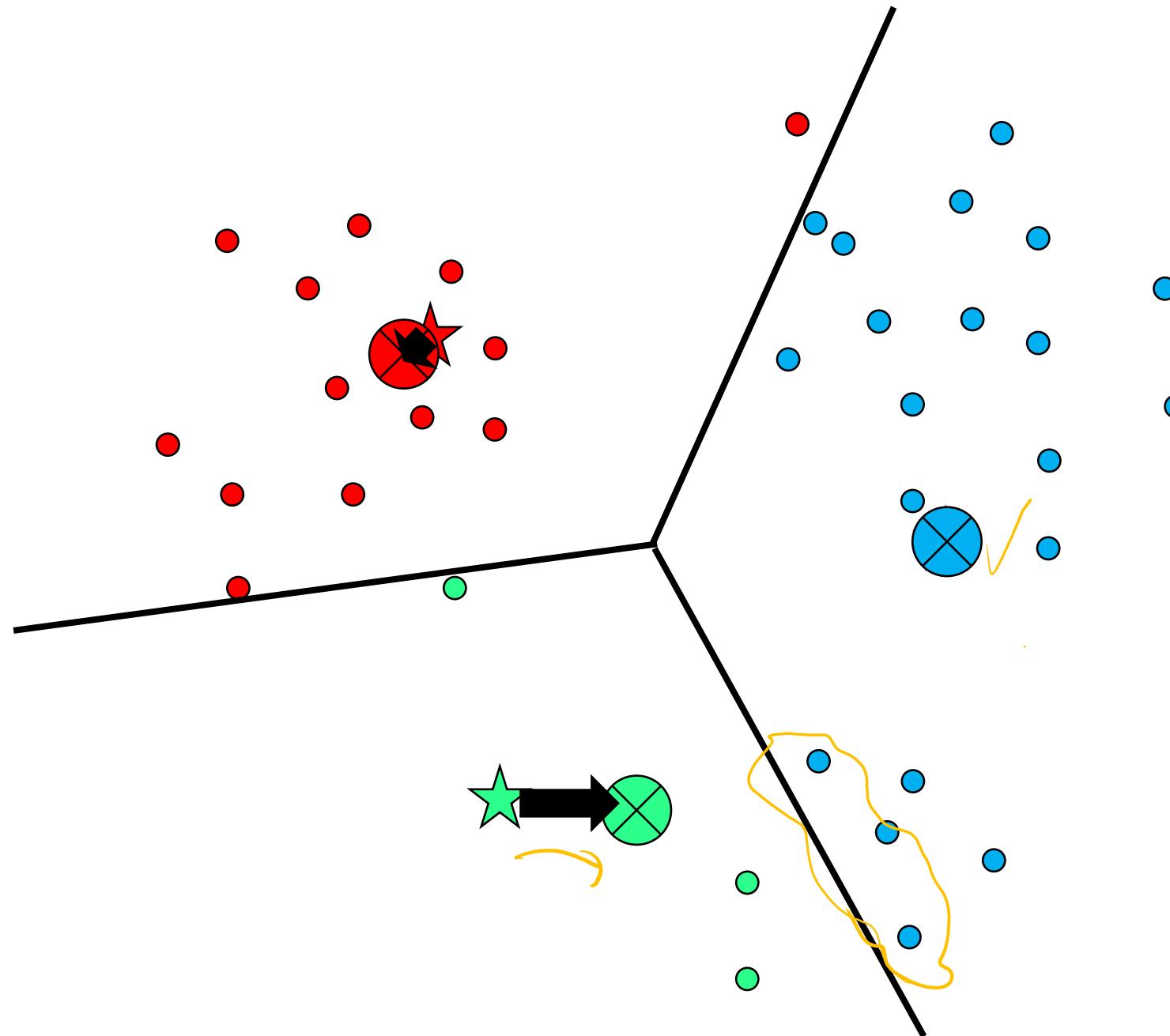
# K-means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



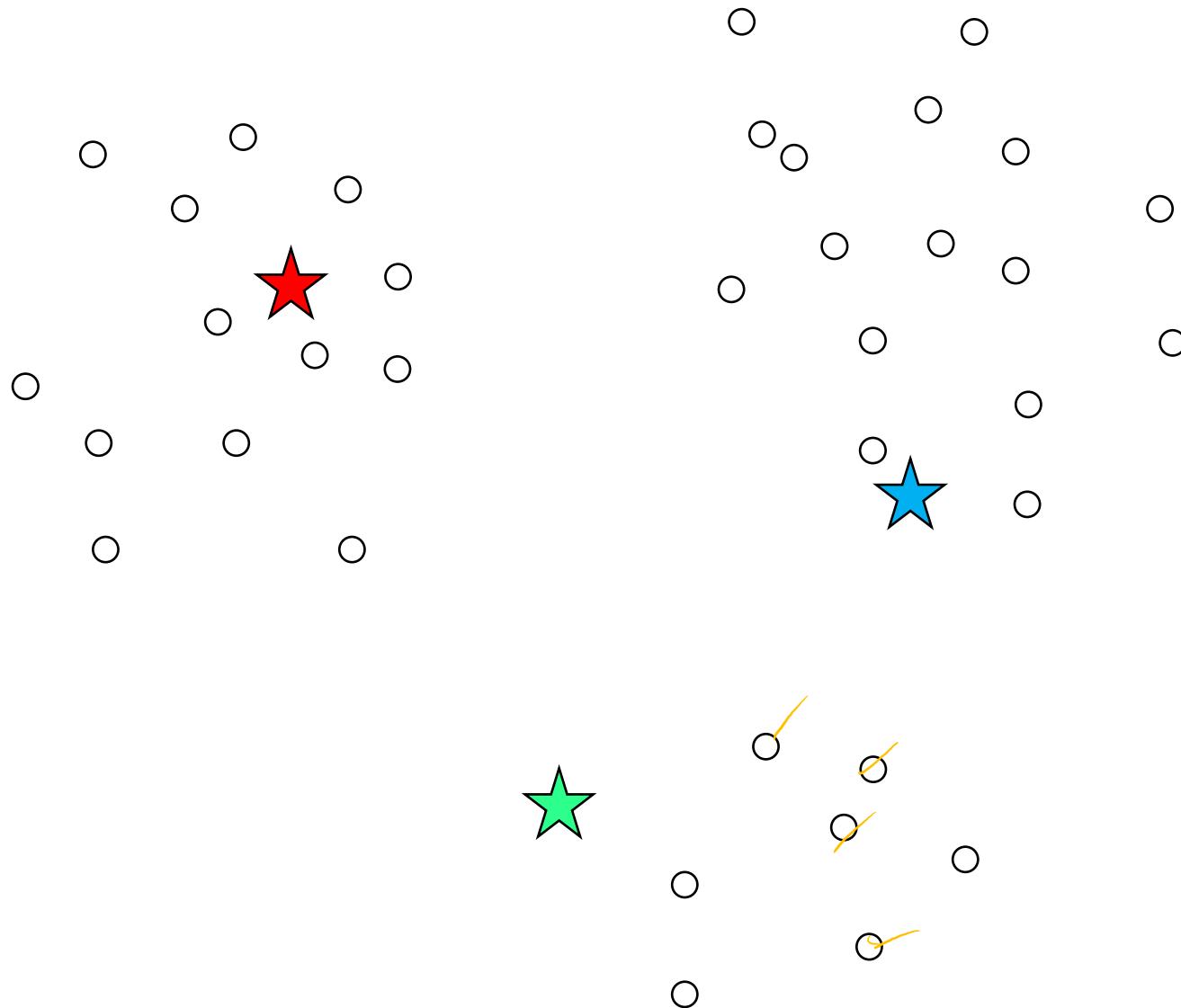
# K-means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



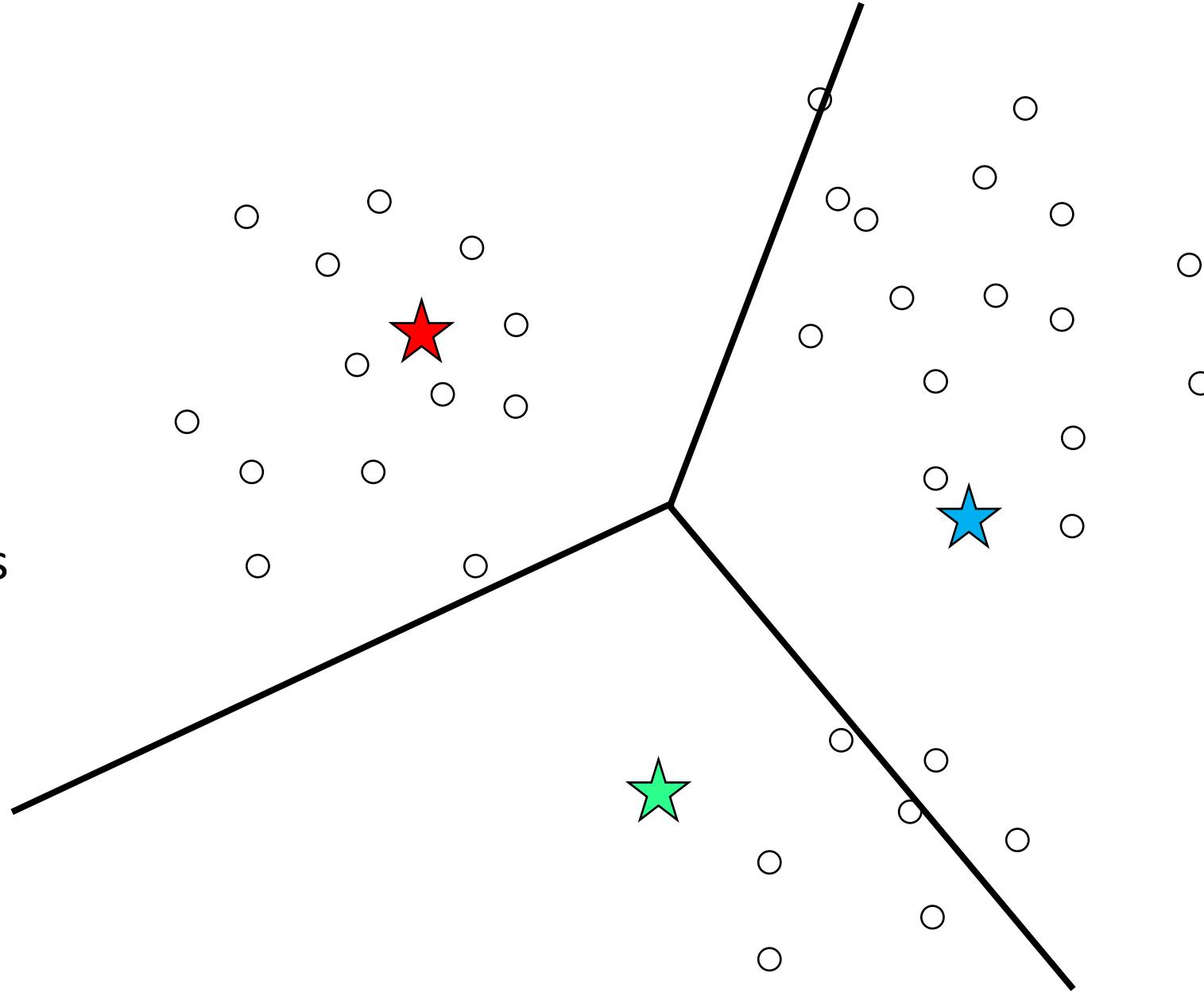
# K-means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



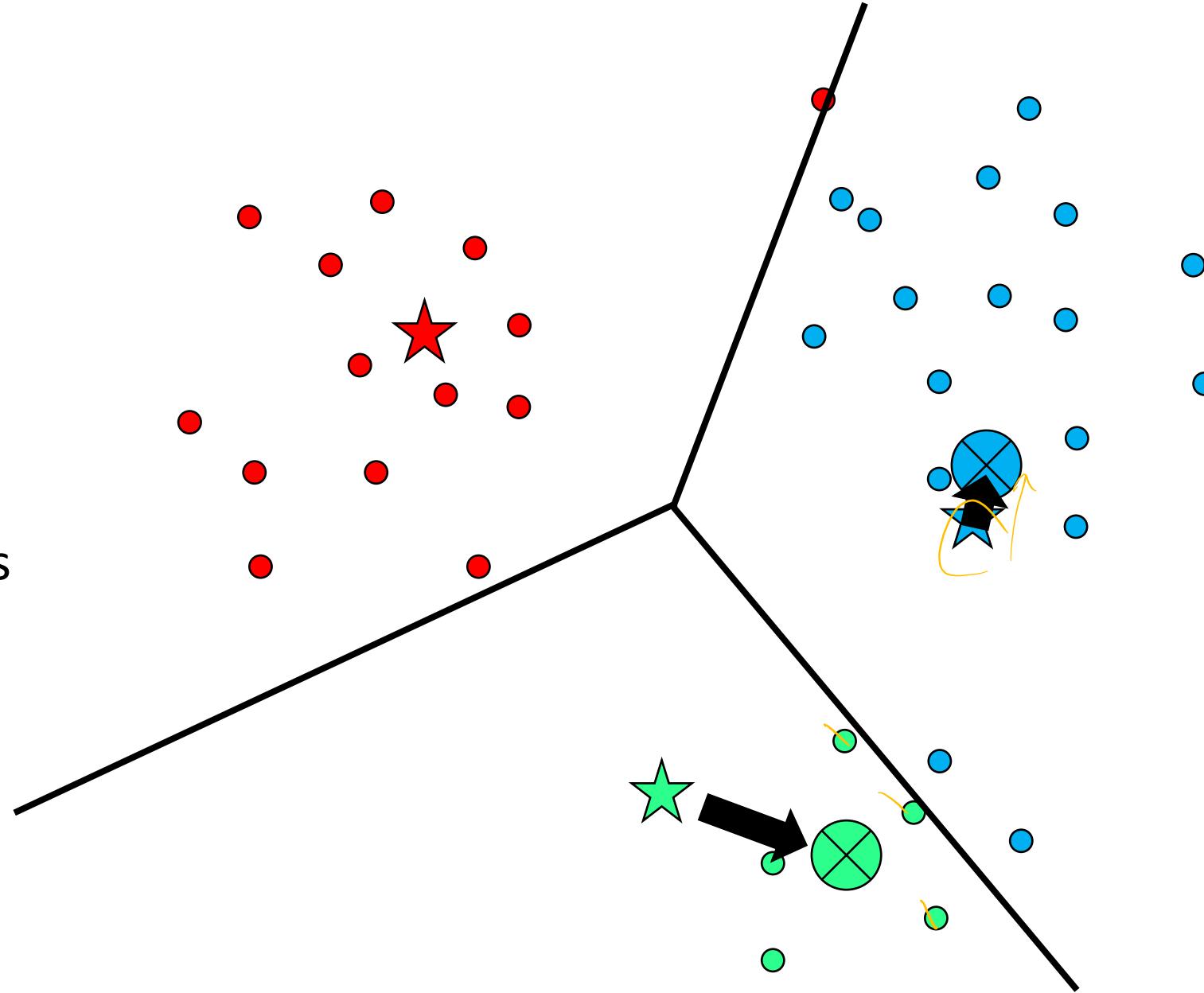
# K-means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



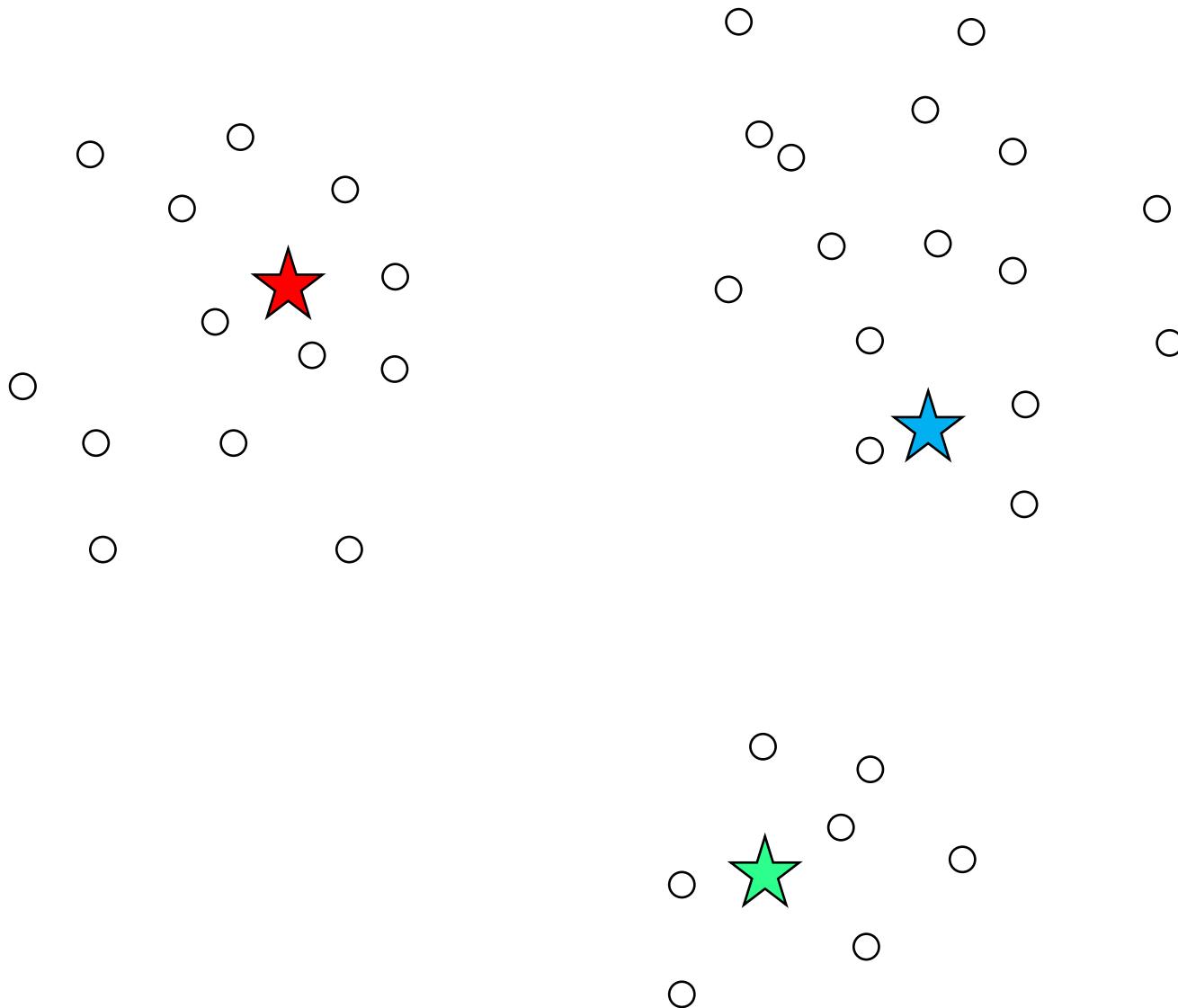
# K-means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



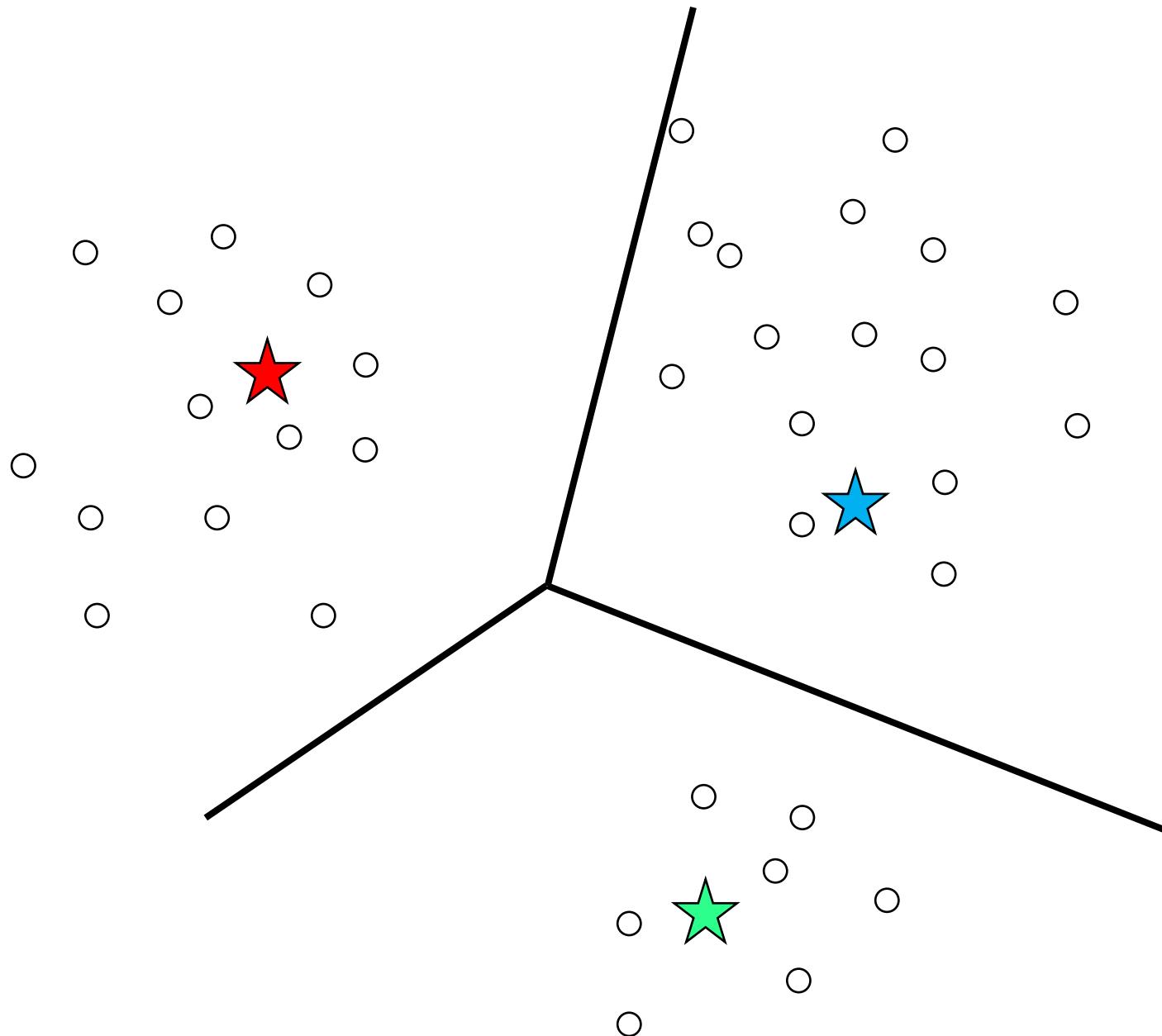
# K-means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



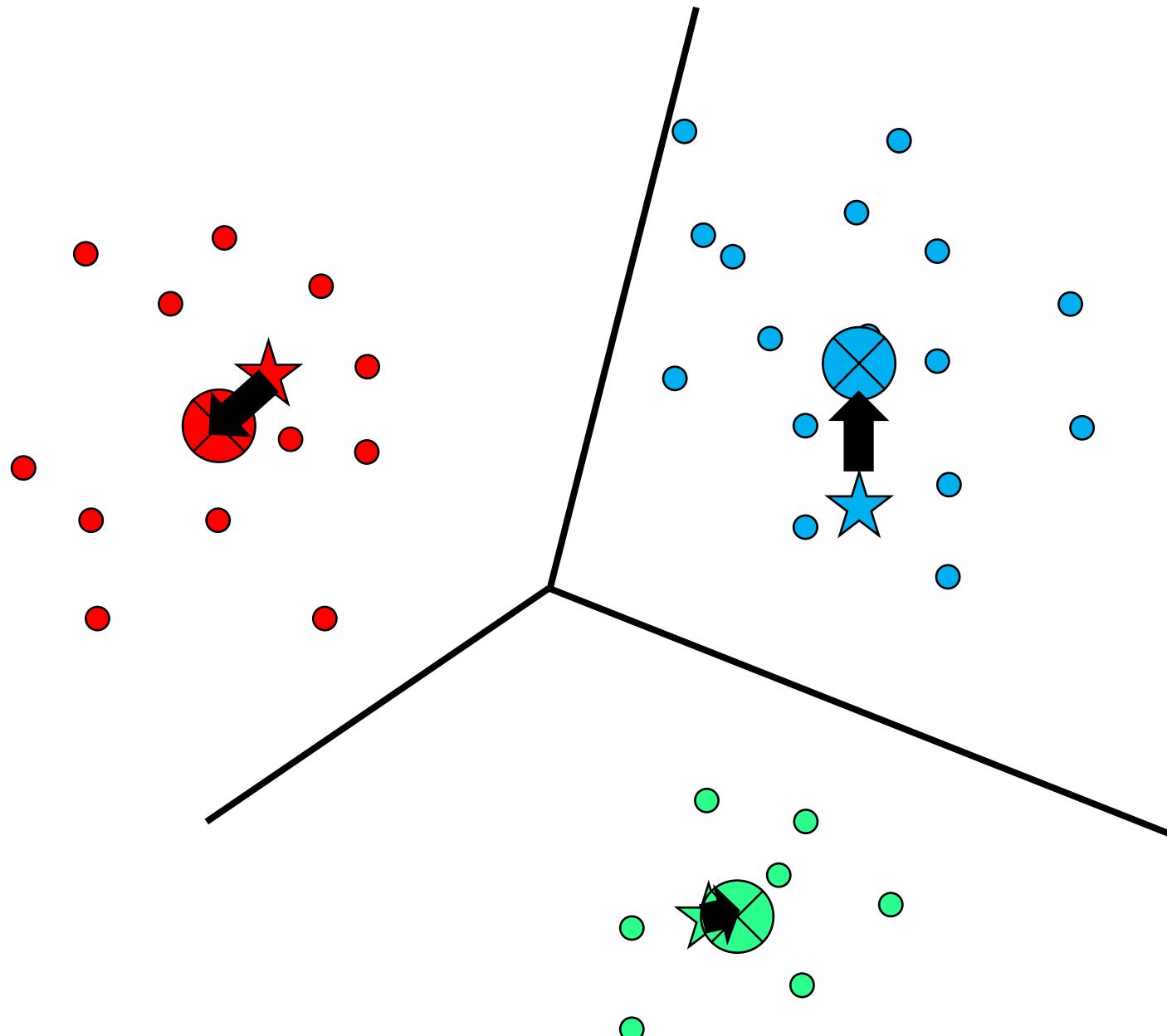
# K-means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



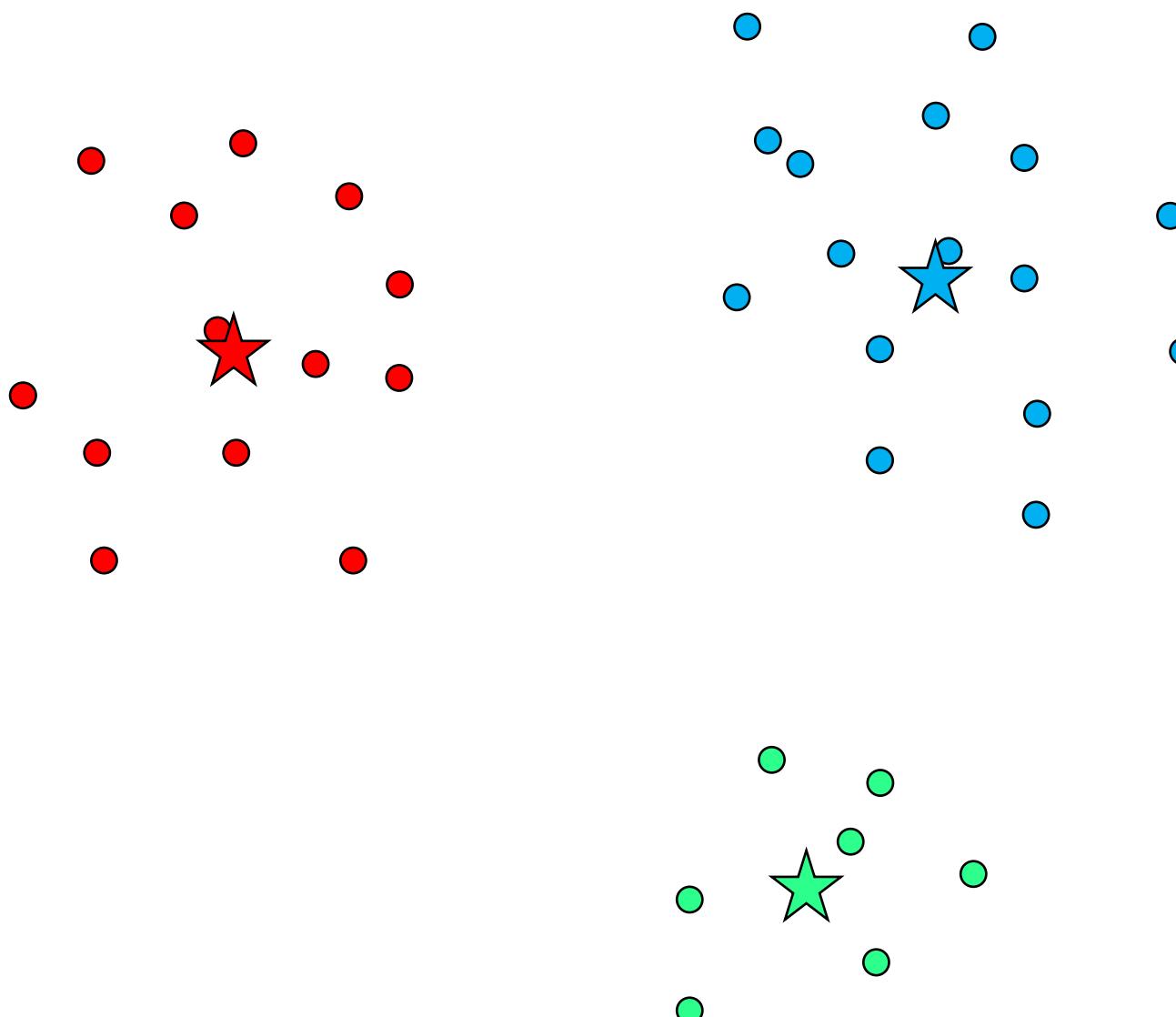
# K-means Example

- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!

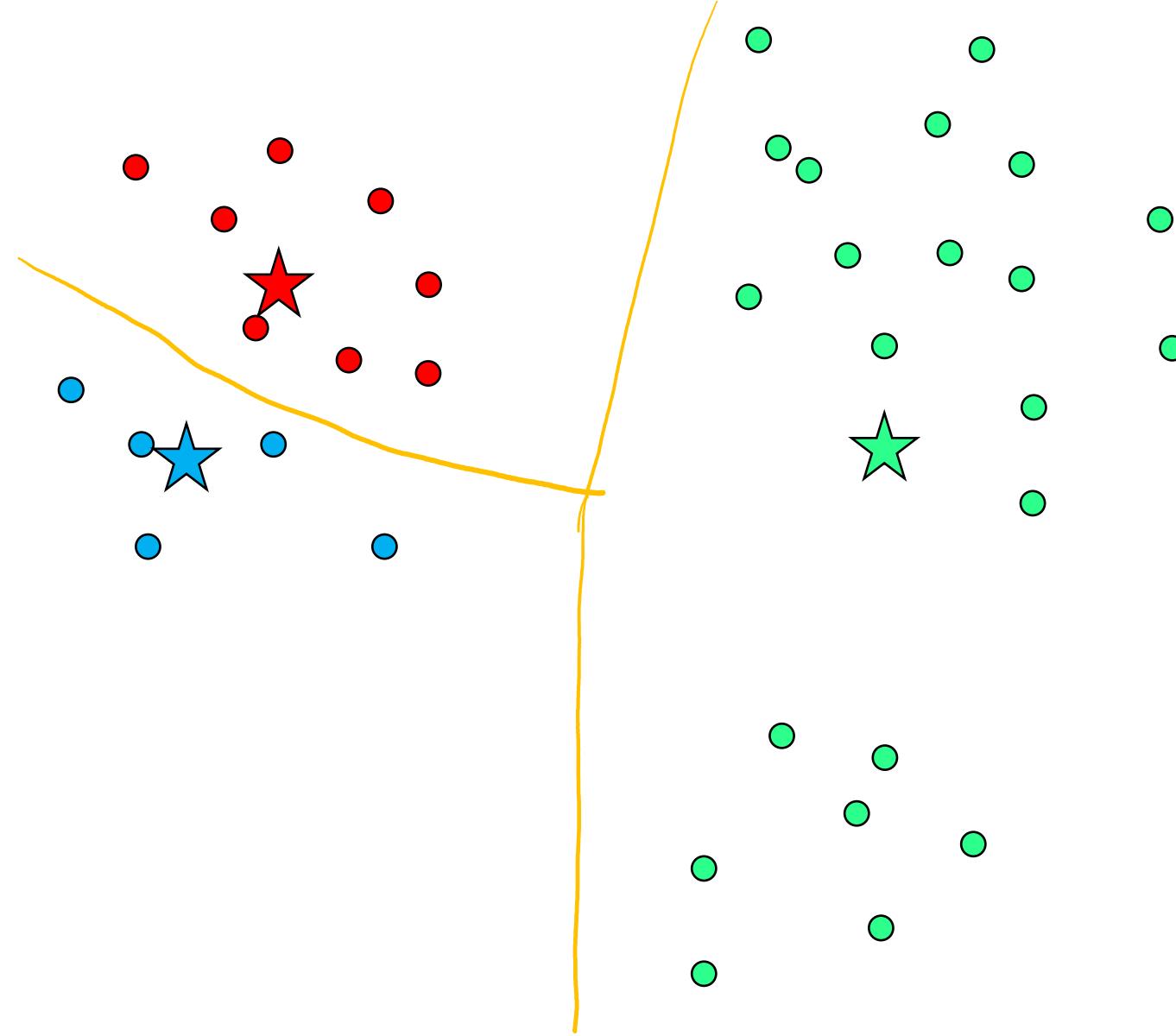


# K-means Example

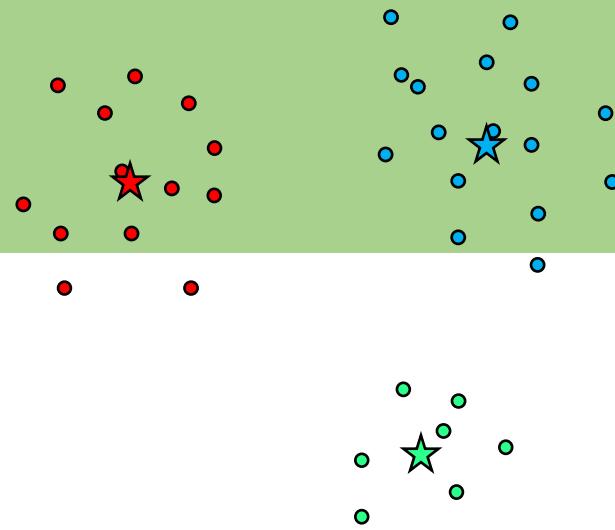
- K is number of clusters
- STEP 1: Guess center locations
- STEP 2: Map out what data point is closest to what center
- STEP 3: Center moves to the centroid of all points it “owns”
- REPEAT!



Doesn't always  
work... can get  
stuck!



# What are we trying to do?



## K-means Objective:

Find cluster centers  $\mathbf{m}$  and assignments  $\mathbf{r}$  to minimize the sum of squared distances of data points  $\{\mathbf{x}^{(n)}\}$  to their assigned cluster centers

$$\min_{\{\mathbf{m}\}, \{\mathbf{r}\}} J(\{\mathbf{m}\}, \{\mathbf{r}\}) = \min_{\{\mathbf{m}\}, \{\mathbf{r}\}} \sum_{n=1}^N \sum_{k=1}^K r_k^{(n)} \|\mathbf{m}_k - \mathbf{x}^{(n)}\|^2$$

$$\text{s.t. } \sum_k r_k^{(n)} = 1, \forall n, \quad \text{where } r_k^{(n)} \in \{0, 1\}, \forall k, n$$

where  $r_k^{(n)} = 1$  means that  $\mathbf{x}^{(n)}$  is assigned to cluster  $k$  (with center  $\mathbf{m}_k$ )

**Note:  $N$  points,  $K$  clusters**

Source: Ethan Fetaya, James Lucas, Emad Andrews

- **Initialization:** Set  $K$  cluster means  $\mathbf{m}_1, \dots, \mathbf{m}_K$  to random values
- Repeat until convergence (until assignments do not change):
  - ▶ **Assignment:** Each data point  $\mathbf{x}^{(n)}$  assigned to nearest mean

$$\hat{k}^n = \arg \min_k d(\mathbf{m}_k, \mathbf{x}^{(n)})$$

(with, for example, L2 norm:  $\hat{k}^n = \arg \min_k \|\mathbf{m}_k - \mathbf{x}^{(n)}\|^2$ )

and **Responsibilities** (1-hot encoding)

$$r_k^{(n)} = 1 \longleftrightarrow \hat{k}^{(n)} = k$$

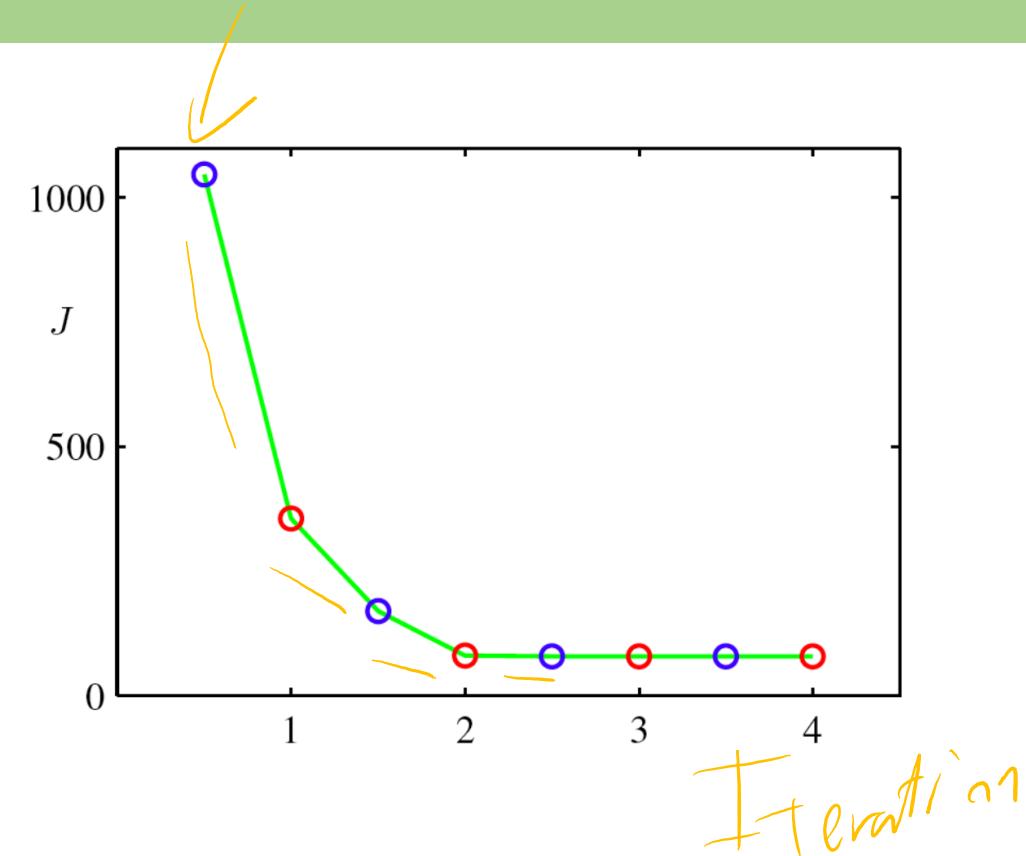
$$\hat{K}^{(3)} = 2$$

- ▶ **Refitting:** Model parameters, means are adjusted to match sample means of data points they are responsible for:

$$\mathbf{m}_k = \frac{\sum_n r_k^{(n)} \mathbf{x}^{(n)}}{\sum_n r_k^{(n)}}$$

# k-Means Convergence

- Whenever an assignment is changed, the sum squared distances,  $J$ , of data points from their assigned cluster centers is reduced
- Whenever a cluster is moved,  $J$  is reduced
- **Test for convergence:** if the assignments do not change in the assignment step, we have converged (to at least a local minimum).

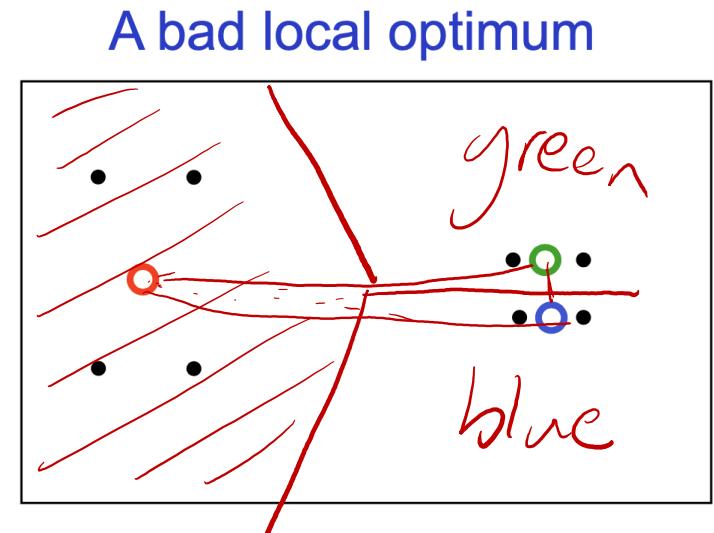


# Local Minima

- There is nothing to prevent k-means from getting stuck at a local minimum

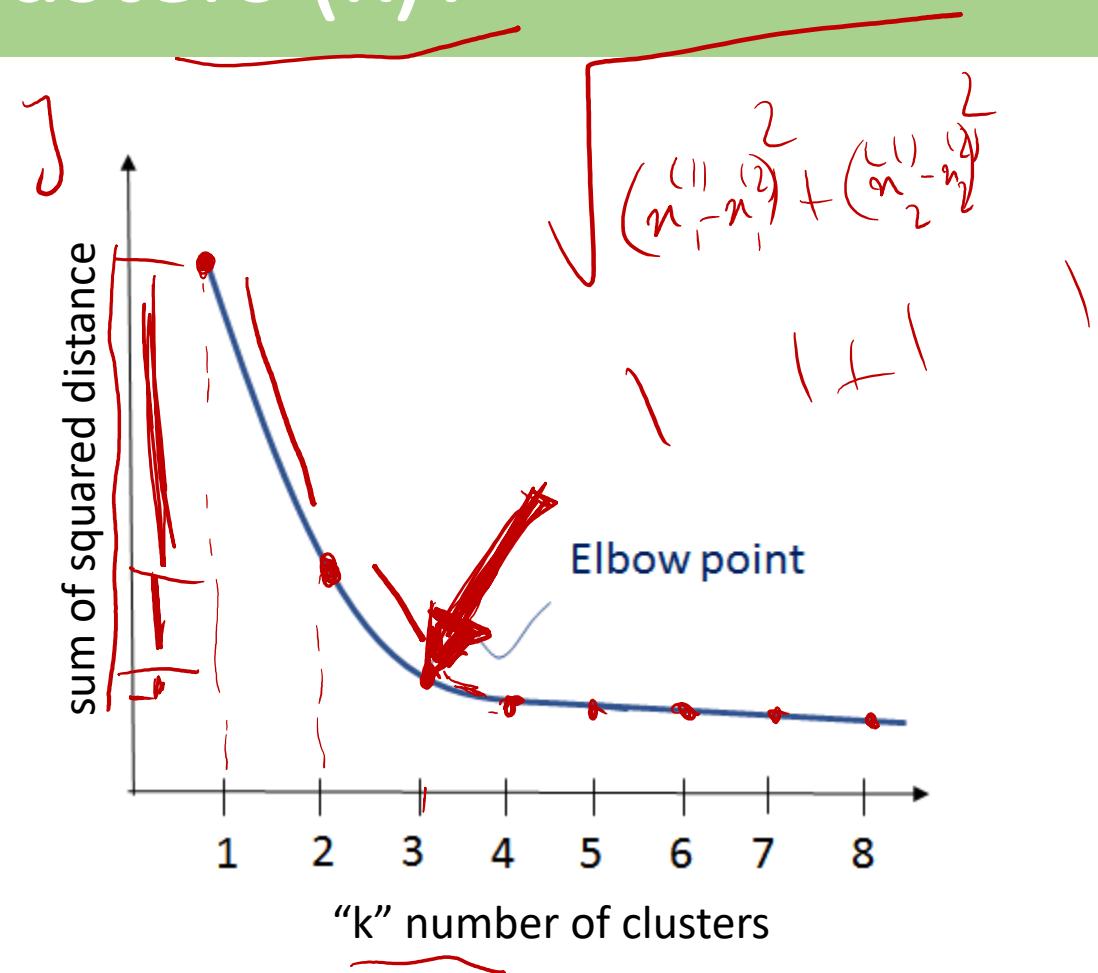
Options for avoiding local minimum:

- we could try many random starting points
- split a big cluster into two
- merge to nearby clusters



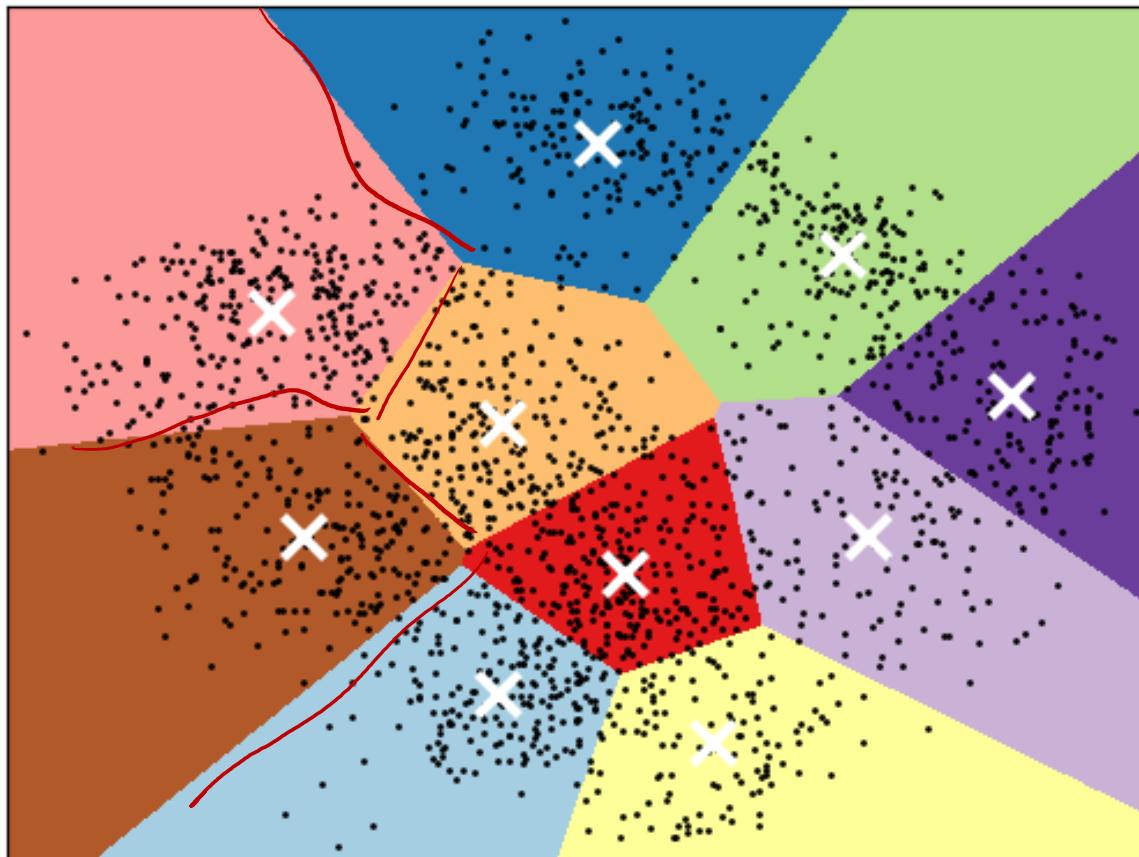
# How to choose number of clusters (k)?

- As k increase the sum of squared distance goes towards zero
- If the plot looks like an arm, then the elbow of the arm is the optimal k
  - e.g., the elbow is at k=3 indicating the optimal number of clusters is 3



# Shape of K-Means Clusters

- K-means split the space according on the closest mean:

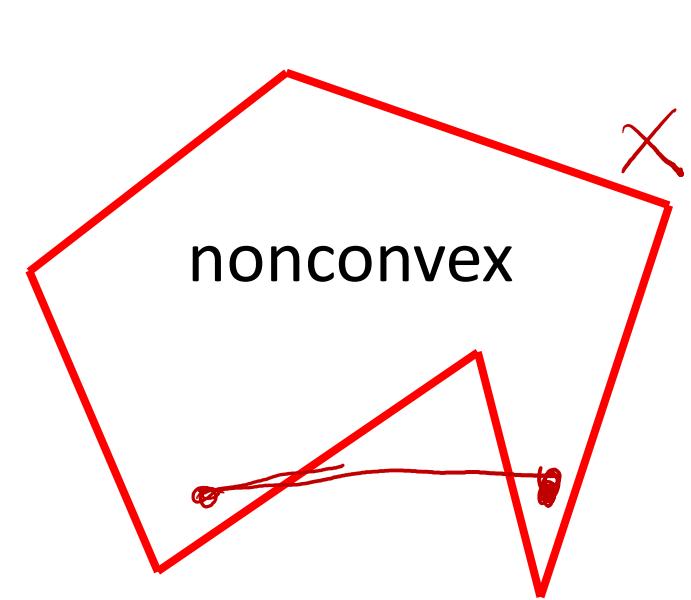
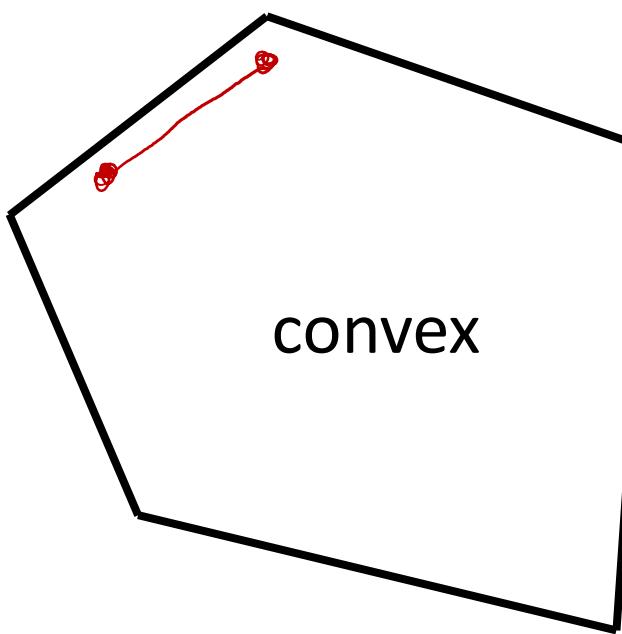
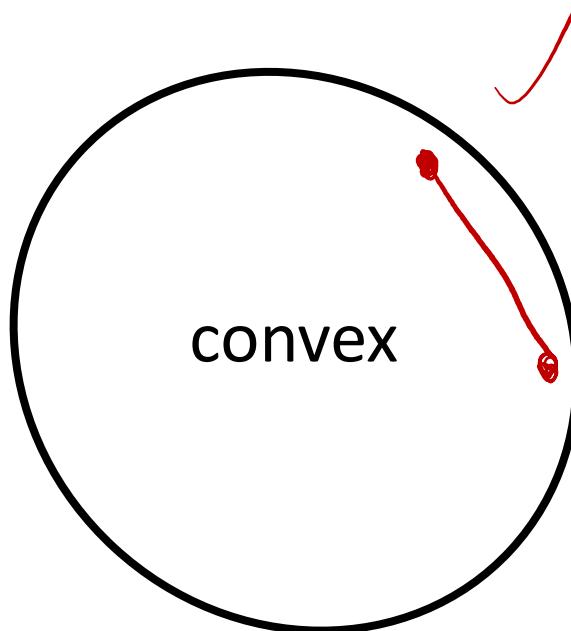


Source: [scikit-learn](#)

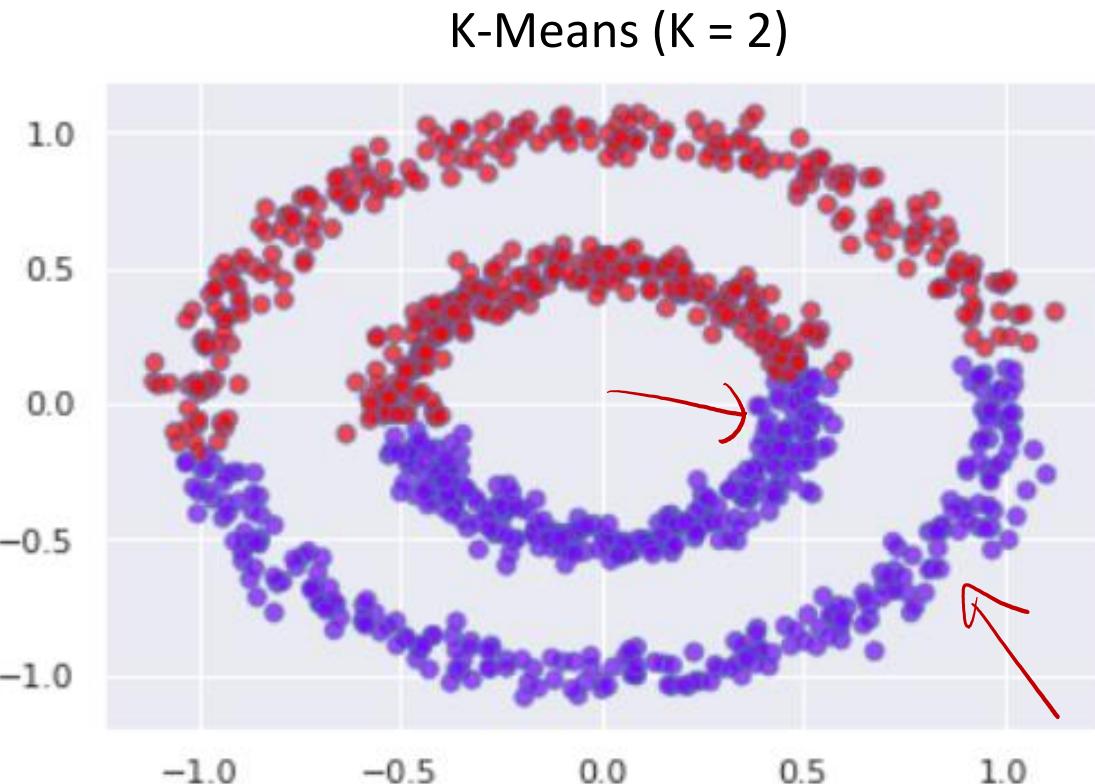
- Note that the clusters form convex regions.

# Convex Region

- A region is convex if any line between two points in the region remains in the region.



# K-Means with Non-Convex Clusters

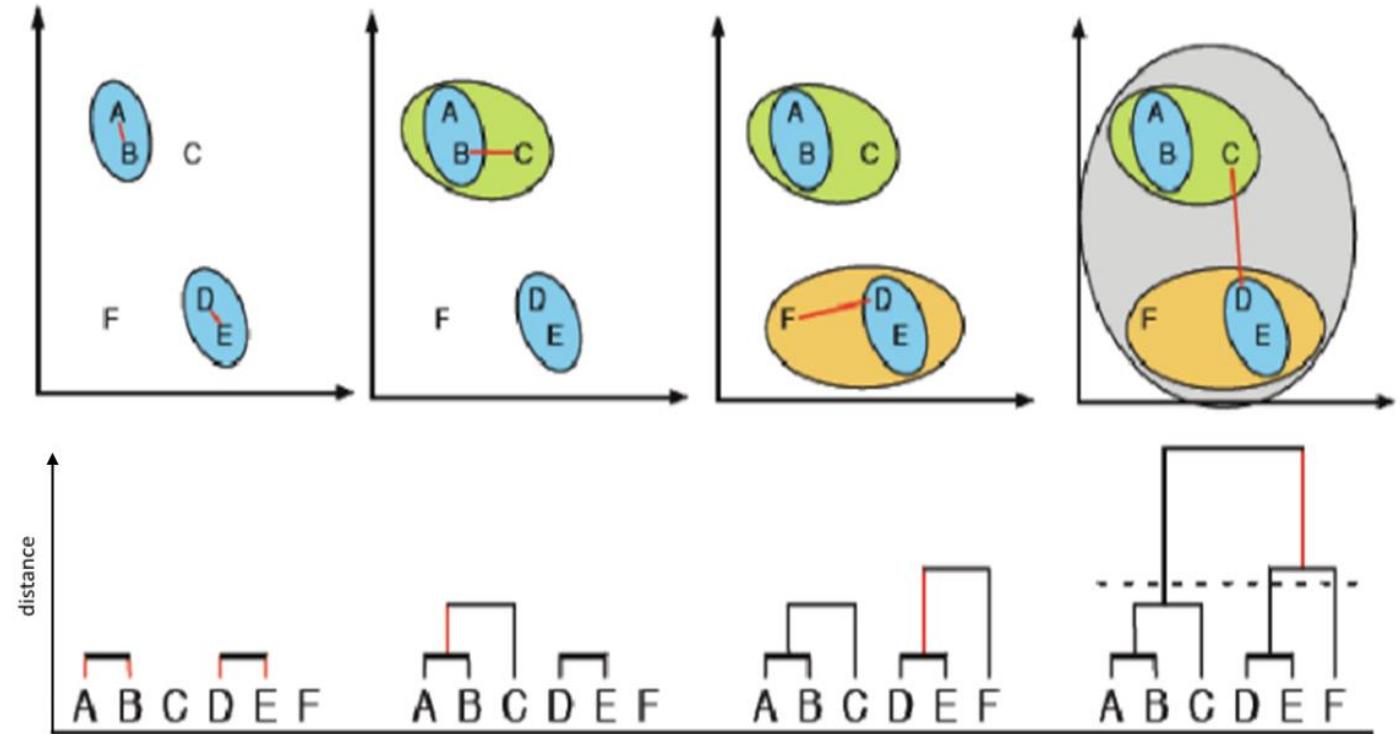


K-means is unable to separate non-convex clusters

Source: [Cory Malkin](#)

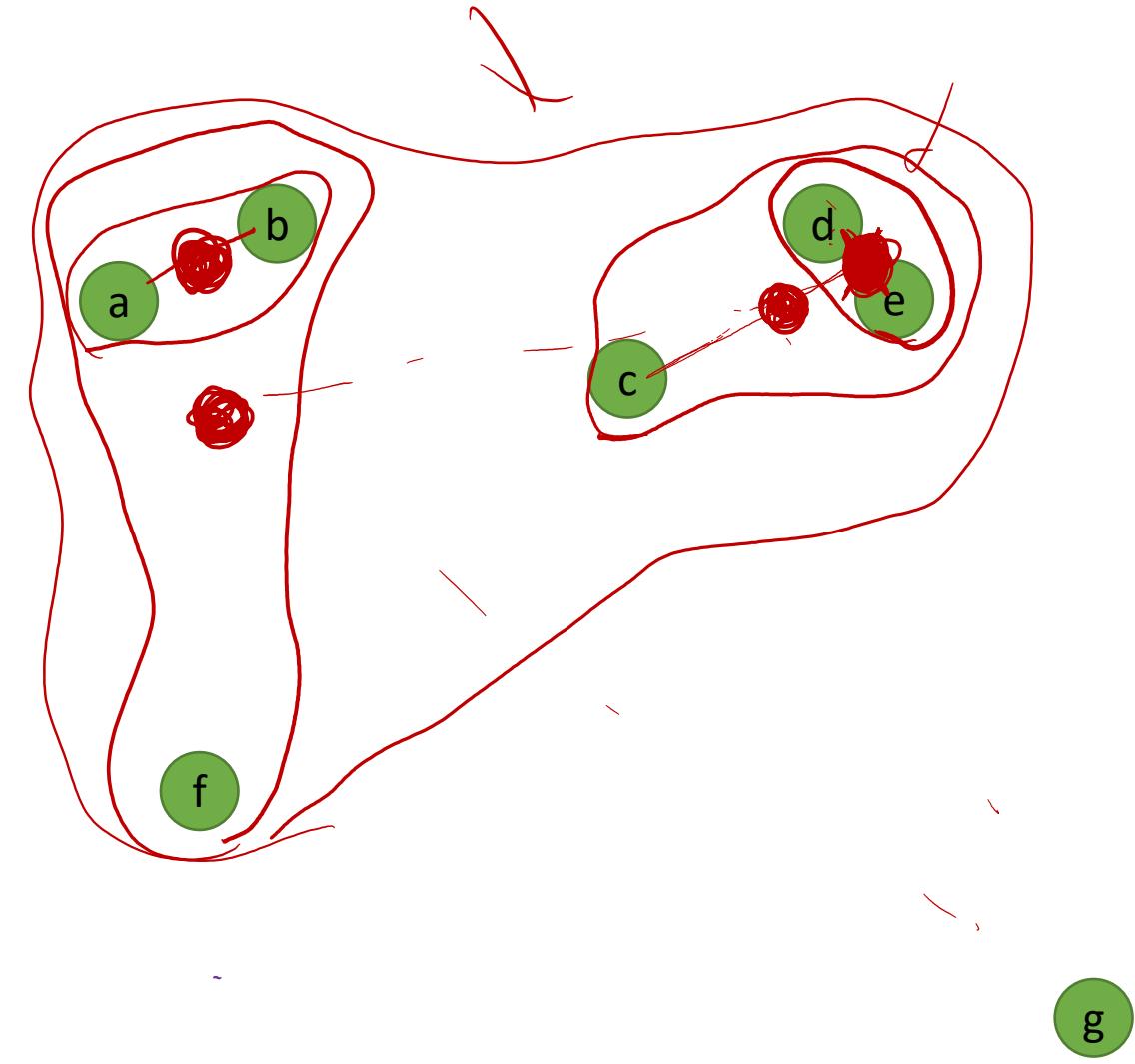
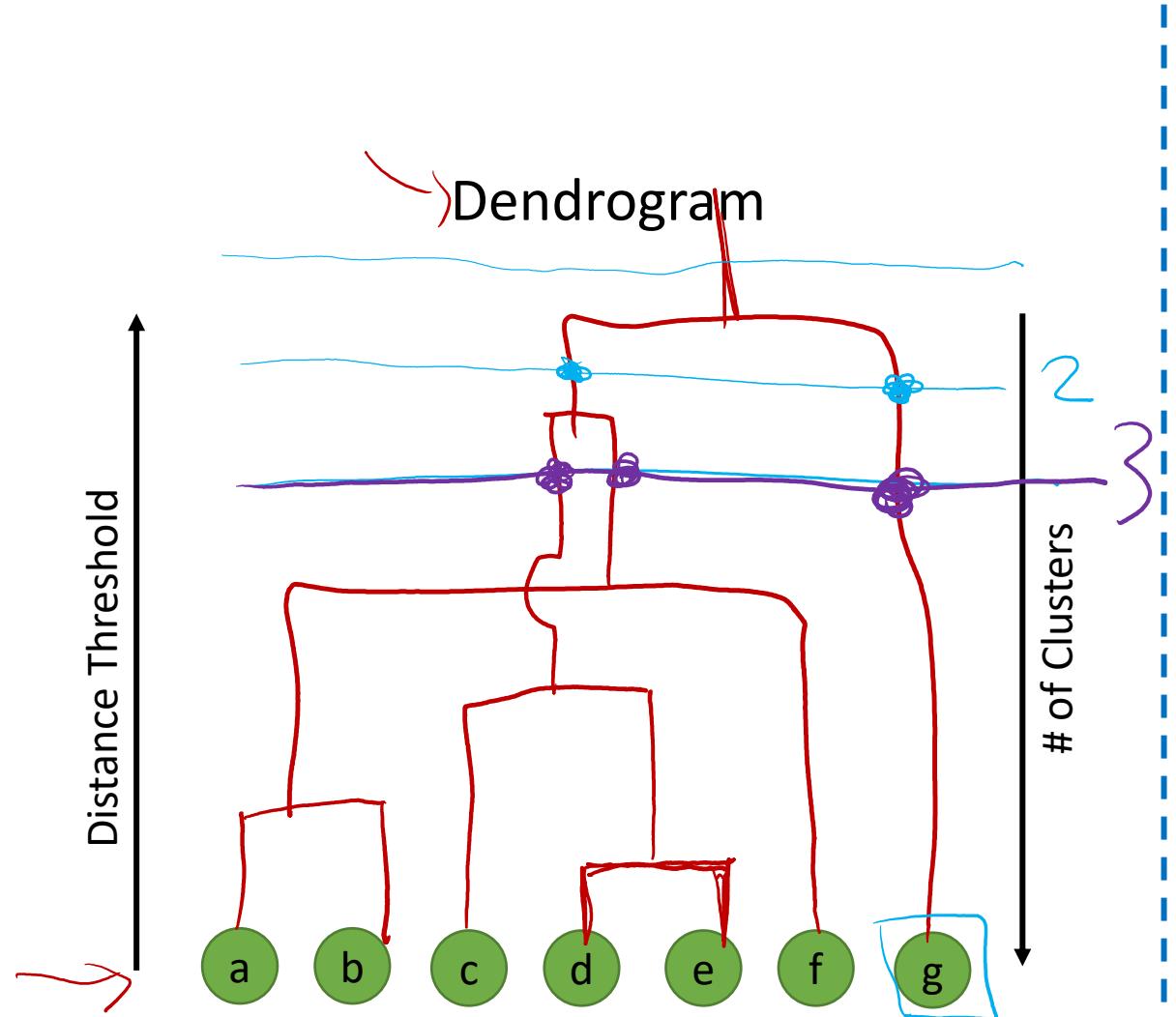
# Agglomerative Clustering

- A type of Hierarchical Clustering
- Algorithm:
  1. Starts with each point in its own cluster
  2. Each step merges the two “closest” clusters

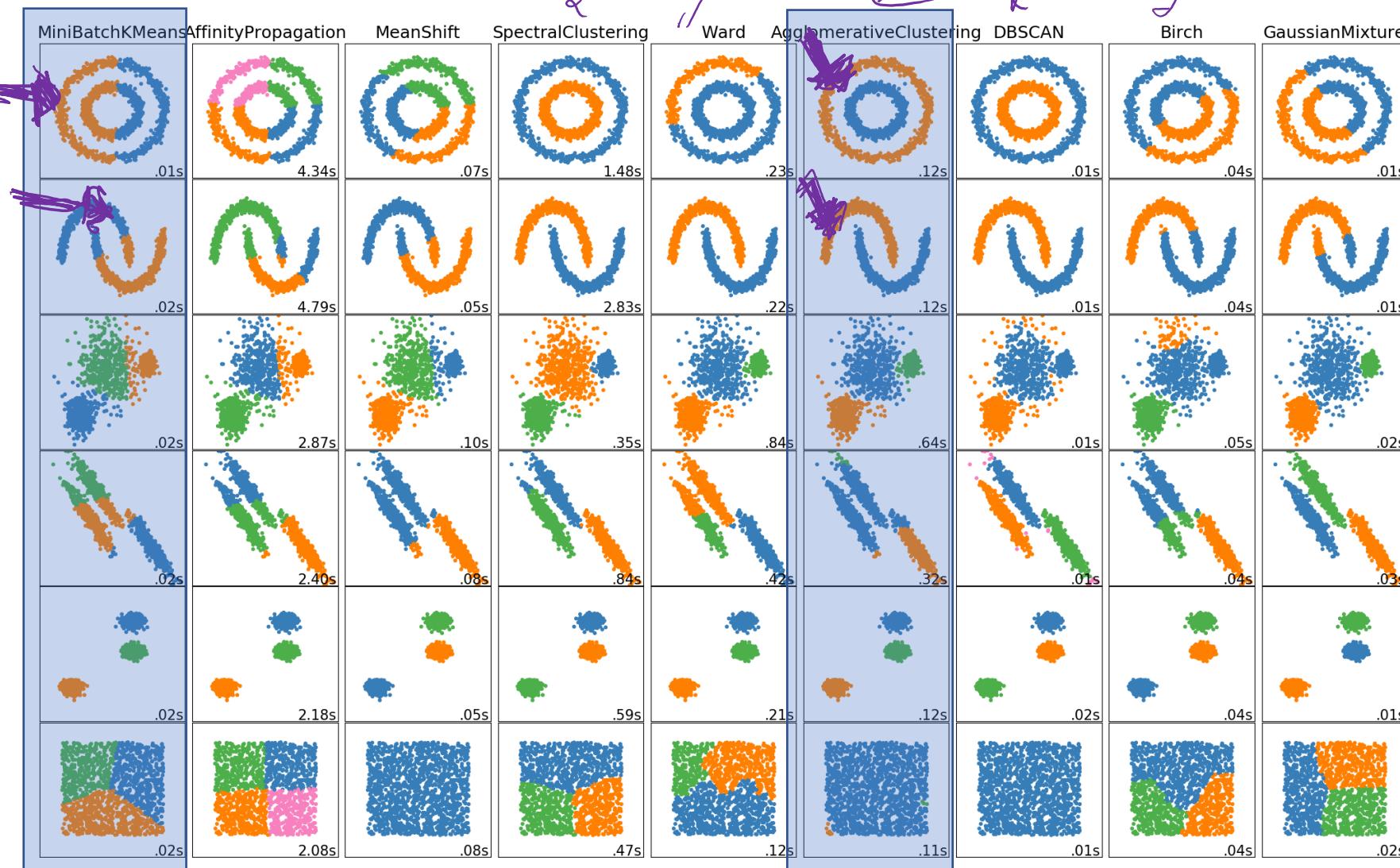


Source: [MachineLearningStories](#)

# Example: Agglomerative Clustering



# Comparison of Clustering Methods

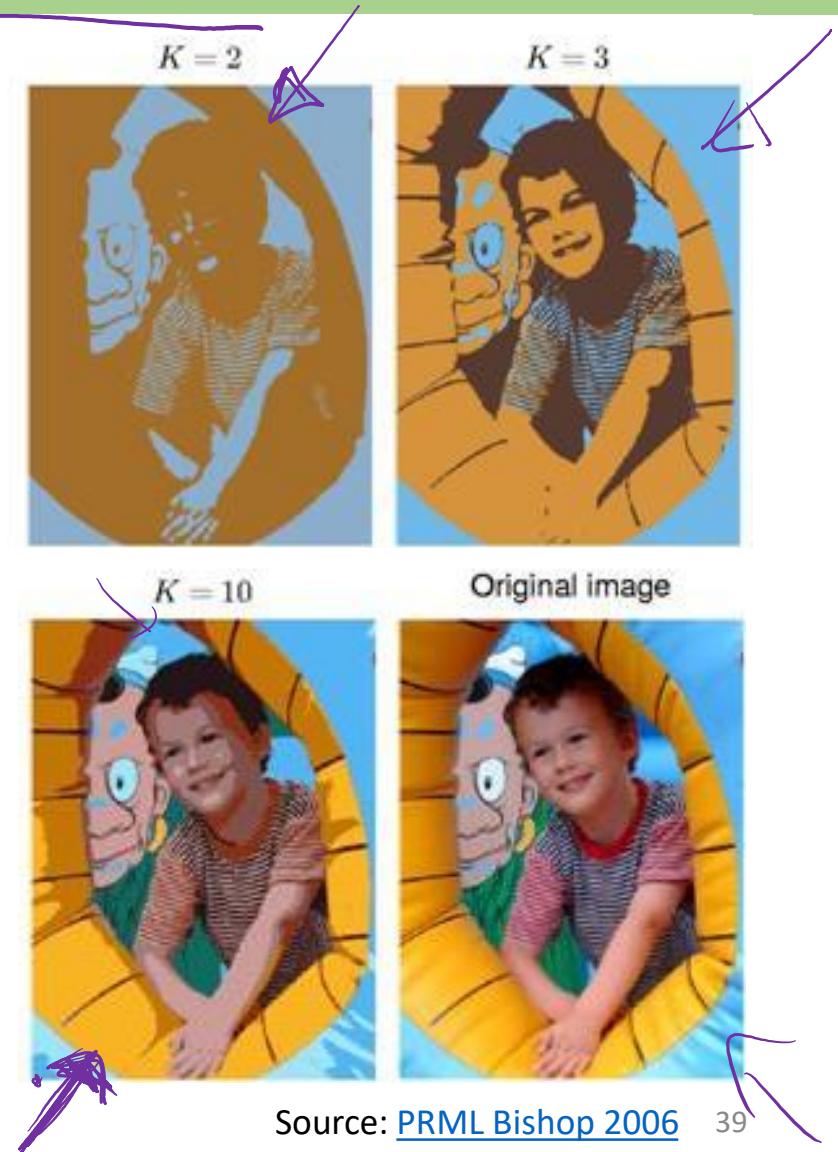


- There are many clustering algorithms to chose from
- Performance will depend on your data

Source: [scikit-learn](#)

# Applications in Computer Vision

- Replace samples with the mean of their cluster (vector quantization)
- Visual example:
  - Inputs: color pixels represented as RGB values
  - Outputs: cluster (average RGB) value obtained using k-means
- Q: How can this be applied for compression?
- Q: How can this be applied for image segmentation?



# K-Means Code Example (Google Colab)

10 : 09

# Part 2 Probability Theory

**Readings:**

- Chapter 6.1-5 MML Textbook

# Overview

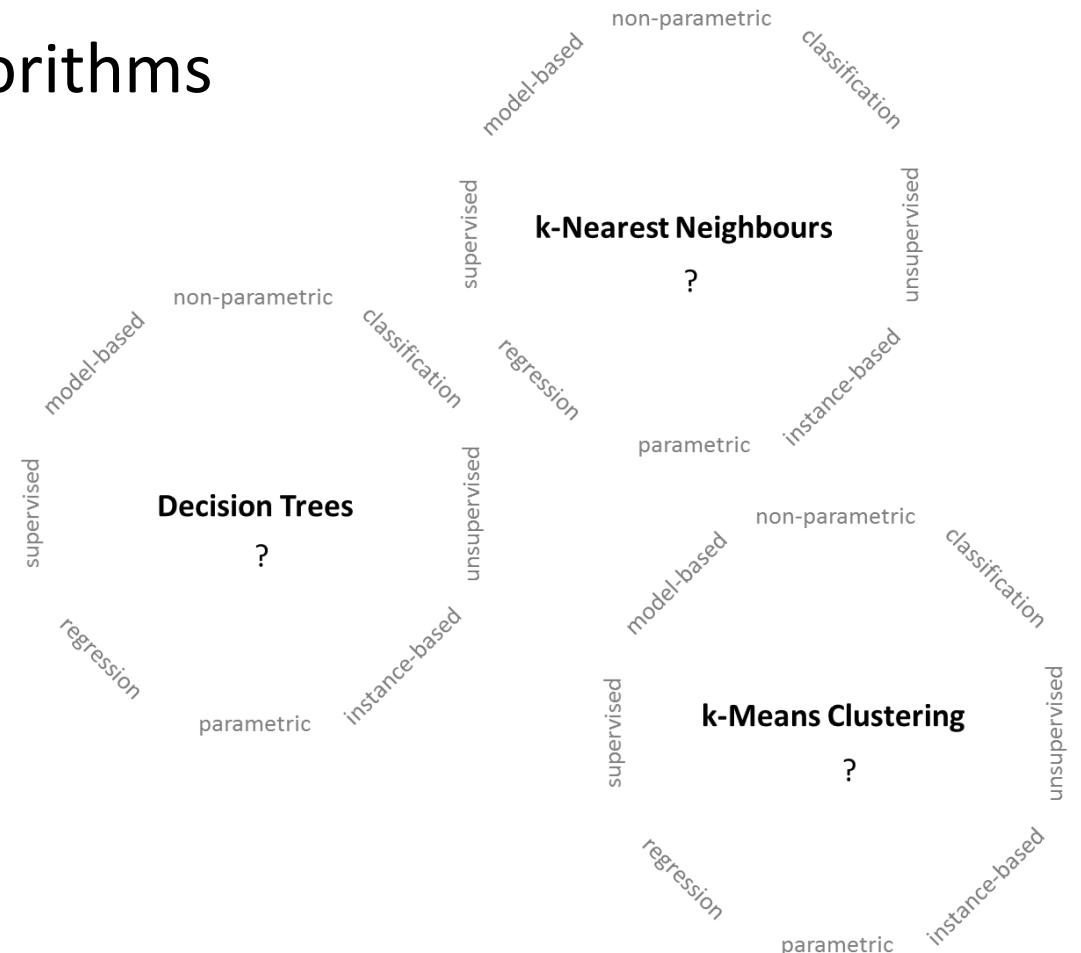
- Compared several machine learning algorithms

- Supervised Learning

- k-Nearest Neighbours
- Decision Trees

- Unsupervised Learning

- k-Means Clustering



- Q: How did we measure performance?

# Measuring Uncertainty

- In machine learning it is important to understand how confident we can be about the decisions (or predictions) being made by our models/algorithms.
- Q: A model/algorithm is tested on two new samples and classifies both correctly. What is the accuracy and how confident are you with that result?

# Agenda for parts 2 and 3 of Lecture 4

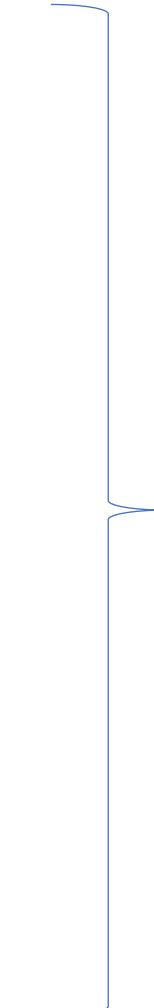
## Probability Theory

- Examples
- Summary Statistics
- Gaussians

## Mixture of Gaussians

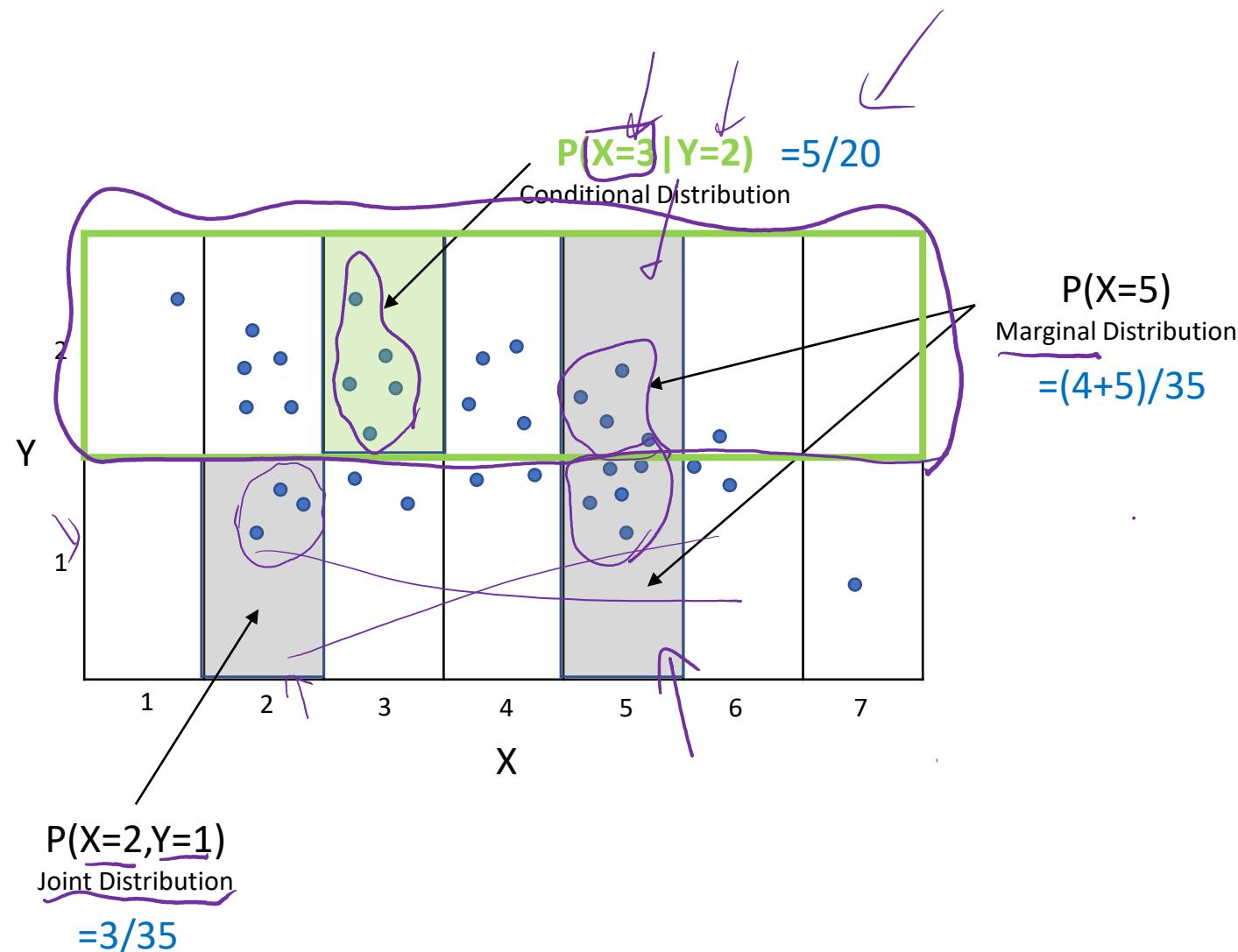
- Anomaly Detection

- Performance Metrics
  - Precision and Recall
  - Confusion Matrix
  - ROC and AUC

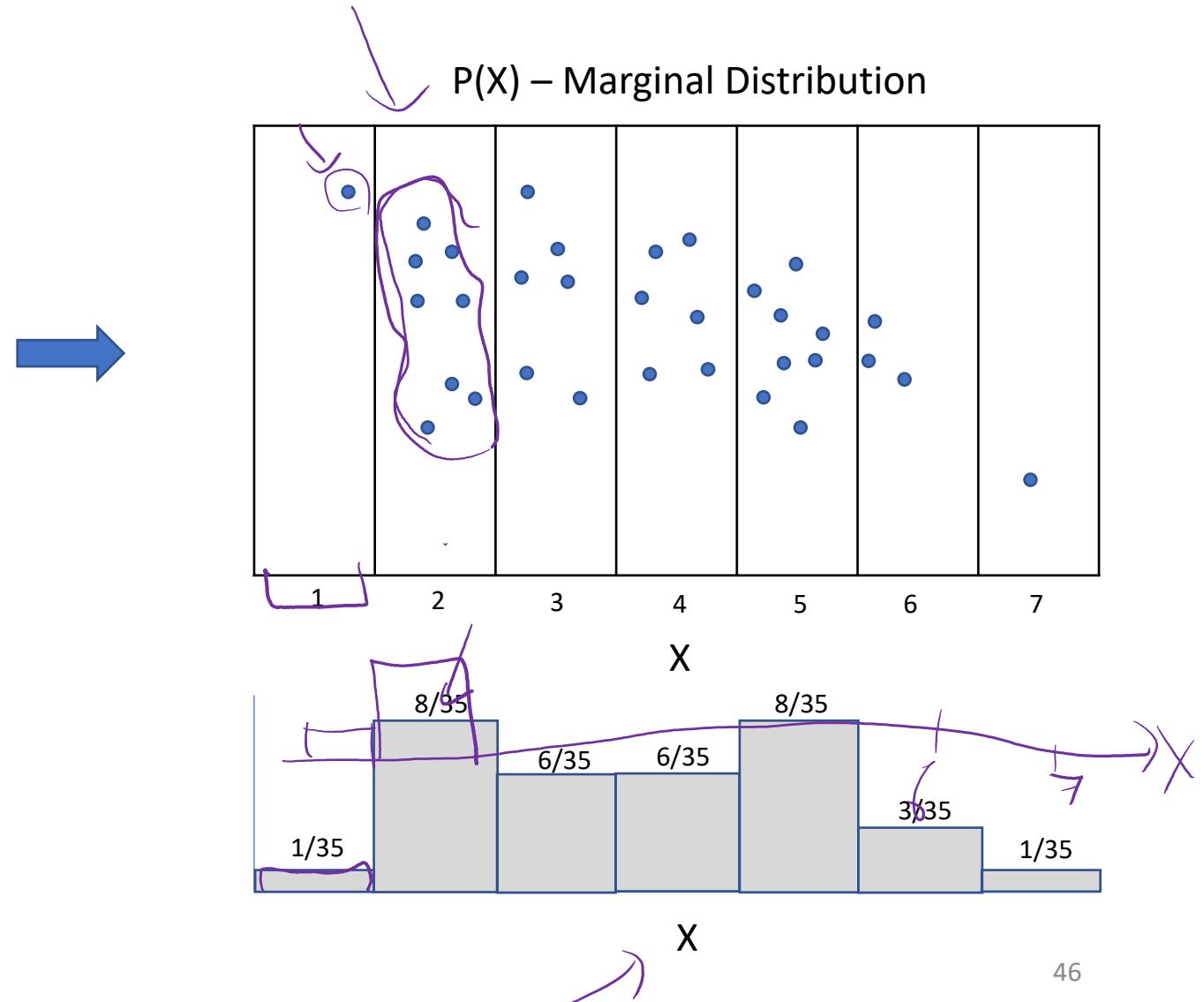
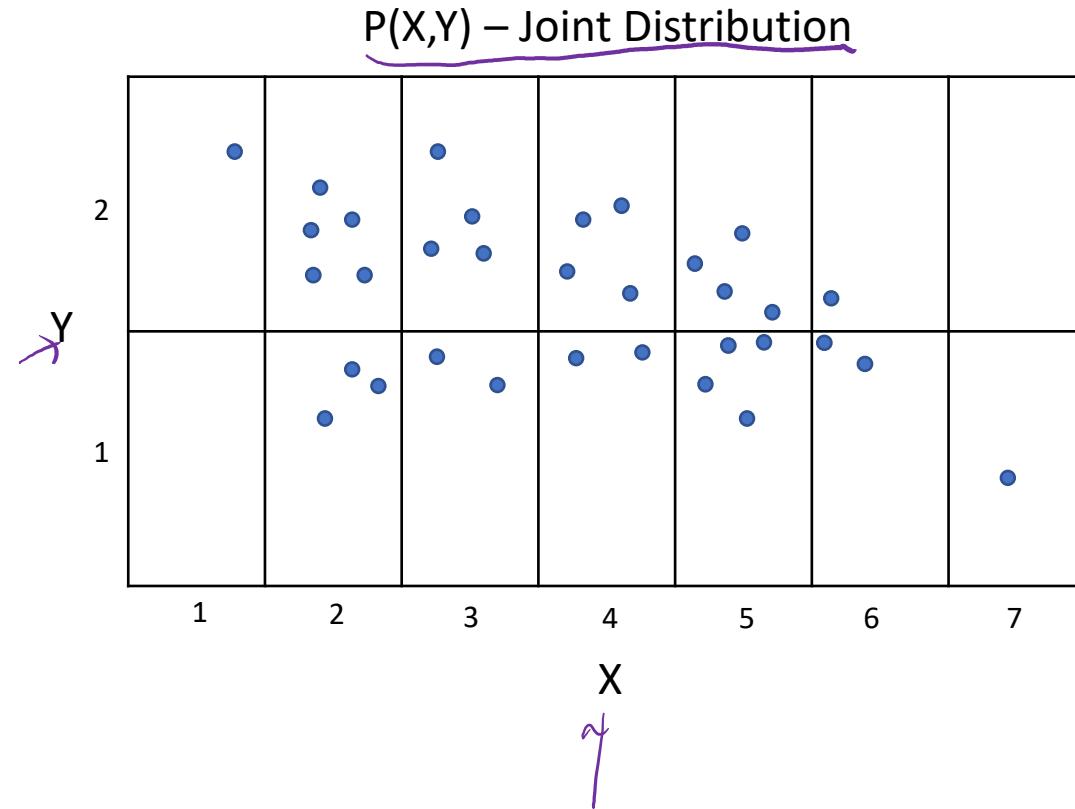


**Theme:  
Measuring Uncertainty**

# Example: Random Variables (discrete)

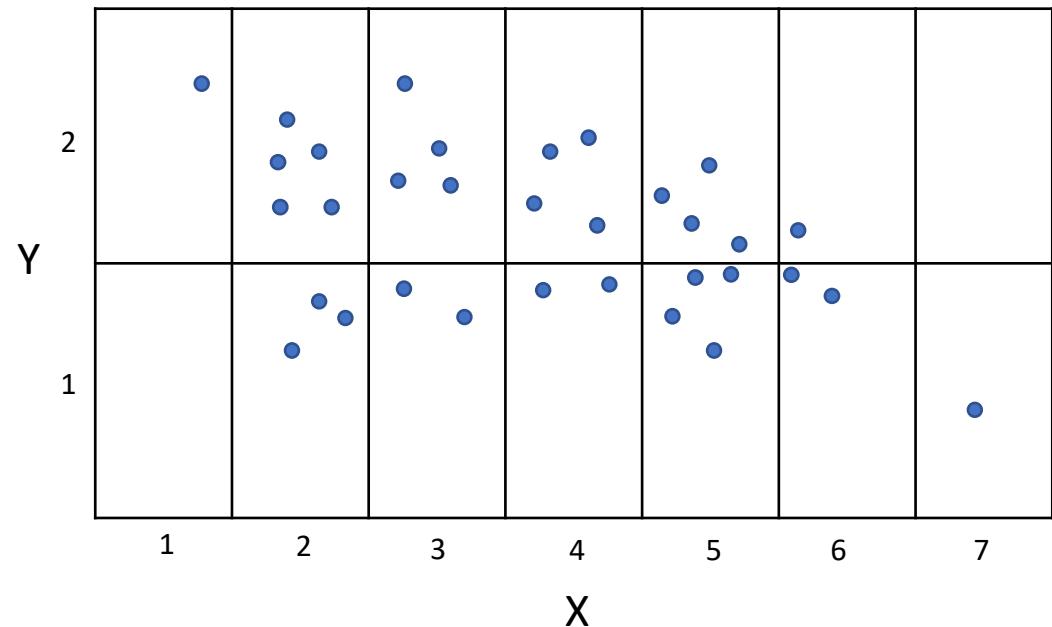


# Example: Marginal Distribution of X

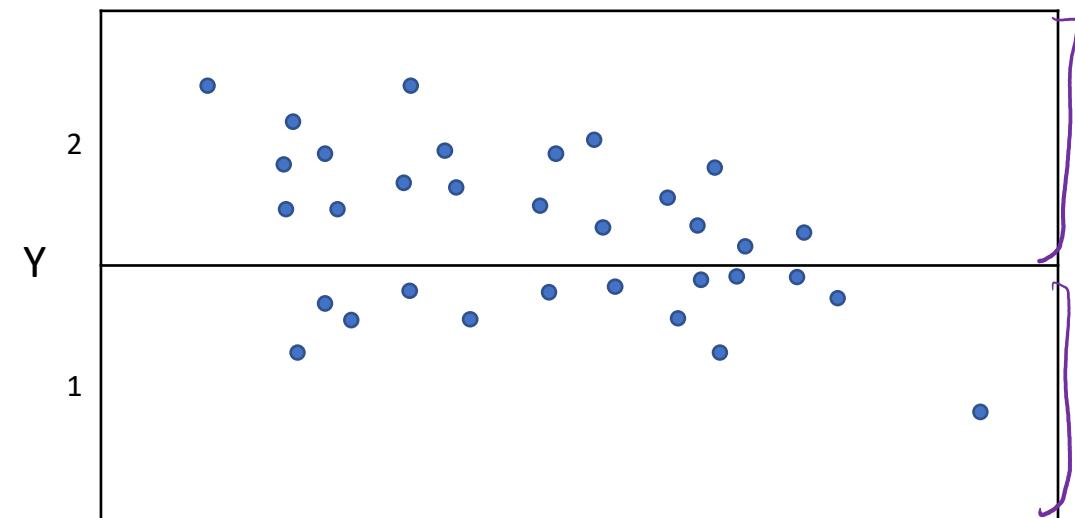


# Example: Marginal Distribution of Y

$P(X,Y)$  – Joint Distribution

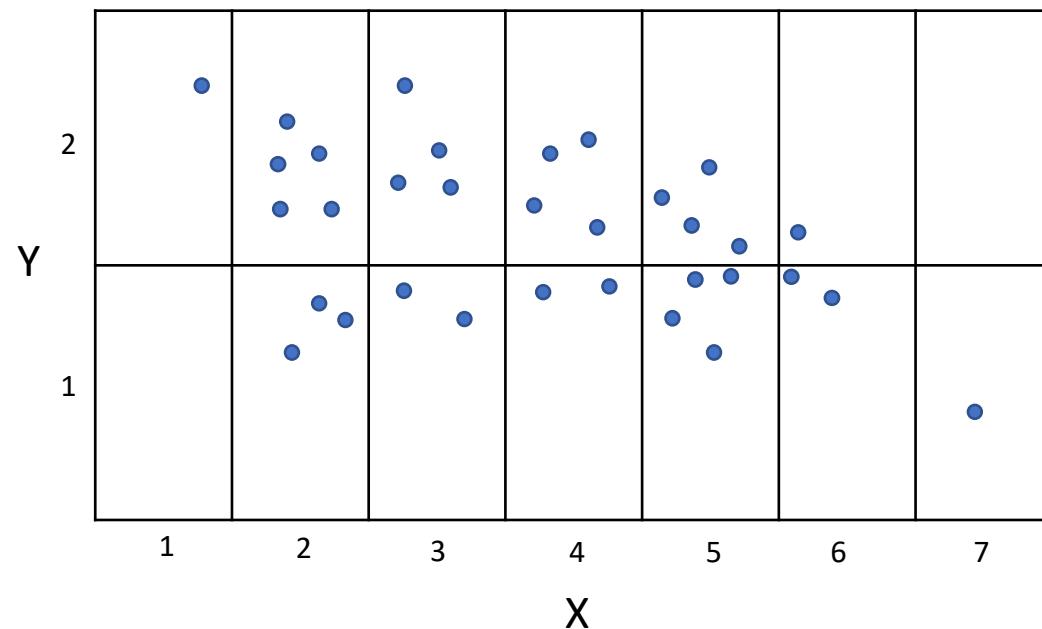


$P(Y)$  – Marginal Distribution

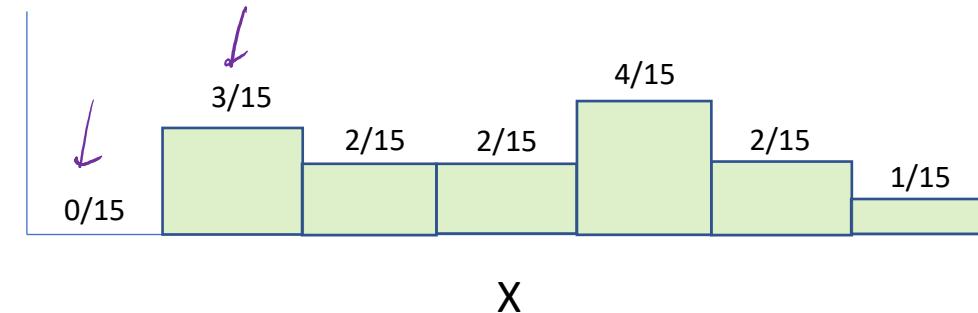
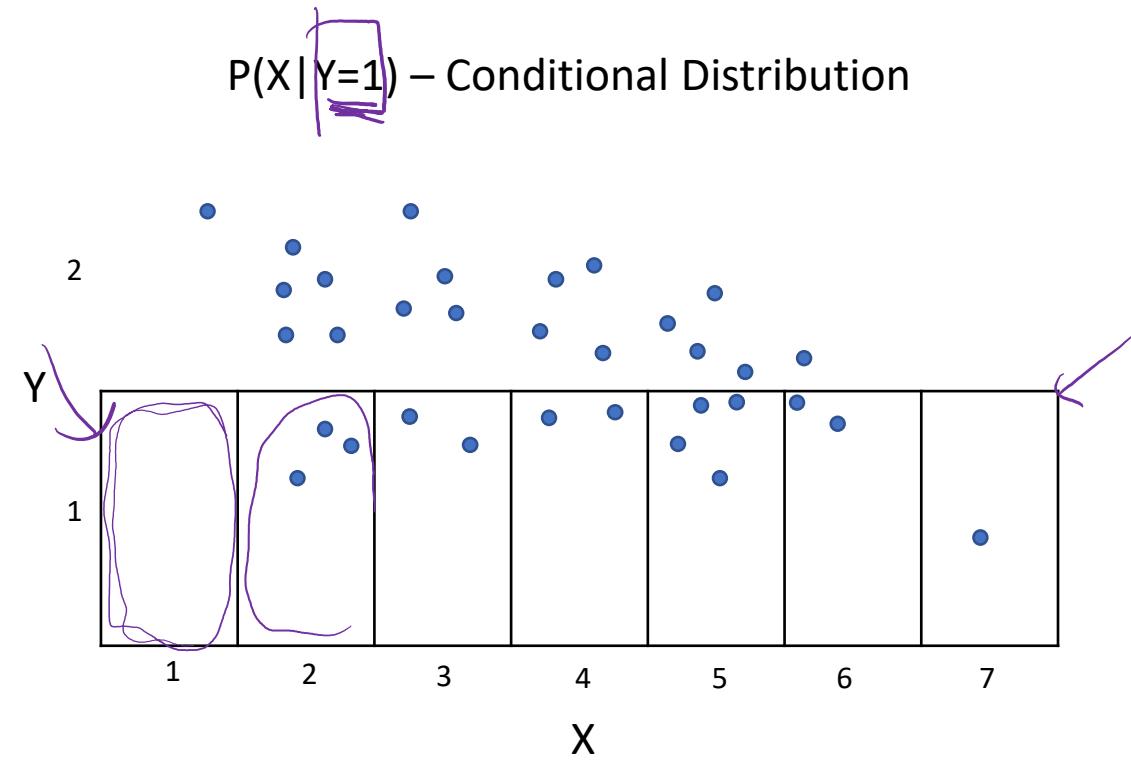


# Example: Conditional Distribution $X|Y$

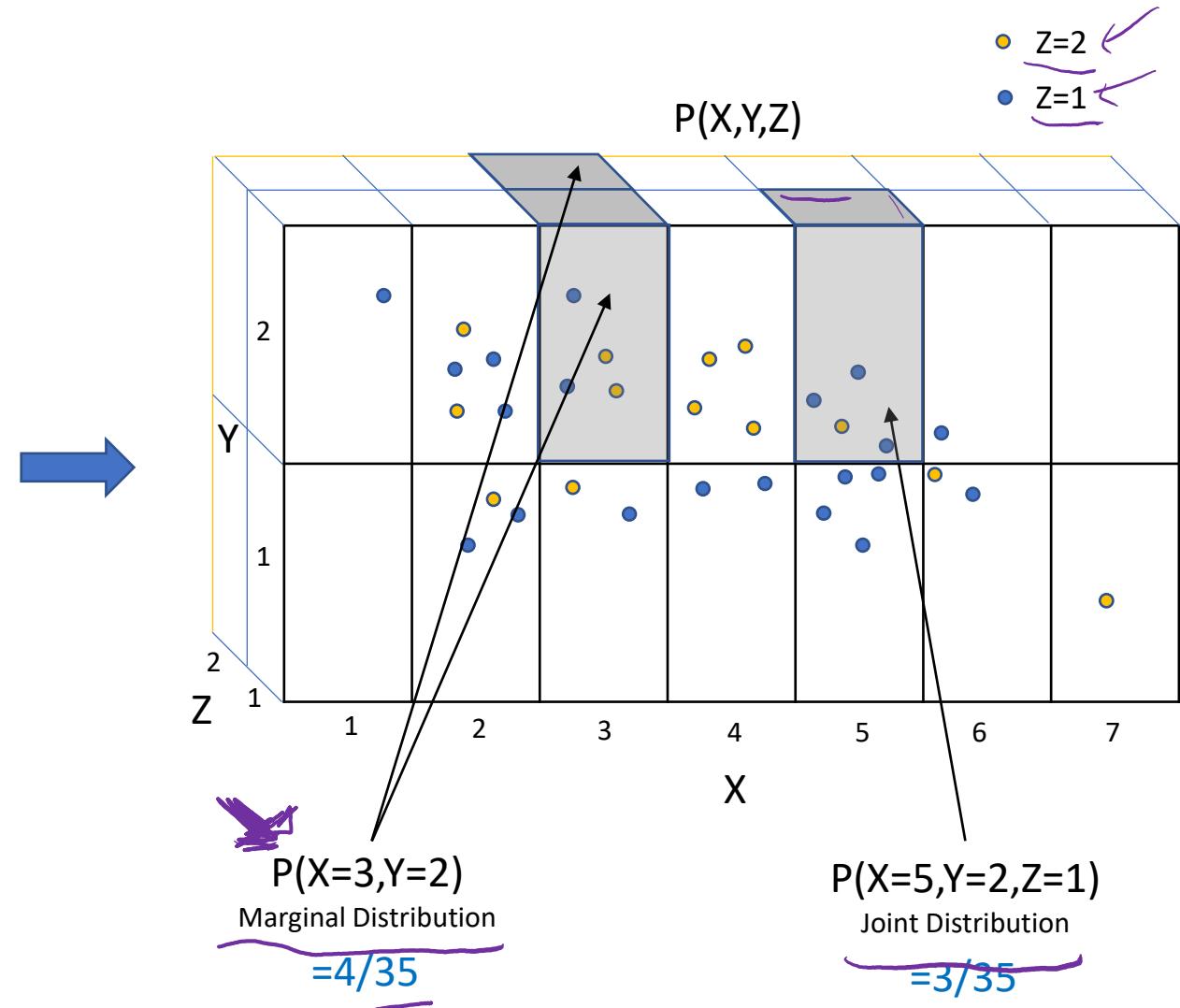
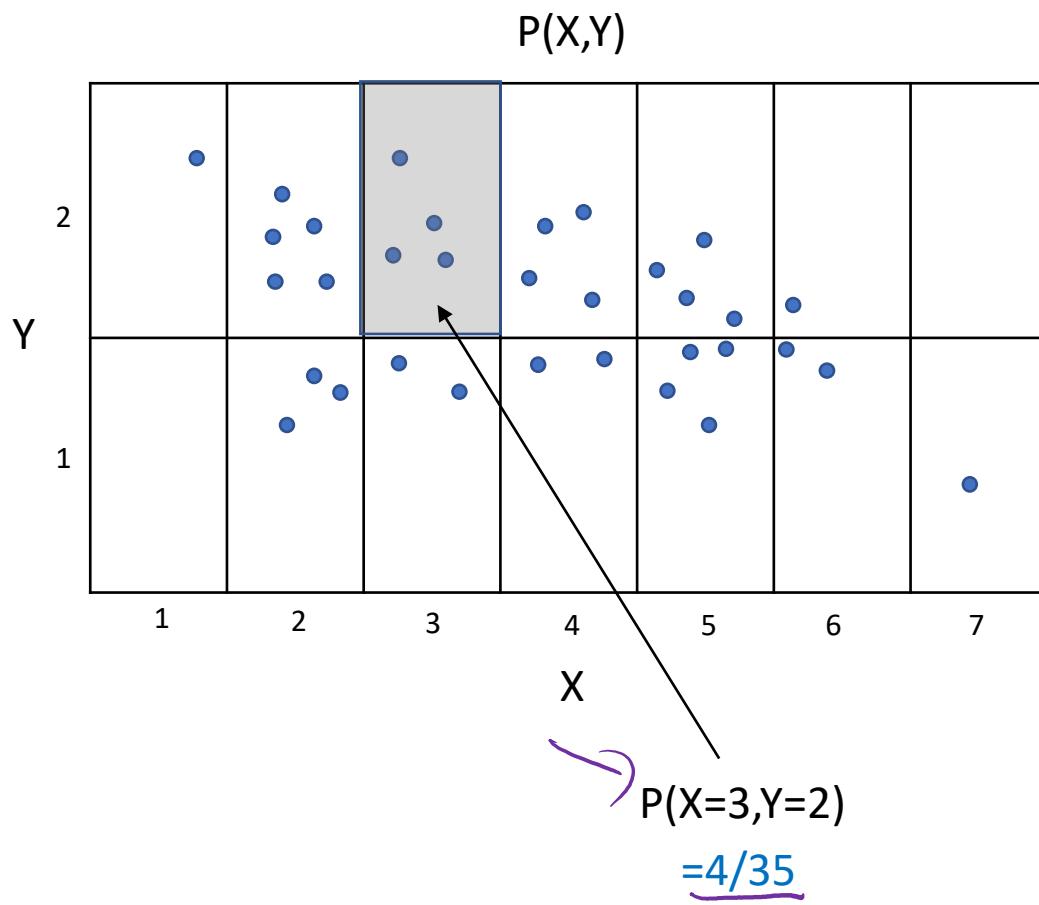
$P(X,Y)$  – Joint Distribution



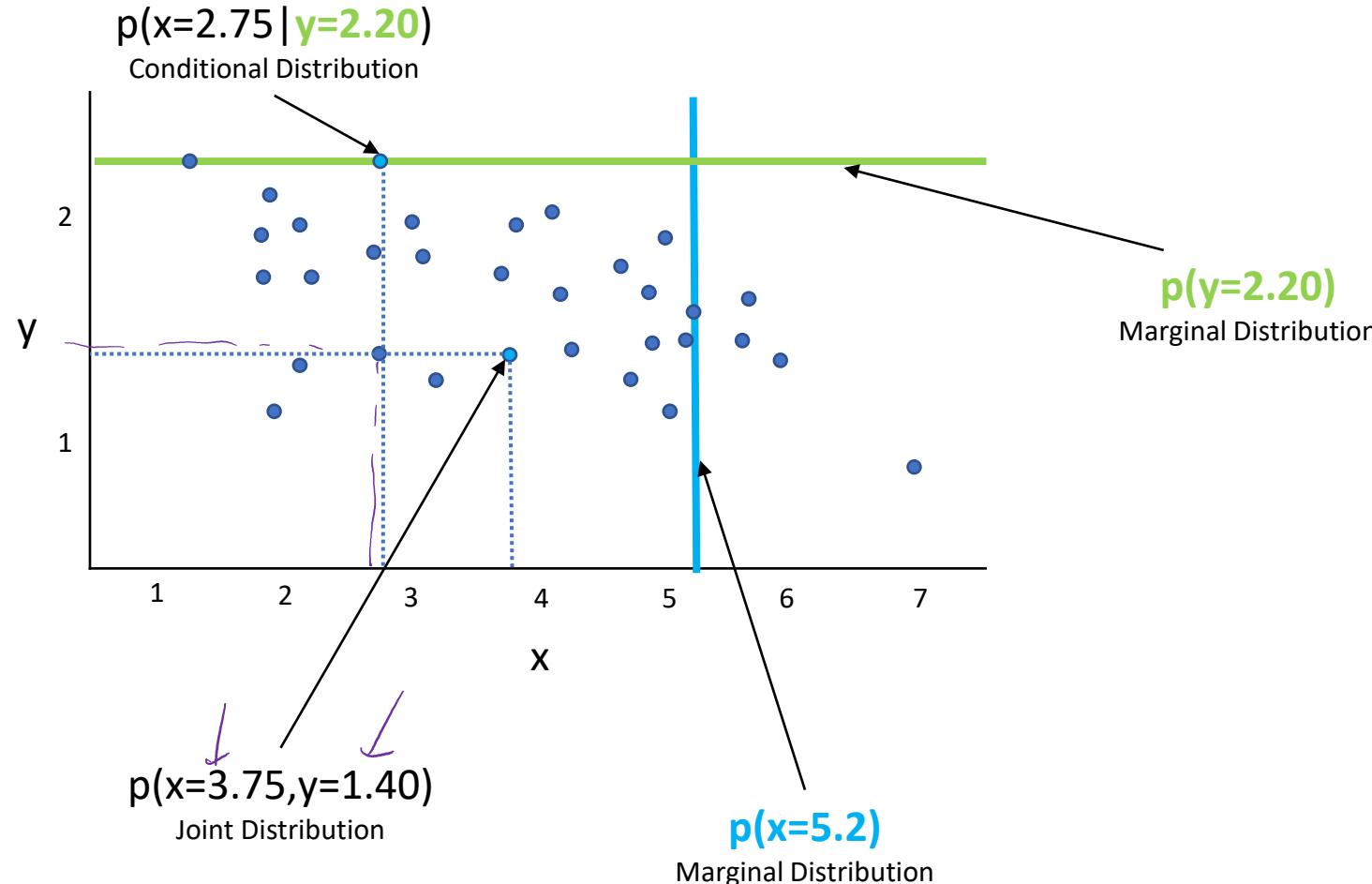
$P(X|Y=1)$  – Conditional Distribution



# Example: From 2 to 3 Random Variables



# Example: Continuous Distributions



- Probability at a point is meaningless
- Needs to add up to 1
- Consider areas

# Probability with Real Data

```
pd.crosstab(df['Pclass'], df['Sex'])
```

Counts

		Sex		↓
		0	1	
Class	1	80	136	216
	2	97	87	184
All	3	372	119	491
	All	549	342	891

Probabilities



		Sex		Class
		0	1	
1	0.090	0.153	0.242	All
	0.109	0.098	0.207	
2	0.418	0.134	0.551	
All	0.616	0.384	1.000	

```
pd.crosstab(df['Pclass'], df['Sex']) / 891
```

# Probability with Real Data

Counts	Sex			
	0	1	All	
Class	1	80	136	216
	10 dead 70 survived			
2	97	87	184	
3	372	119	491	
All	549	342	891	

Probabilities	Sex			
	0	1	All	
Class	1	0.090	0.153	0.242
2	0.109	0.098	0.207	
3	0.418	0.134	0.551	
All	0.616	0.384	1.000	

Q: What is the probability of a random passenger from the dataset belonging to the 3<sup>rd</sup> class?

Given that there were 70 first-class male passengers who survived and 10 first-class male passengers who did not, what is the probability of survival for first-class male passengers?

$$\frac{70}{80} = \frac{7}{8}$$

Q: What is the probability of there being a first-class male passenger that survived?

# Example: Permutations and Combinations

## Permutation

- arrangement of items in which **order matters**

## Combination

- selection of items in which **order does not matter**

Order Matters	Repetition Allowed	Formula
– Yes (Permutation)	Yes	$P(n, r) = n^r$
– Yes (Permutation)	No	$P(n, r) = \frac{n!}{(n - r)!}$
– No (Combination)	No	$C(n, r) = \frac{n!}{r!(n - r)!}$
– No (Combination)	Yes	$C(n + r - 1, r) = \frac{(n + r - 1)!}{r!(n - 1)!}$

n – number of items in a set

r – number of items selected from the set

# Example: Permutations (n=5, r=3)

Q: How many ways can we fill 3 positions in a company using a pool of 5 applicants?

$$P(n, r) = \frac{n!}{(n - r)!}$$

$\approx \frac{5!}{2!} \approx 5 \times 4 \times 3 \approx 60$

Q: How many ways can we fill 3 positions in a start-up using a pool of 5 applicants if each person can potentially be given more than one position?

$$P(n, r) = n^r \approx 5^3 = 125$$

$5 \times 5 \times 5$

# Example: Combinations ( $n=5$ , $r=3$ )

Q: How many ways can we select 3 people from a pool of 5 applicants to give them a tour of the company?

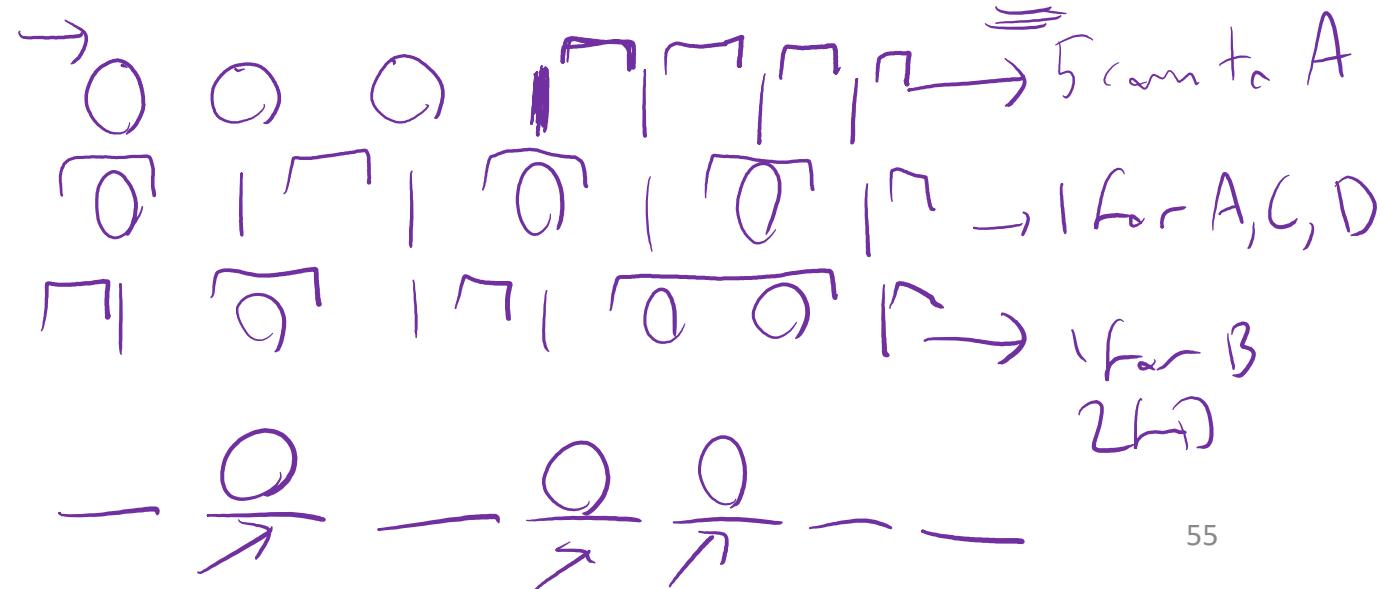
$$C(n, r) = \frac{n!}{r!(n - r)!}$$

$$P(n, r) = \frac{5!}{3!} = 10$$

Q: How many ways can we distribute three identical coins among 5 individuals?

$$C(n + r - 1, r) = \frac{(n + r - 1)!}{r!(n - 1)!} = \binom{7}{3}$$

$$\geq \frac{7!}{3!4!} = \frac{7 \times 6 \times 5}{6} = 35$$



# Let us Summarize...

# Why Probability?

- Probability theory is a **mathematical framework for quantifying our uncertainty** about the world.
  - It allows us (and our software) to reason effectively in situations where being certain is impossible.
- Probability theory is at the foundation of many machine learning algorithms.

# Perspectives on Probability

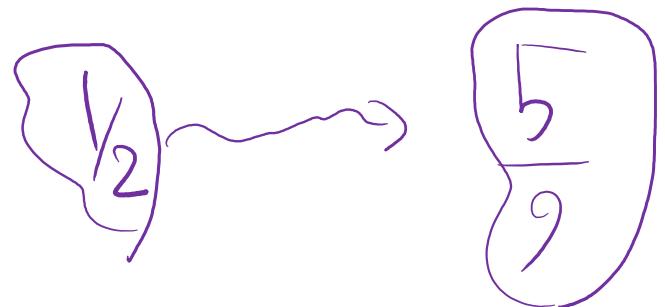
- **Objectivist perspective:** randomness is fundamental to the universe.
  - They would say that the probability of a fair coin coming up heads is 0.5, because that's the nature of fair coins.
- **Subjectivist perspective:** probabilities represent our degree of belief that an event will occur.
  - If we knew the initial position of the coin and how the force was applied, then we could determine with certainty if it would come up heads or tails.
  - Under this perspective, probability is a measure of our ignorance (like not knowing how the force is applied to the coin).

# Frequentists

- **Frequentist's position:** estimations come from experiments and experiments only.
  - e.g., if we want to estimate how likely a six-sided die is to roll a 4, we should roll the die many times and observe how frequently 4 appears.
    - This method works well when we have a large amount of data, but with fewer examples we can't be confident in our estimates.
    - If we haven't seen a 4 after five rolls, does that mean a 4 is impossible?
    - The other issue is that we **can't inject any of our prior knowledge about dice into our estimates.** If we knew the die was fair, not seeing a 4 in the first five rolls is completely understandable.
- **Bayesian perspective** allows us to combine our prior beliefs with our observations

# Bayesian vs Frequentist

- For example: Imagine that a coin we believe to be fair is flipped three times and results in three heads.
- **Frequentist calculation** would suggest the coin is loaded (although with low confidence)
- **Bayesian** our prior knowledge that the coin is fair allows us to maintain some degree of belief that a tails is still possible.
  - The actual mechanics of how we combine our prior belief relies on something called Bayes' rule, which will be covered later.



# Mathematical Framework

- Probability theory is a mathematical framework.
- As with any mathematical framework there is some vocabulary and important rules needed to fully leverage the theory as a tool for machine learning.

# Probability Spaces

- Probability is all about the possibility of various outcomes. The set of all possible outcomes is called the sample space.
  - e.g., sample space for coin flip is {heads, tails}.
  - e.g., the sample space for the temperature of water is all values between the freezing and boiling point.
- Only one outcome in the sample space is possible at a time, and the sample space must contain all possible values.

# Random Variables

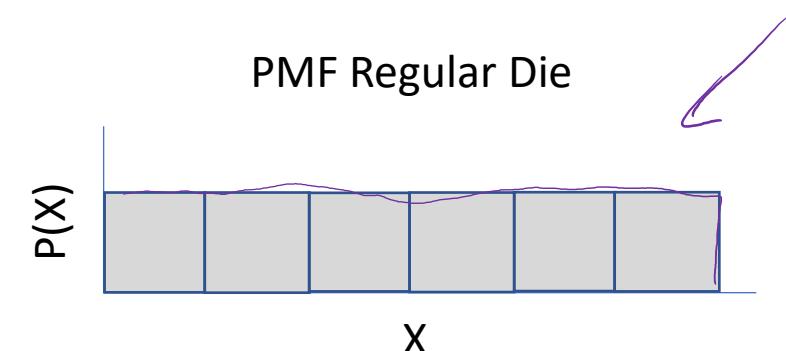
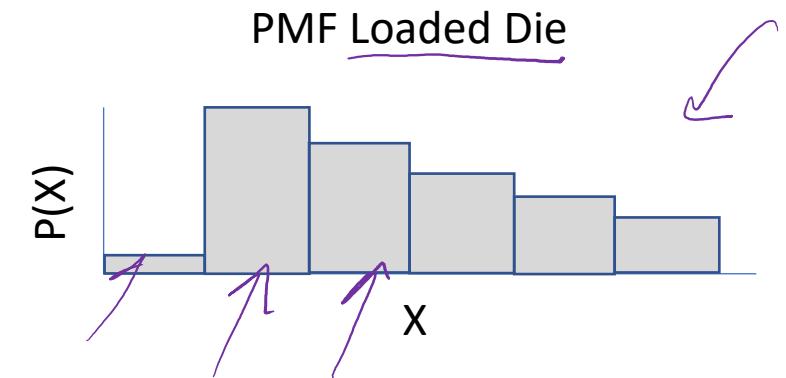
- Are variables which randomly takes on values (discrete or continuous) from a sample space.
- Probability of any event has to be between 0 (impossible) and 1 (certain), and the sum of the probabilities of all events should be 1.

$$0 \leq p(x) \leq 1$$

$$\sum_x p(x) = 1$$

# Discrete Probabilities

- Discrete random variables are described with a **probability mass function (PMF)**.
- PMF maps each value in the variable's sample space to a probability.
  - e.g., PMF for a loaded die and how does it compare with a normal die



# Probability Distributions

- A common discrete distribution is the **Bernoulli**.
  - A Bernoulli distribution specifies the probability for a random variable which can take on one of two values.
    - e.g., heads or tails
  - We can specify the entire distribution with a single parameter  $p$ , the probability of the positive outcome.
    - e.g., for a fair coin we have  $p = 0.5$ ,
    - e.g., given the probability of rain is  $p = 0.2$ , we can infer the probability of no rain is  $0.8$ .
- Other common discrete distributions are the **binomial** (e.g., handles multiple tosses of a coin) and **multinomial** distributions (e.g., rolling a die), and **Poisson** (events occurring in fixed interval of time).
  -

# Types of Probabilities

## ➤ Joint Probability

- a joint distribution over two random variables  $x, y$  specifies the probability of any setting of the random variables.

$$\underline{P(x, y)}$$

## ➤ Marginal Probability

- called the marginal probability distribution, since we've "marginalized" away the random variable  $y$  (uses the **sum rule**).

$$\underline{P(\underline{x})} = \sum_{\underline{y}} P(x, y)$$

## ➤ Conditional Probability

- the probability of an event given that another event has already been observed.

$$P(\underline{x}|y) = \frac{P(x, y)}{\underline{P(y)}}$$

# Bayes' Theorem

- Product Rule:

$$\underline{P(x, y)} = \underline{P(x|y)} \cdot P(y).$$

- We can write the product rule for two variables in two equivalent ways:

$$P(x, y) = P(y|x) \cdot P(x)$$

- By setting both equations equal and divide by  $P(y)$ , we get **Bayes' rule**:

$$P(x|y) = \frac{P(y|x) \cdot P(x)}{P(y)}$$

Note  
Bayes's rule is crucially important to much of statistics and machine learning. Driving force behind Bayesian statistics (Bayesian perspective).

This simple rule allows us to update our beliefs about quantities as we gather more observations from data.

# Bayes' Theorem Example (made-up numbers)

→ If a random person has a fever, what is the chance that it is COVID?

$$P(x|y) = \frac{P(y|x) \cdot P(x)}{P(y)}$$

We know:

- 1.  $P(\text{fever} | \text{COVID}) = 25\%$ : 25% of infected people have fever
- 2.  $P(\text{COVID}) = 2\%$  : Fraction of world population having COVID
- 3.  ~~$P(\text{fever}) = 1\%$~~  : 1 in every 100 persons has fever

$$P(\text{COVID} | \text{fever}) = \frac{0.25 \times 0.02}{0.01} = 0.5$$

# Bayes' Theorem Example (made-up numbers)

If a random person feel tired, what is the chance that it is COVID?

$$P(x|y) = \frac{P(y|x) \cdot P(x)}{P(y)}$$

We know:

1.  $P(\text{tired} | \text{COVID}) = 25\%$ : 25% of infected people feel tired
2.  $P(\text{COVID}) = 2\%$  : Fraction of world population having COVID
3.  $P(\text{tired}) = 25\%$  : 1 in every 4 persons feels tired

Find the answer and discuss over Piazza, if you notice anything interesting...

# Independence

- Two variables  $x$  and  $y$  are said to be independent if

$$P(x, y) = P(x) \cdot P(y)$$

$$\cancel{P(x,y)} \approx P(\text{only } y) P(y)$$

Q: Can you think of an example where this would happen?

$$\approx P(\text{an } x) P(y)$$

# Conditional Independence

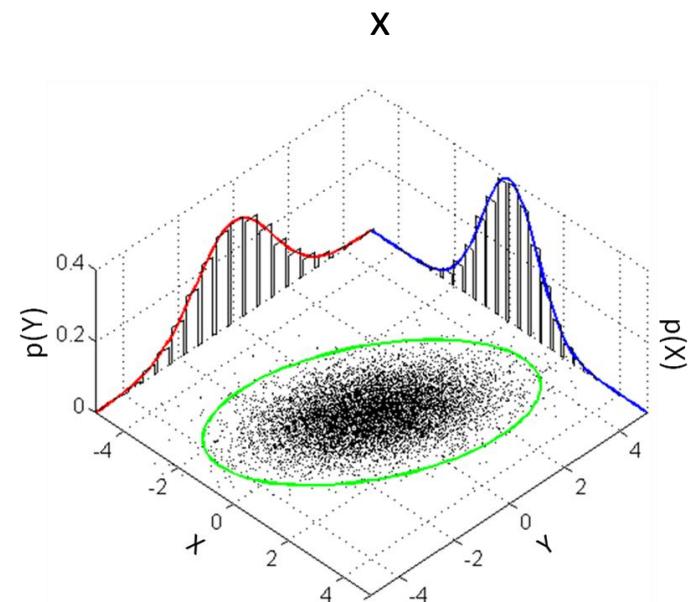
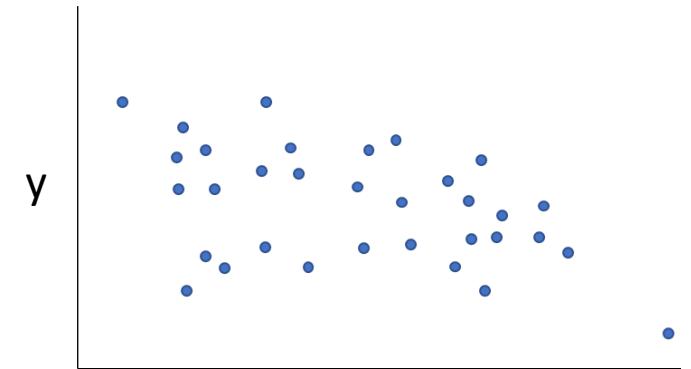
- Two variables  $x$  and  $y$  are called conditionally independent given another variable  $z$  if

$$\underbrace{P(x, y|z)}_{\text{Conditionally independent}} = P(x|z) \cdot P(y|z)$$

- Q: Can you think of an example where this would happen?

# Continuous Probabilities

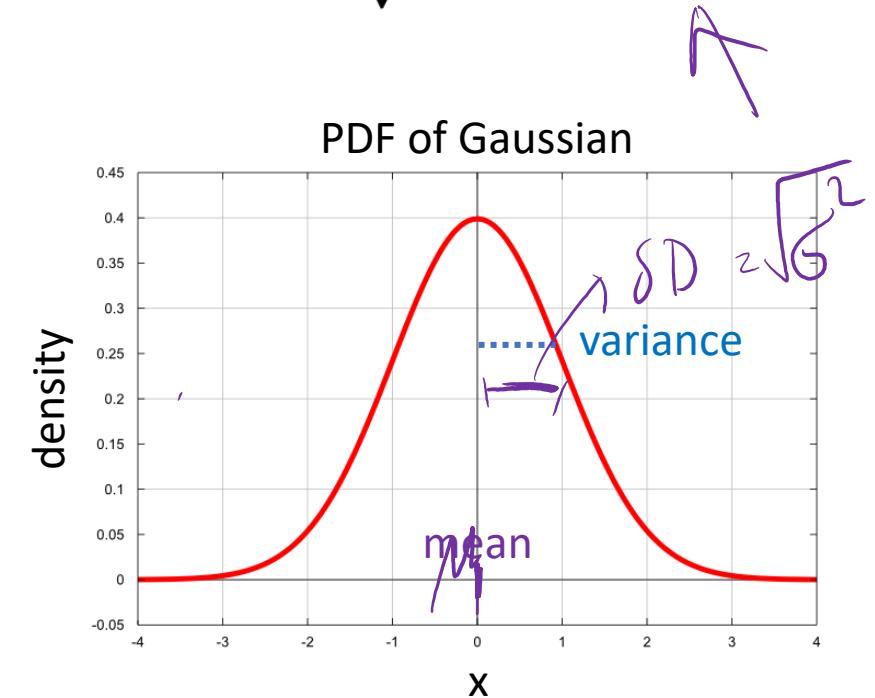
- Continuous random variables are described by **probability density functions (PDF)** which can be a bit more difficult to understand.
  - PDFs map an infinite sample space to relative likelihood values.
  - To understand this, let's look at an example with one of the most famous continuous distributions, the Gaussian (aka Normal) distribution.



# Gaussian Distribution

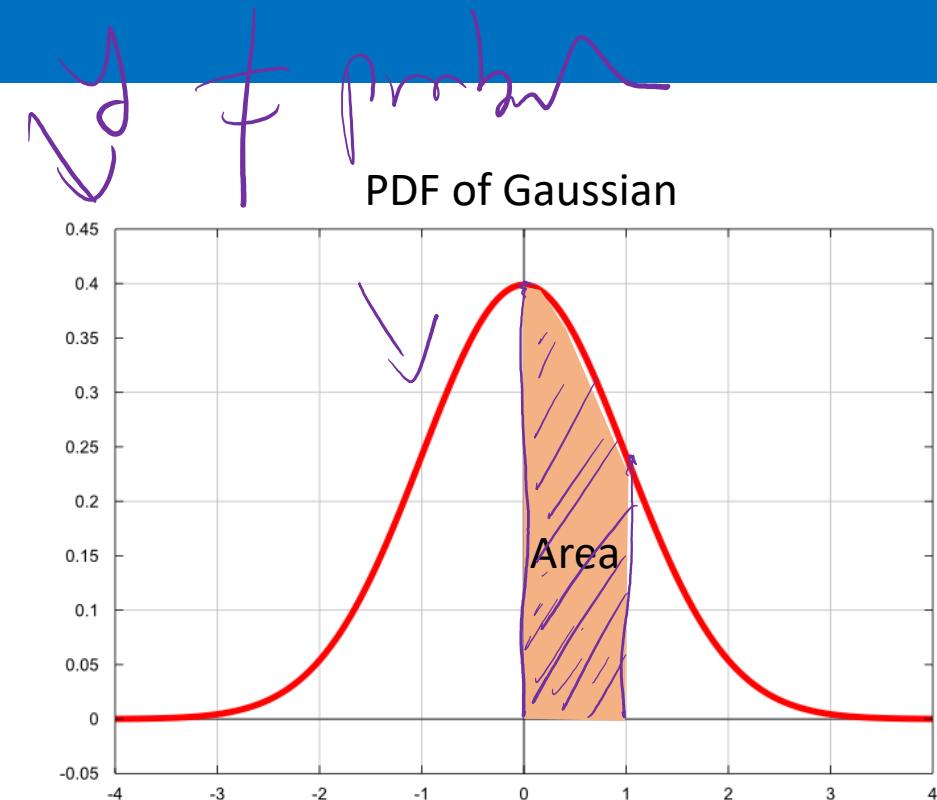
- The Gaussian distribution is parameterized by two values: the mean  $\mu$  (mu) and variance  $\sigma^2$  (sigma squared).
- The **mean specifies the center** of the distribution, and the **variance specifies the width** of the distribution.
- You may have also heard about the standard deviation  $\sigma$ , which is just the square root of the variance.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



# Relative Likelihood

- The value of the PDF is not the actual probability of  $x$ .
- Remember, the **total probability for every possible value needs to sum to 1.**
- Q: How can we sum over infinite number of values?
- A: Need to calculate the area under the PDF to obtain the probability

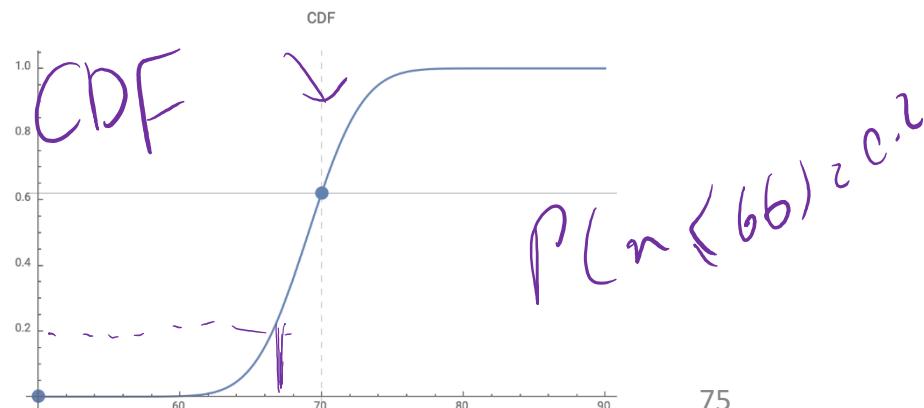
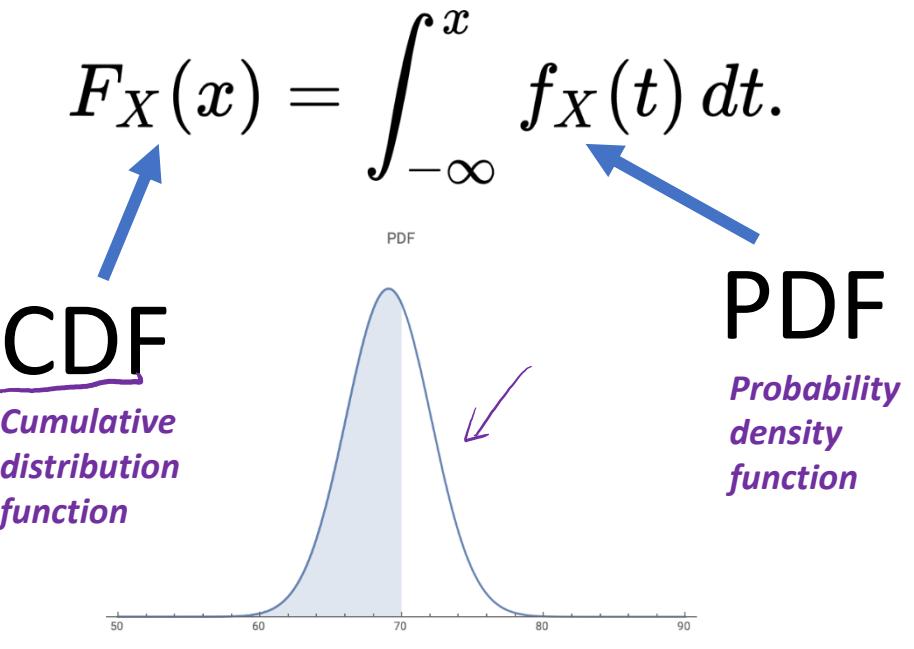


$$p(0 < \underline{x} < 1) = \int_0^1 f(x) dx$$

Since we are interested in the area, it is often more useful to work with a continuous random variable's cumulative density function (CDF).

# Relative Likelihood Con't

- We just determined that the area corresponds to the probability, so  $F(x)$  gives us  $P(X \leq x)$ .
- Use the CDF to determine the **probability of any given range**  $[a, b]$  using  $P(a \leq X \leq b) = F(b) - F(a)$ .
- Note that asking for  $P(X=x)$  is equivalent to asking  $P(x \leq X \leq x) = F(x) - F(x) = 0$



# Functions of Random Variables

- Often useful to create functions which take random variables as input.
- e.g., it costs \$2 to play the game, “guess a number between 1 and 10”.  
Correct guess = \$10, Incorrect guess = \$0, but it costs \$2 to play. Let  $x$  be a random variable indicating whether you guessed correctly. We can write a function:

$$h(x) = \{ \$8 \text{ if } x = 1, \text{ and } -\$2 \text{ if } x = 0 \}$$

- You may be interested in knowing in advance what the expected outcome will be.

# Expectation

- The expected value, or expectation, of a function  $h(x)$  on a random variable  $x \sim P(x)$  is the average value of  $h(x)$  weighted by  $P(x)$ . For a discrete  $x$ , we write this as:

$$\mathbb{E}[h(x)] = \sum_x P(x) \cdot h(x)$$

If  $x$  had been continuous, we would replace the summation with an integral

- The expectation acts as a weighted average over  $h(x)$ , where the weights are the probabilities of each  $x$ .

e.g.,  $\mathbb{E}[h(x)] = P(\text{winning}) \cdot h(\text{winning}) + P(\text{losing}) \cdot h(\text{losing})$   
 $= (1/10) \cdot \$8 + (9/10) \cdot (-\$2) = \$0.80 + (-\$1.80)$   
 $= -\$1$

On average, we'll lose \$1 every time we play!

# Expectation

- A nice property of expectations is that they're linear.
- Let's assume  $h$  and  $g$  are functions of  $x$ , and  $\alpha$  and  $\beta$  are constants. Then we have:

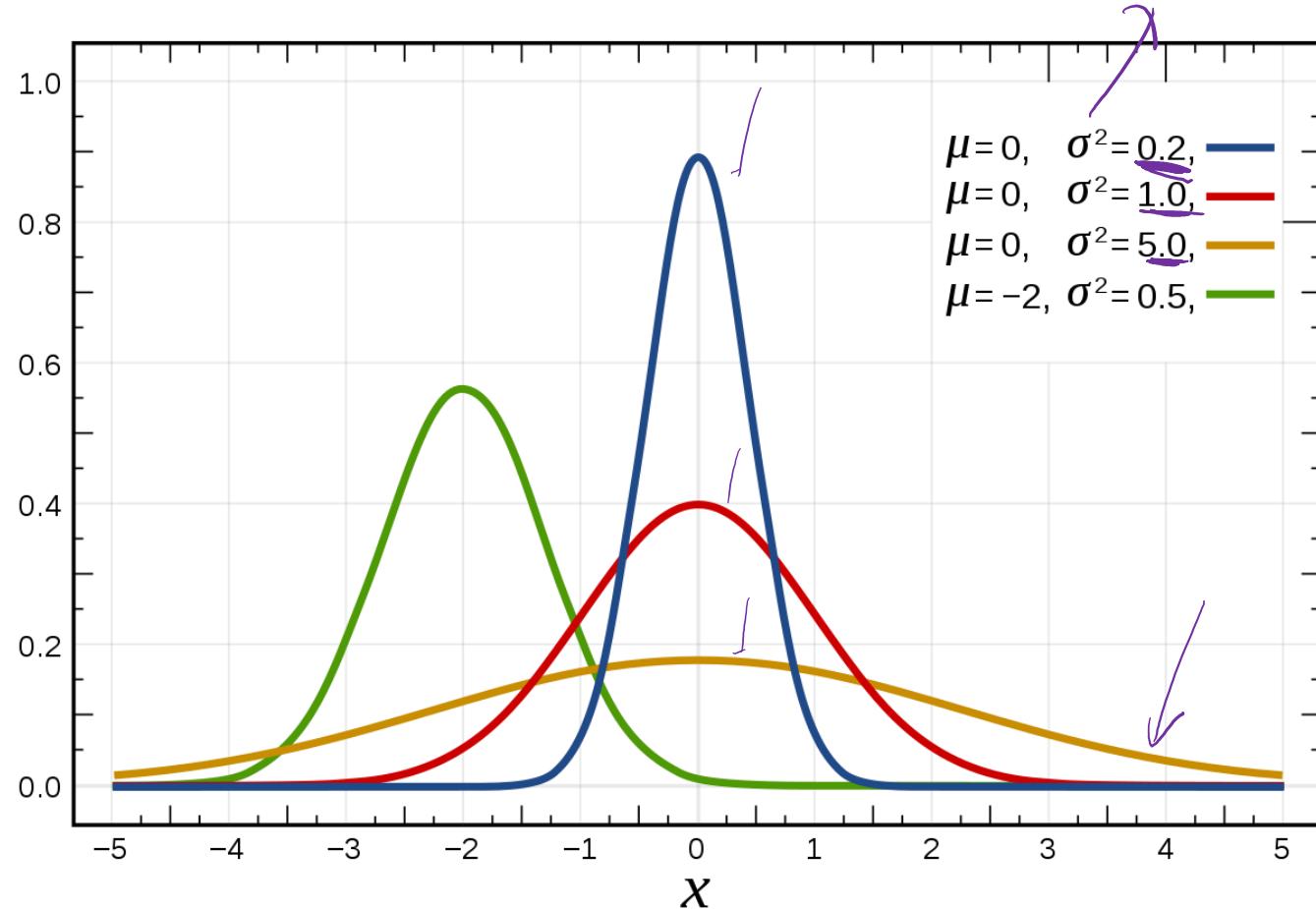
$$\mathbb{E}[\underbrace{\alpha h(x) + \beta g(x)}_{\text{linearity}}] = \alpha \mathbb{E}[h(x)] + \beta \mathbb{E}[g(x)]$$

# Variance

- We saw variance with respect to a Gaussian distribution when we were talking about continuous random variables.  
In general, **variance is a measure of how much random values vary from their mean.**
- Similarly, for functions of random variables, the variance is a measure of the **variability of the function's output from its expected value.**

$$\text{Var}(h(x)) = \mathbb{E}[(h(x) - \mathbb{E}[h(x)])^2]$$

# Describing a Gaussian Distribution



A hand-drawn graph illustrating the normal distribution of data. The horizontal axis is labeled  $\mu \pm 3\sigma$ . Two bell-shaped curves are shown. The first curve, labeled "Blue", is centered at  $\mu$  and has its peak marked with a vertical arrow and labeled "99.7%". The second curve, labeled "Yellow", is shifted to the right and has its peak marked with a vertical arrow and labeled "95%".

# Multivariate Data

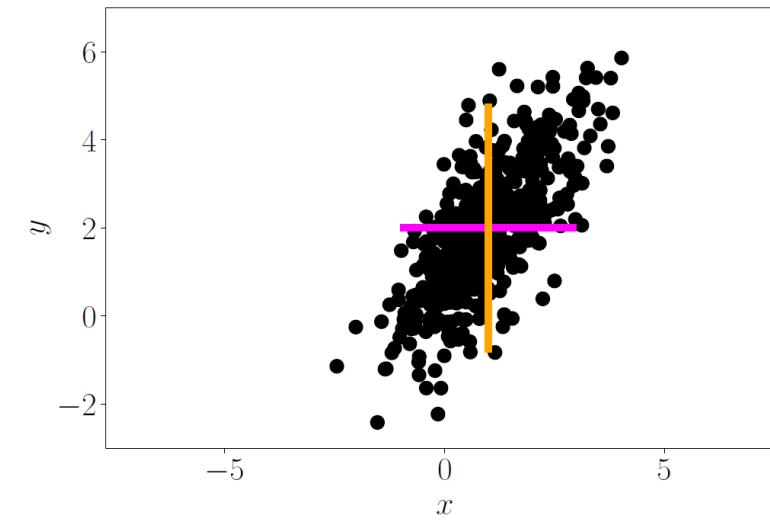
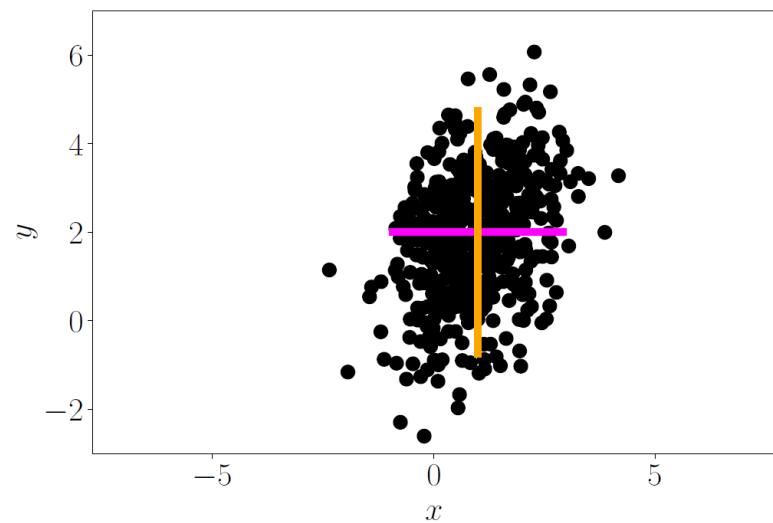
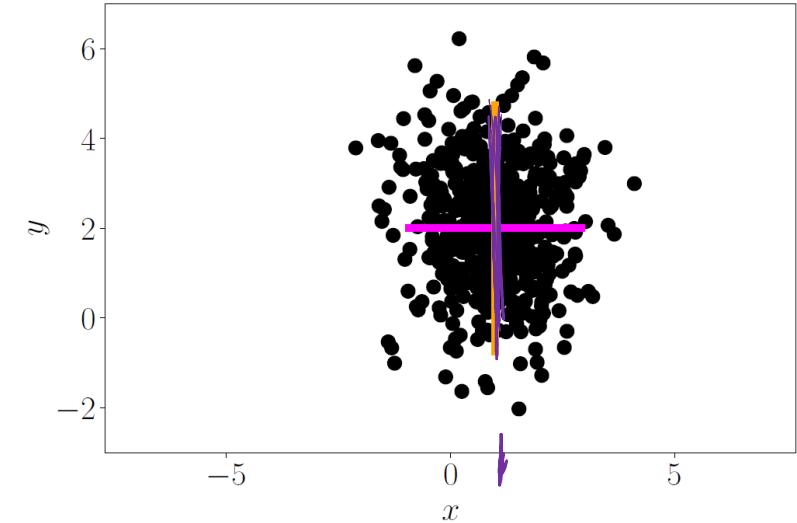
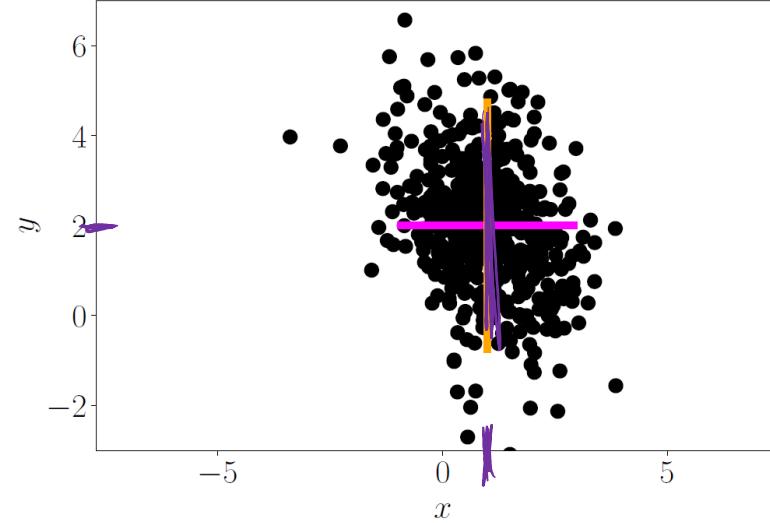
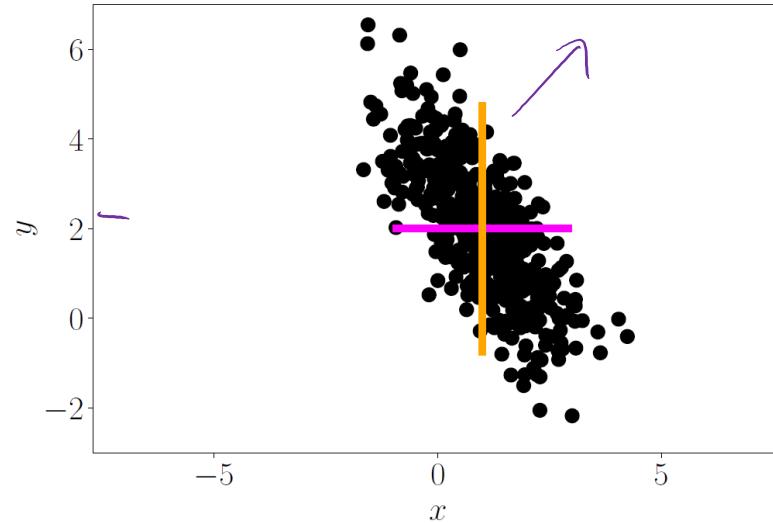
- Multiple measurements (e.g., different sensors) ↗
- $d$  features/attributes (e.g., number of sensors) ↗
- $N$  instances/observations/examples

$$\mathbf{X} = \begin{bmatrix} | & | & | \\ x_1^{(1)} & x_2^{(1)} & \dots & x_d^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_d^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \dots & x_d^{(N)} \end{bmatrix}$$

rows => instances

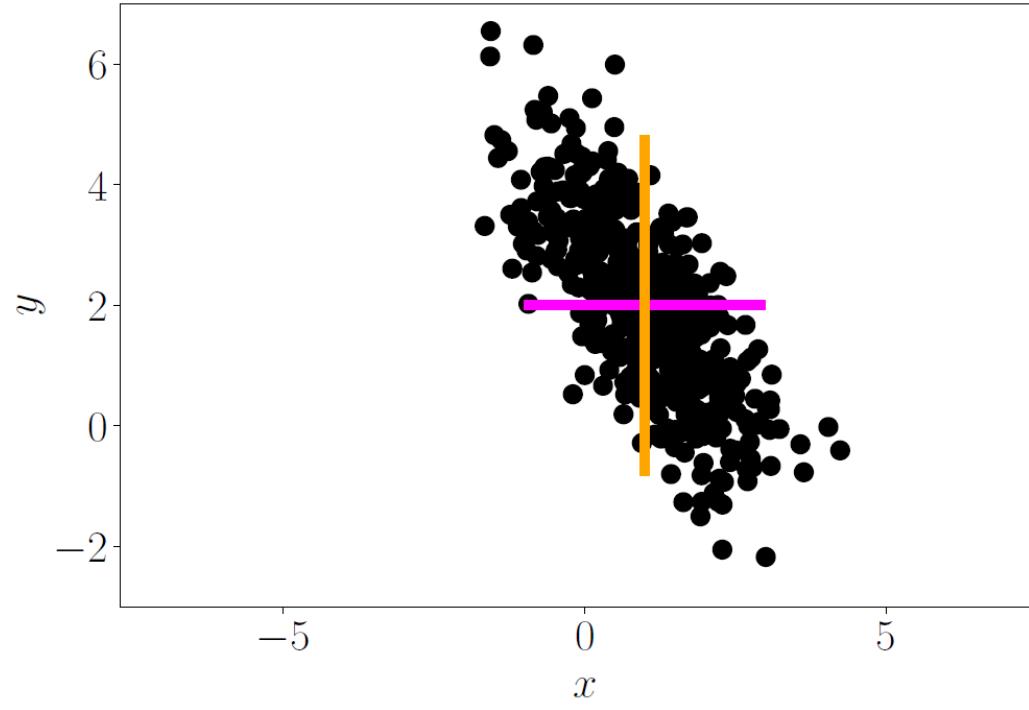
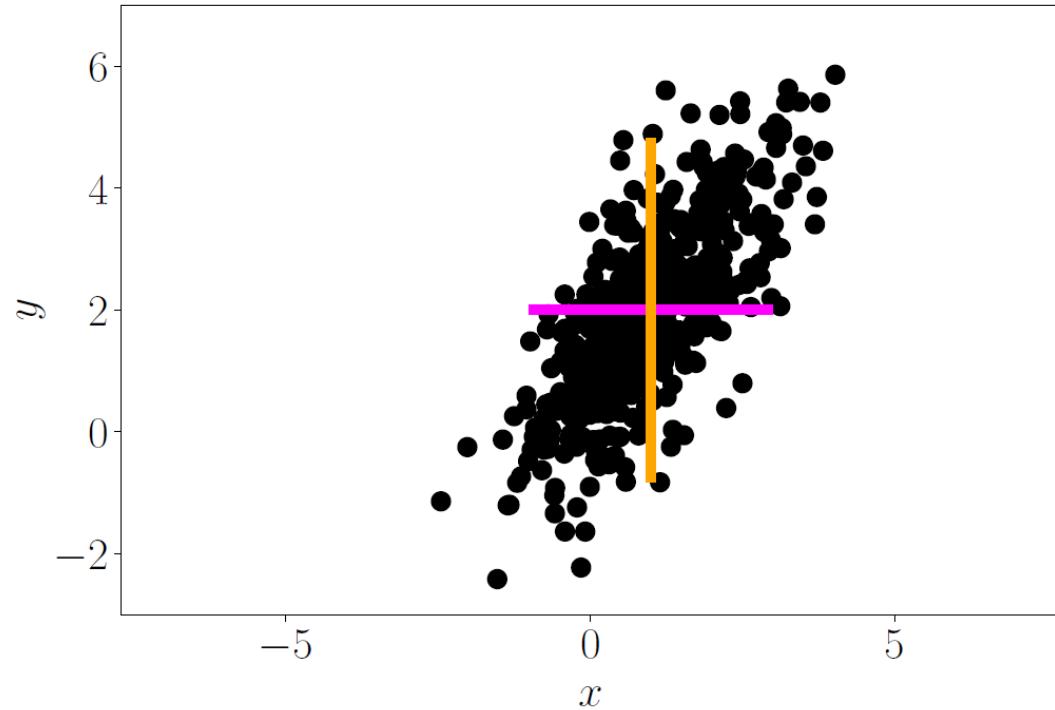
columns => features for a single instance

# Describing Multivariate Datasets



➤ What do all these graphs have in common?

# Covariance



- Variances along each axis remain constant, but properties of the dataset change
- Variances insufficient to characterize the relationship/correlation of two random variables -> **we need cross-variance!**

# Covariance Matrix

- **Expectation (mean):** centre of the dataset

$$\mathbb{E}[\mathbf{x}] = [\mu_1, \dots, \mu_d]^T$$

*Cov ( $\mu_1, \mu_2$ )*

- **Covariance:** “variance” of a d-dimensional random variable is given by a covariance matrix.

$$\Sigma = \text{Cov}(\mathbf{x}) = \mathbb{E}[(\mathbf{x} - \mu)^T (\mathbf{x} - \mu)] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{bmatrix}$$

$$\Rightarrow \boxed{\text{cov}(X, Y) = \mathbb{E} [(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]}$$

# Multivariate Gaussian Distribution

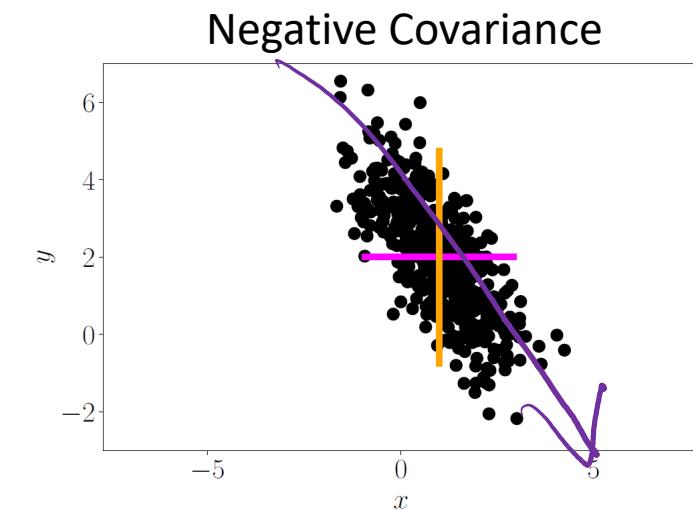
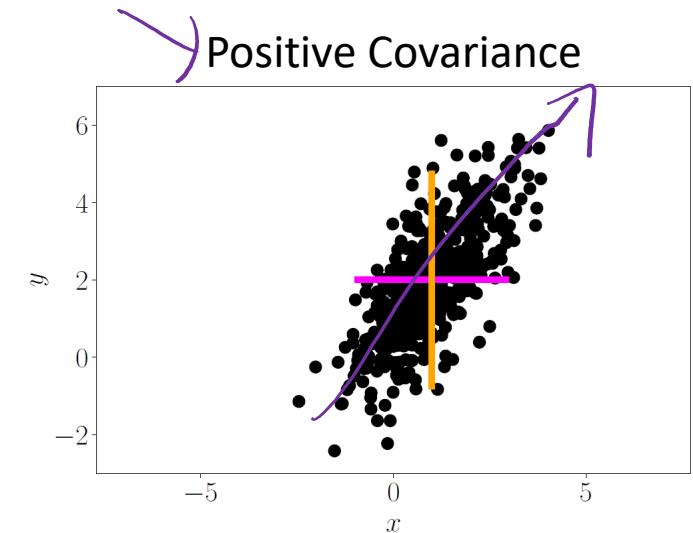

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left[ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]$$

Determinant of Covariance Matrix      Multivariate Sample      Covariance Matrix      Multivariate Mean

Blue arrows point from the labels below to the corresponding terms in the equation: the first arrow points to  $(2\pi)^{d/2}$ , the second to  $\mathbf{x}$ , the third to  $\Sigma^{-1}$ , and the fourth to  $\boldsymbol{\mu}$ .

# Covariance

- When the **absolute value of covariance is high**, the two variables tend to vary far from their means at the same time.
- When the **sign of the covariance is positive**, the two functions map to higher values together.
- When the **sign of the covariance is negative**, the one function maps to higher values, the other maps to lower values (or vice versa)



# Bivariate Normal

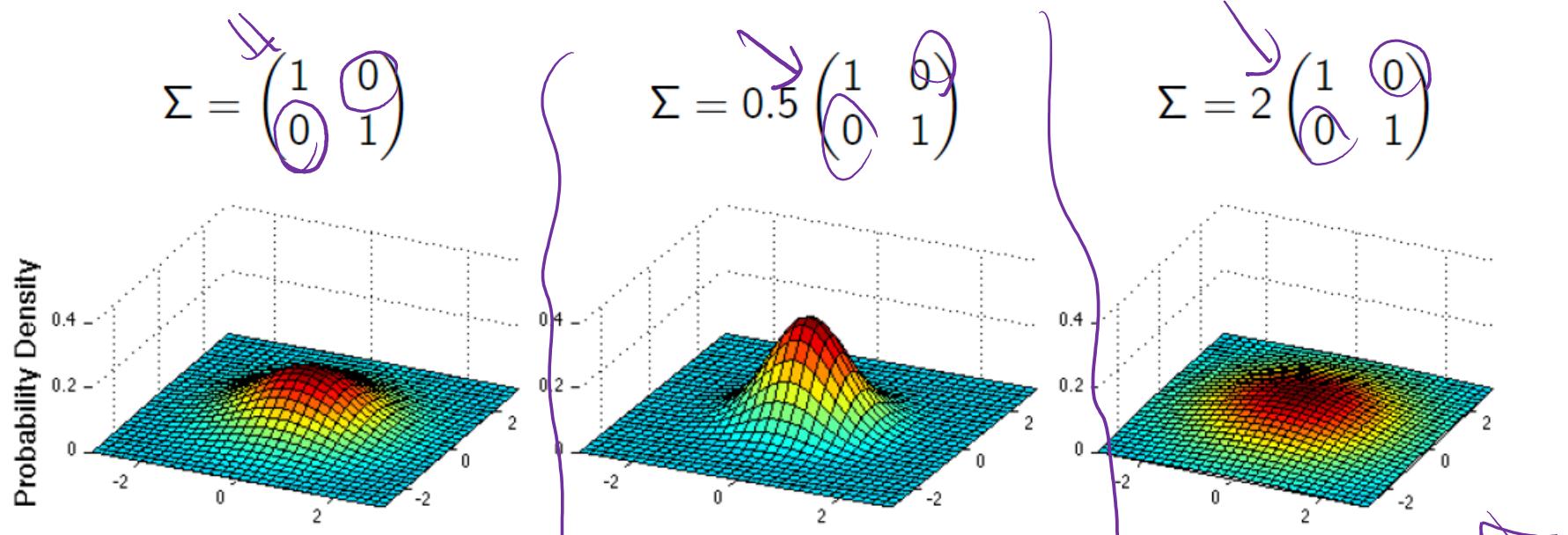


Figure: Probability density function

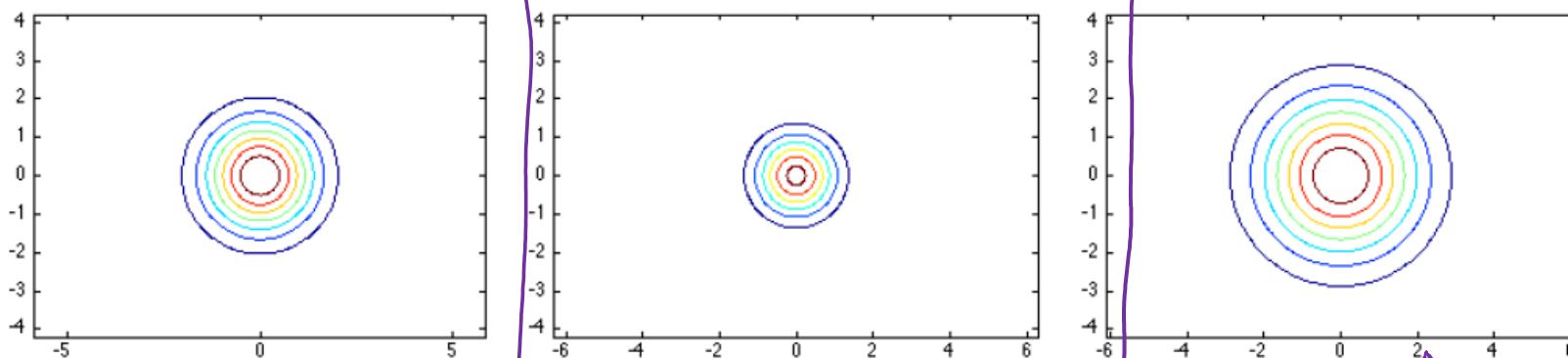


Figure: Contour plot of the pdf

# Bivariate Normal

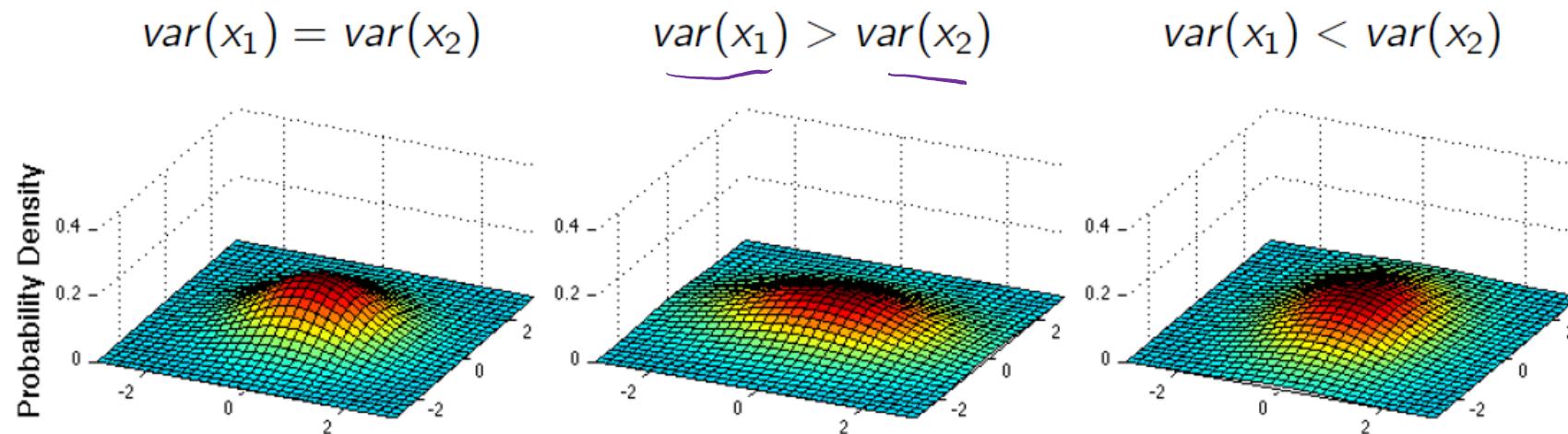


Figure: Probability density function

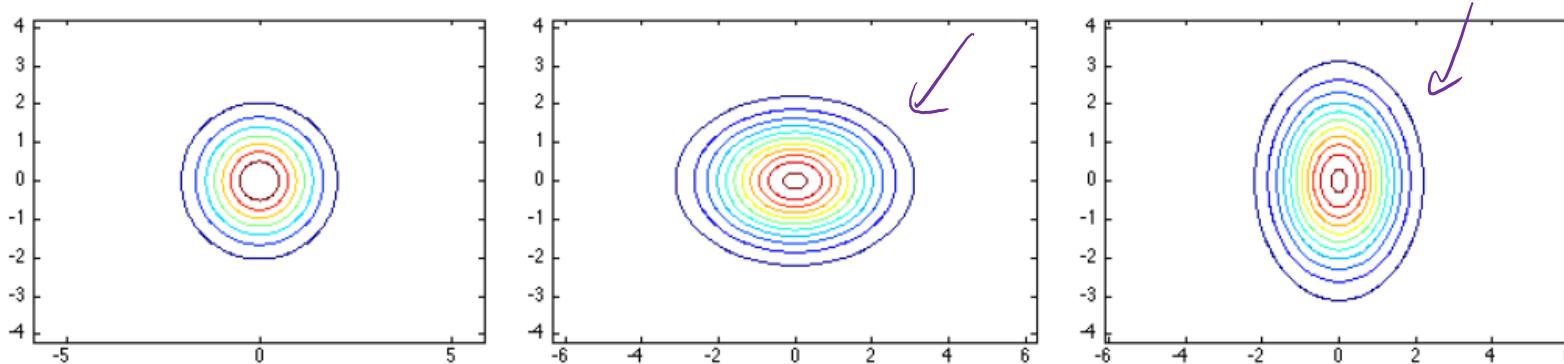


Figure: Contour plot of the pdf

# Bivariate Normal

$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$

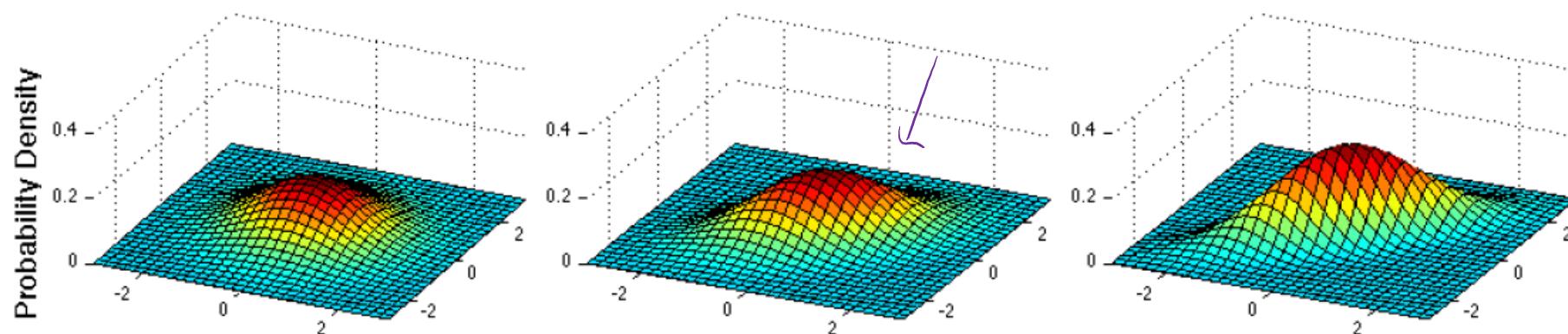


Figure: Probability density function

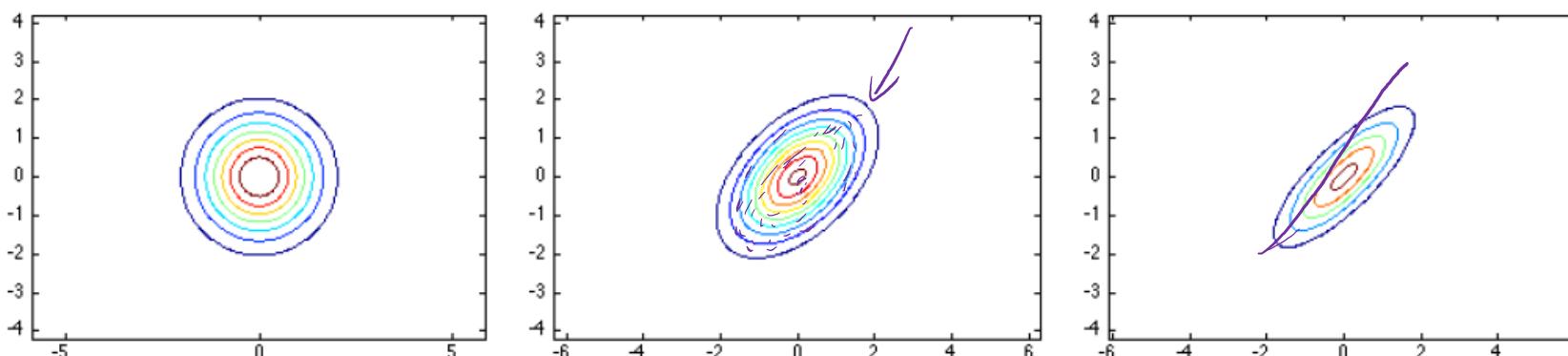


Figure: Contour plot of the pdf

# Bivariate Normal

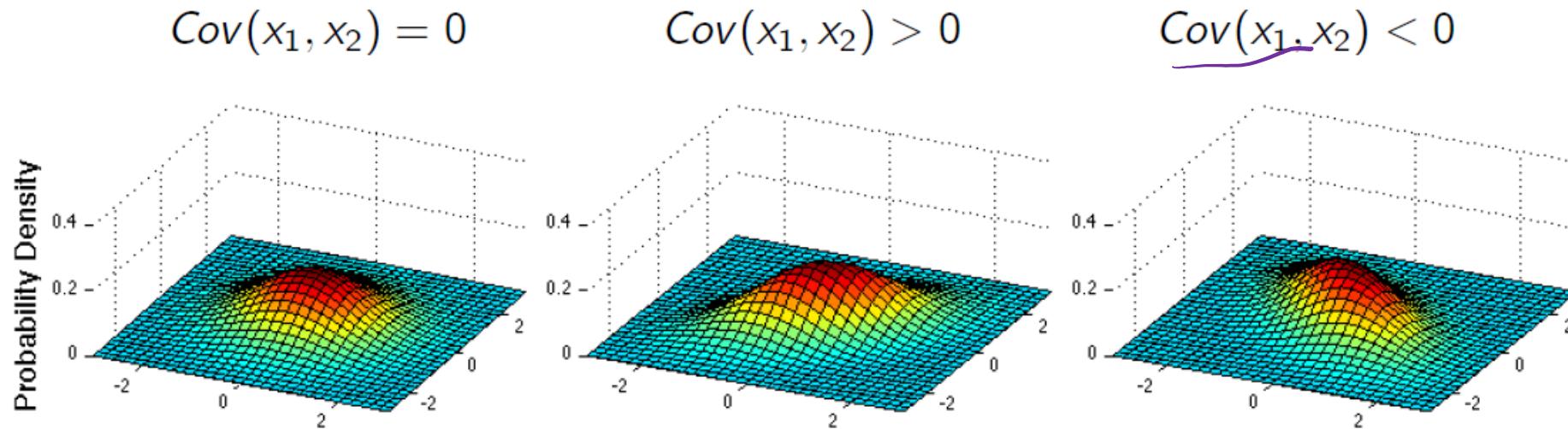
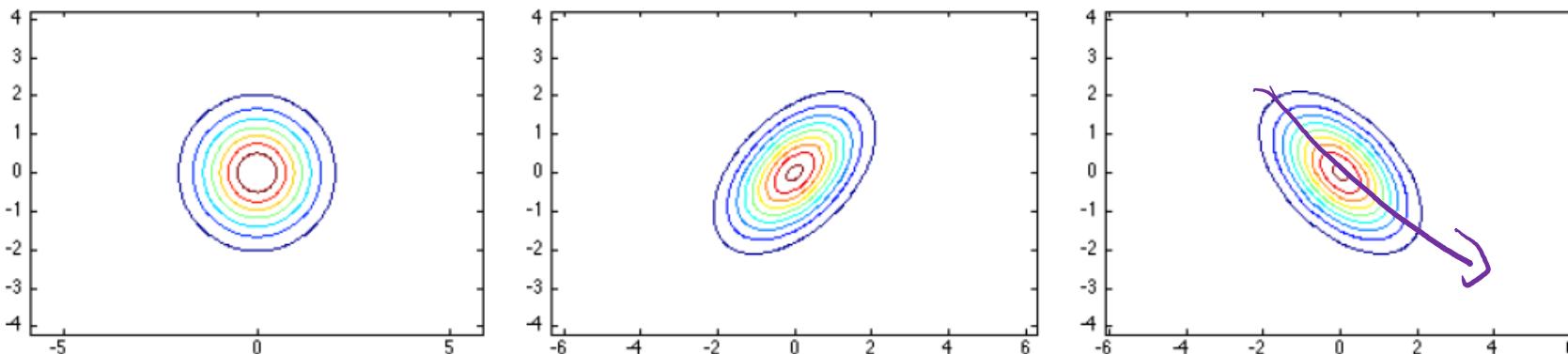
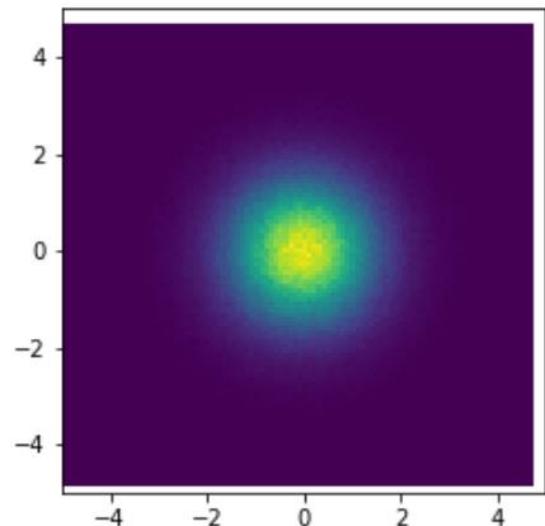
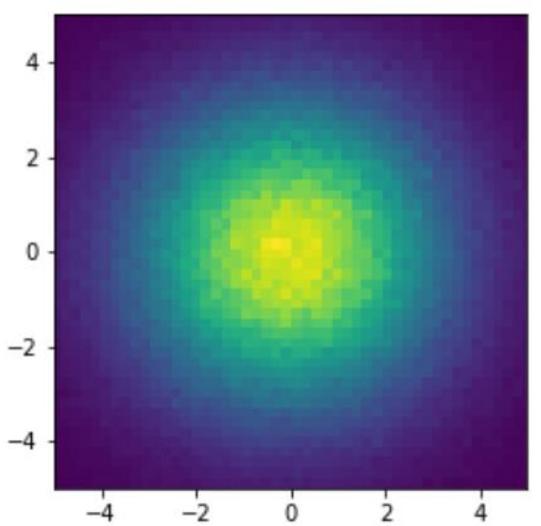


Figure: Probability density function

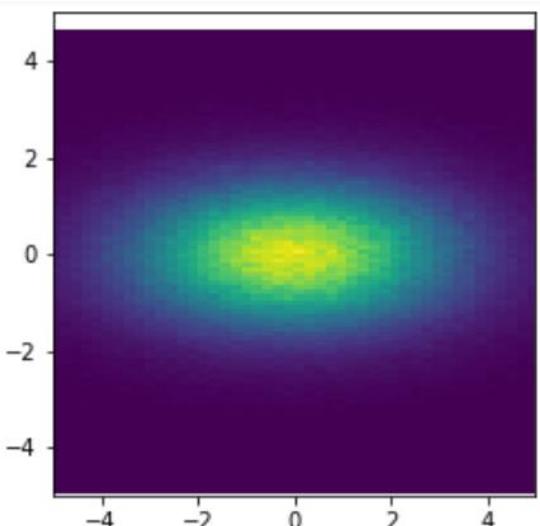




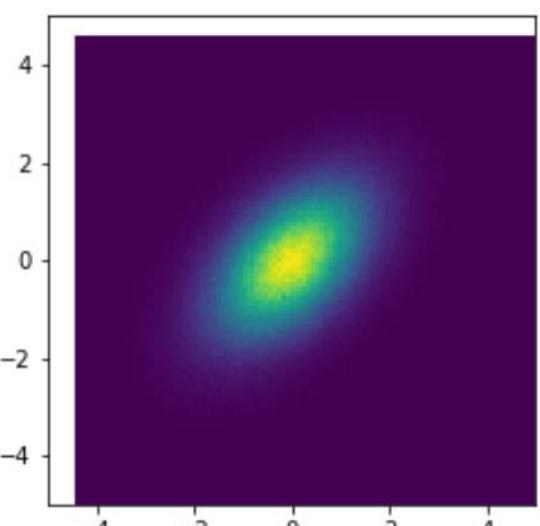
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$   
0.0



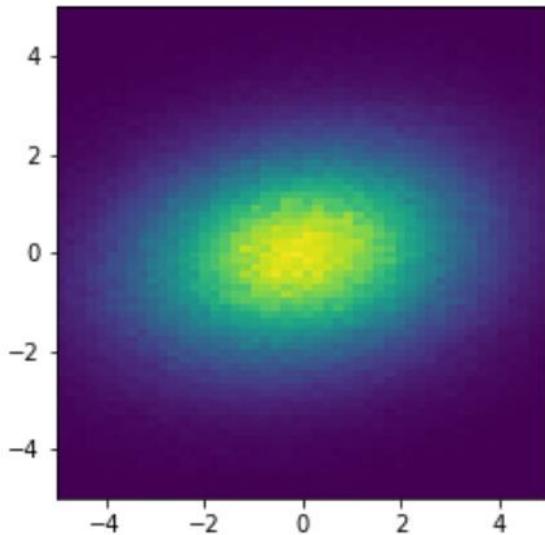
$\begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$   
0.0



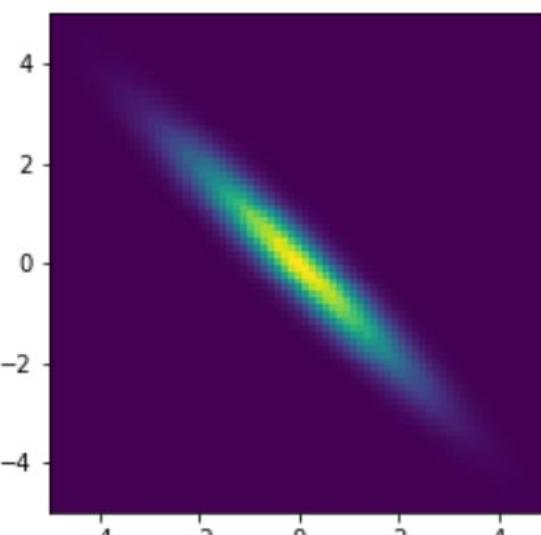
$\begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}$   
0.0



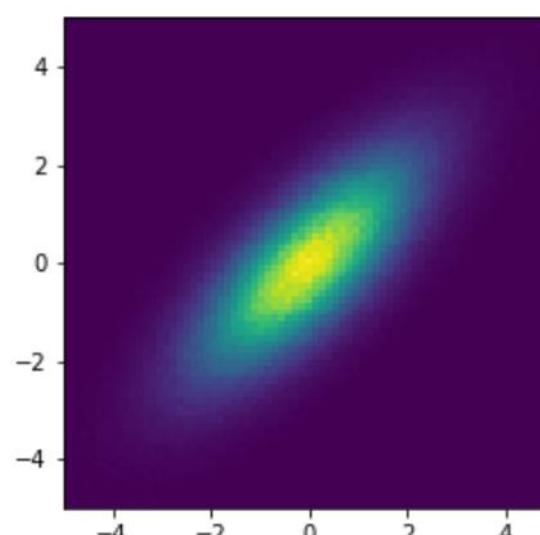
$\begin{bmatrix} 1. & 0.5 \\ 0.5 & 1. \end{bmatrix}$   
0.5



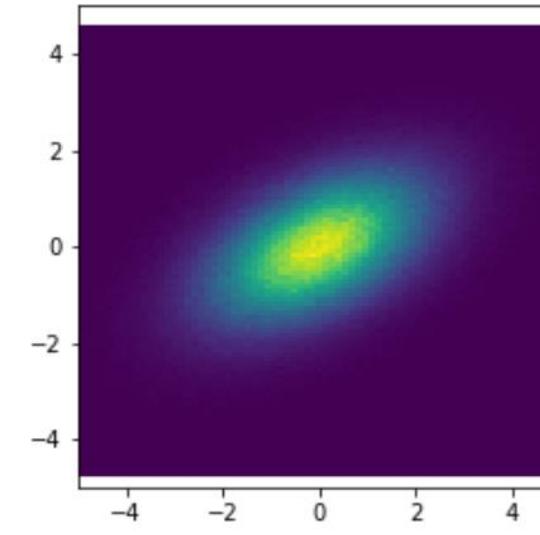
$\begin{bmatrix} 5. & 0.5 \\ 0.5 & 2. \end{bmatrix}$   
0.15811388300841897



$\begin{bmatrix} 2. & -1.9 \\ -1.9 & 2. \end{bmatrix}$   
-0.9499999999999997



$\begin{bmatrix} 2. & 1.5 \\ 1.5 & 2. \end{bmatrix}$   
0.7499999999999999

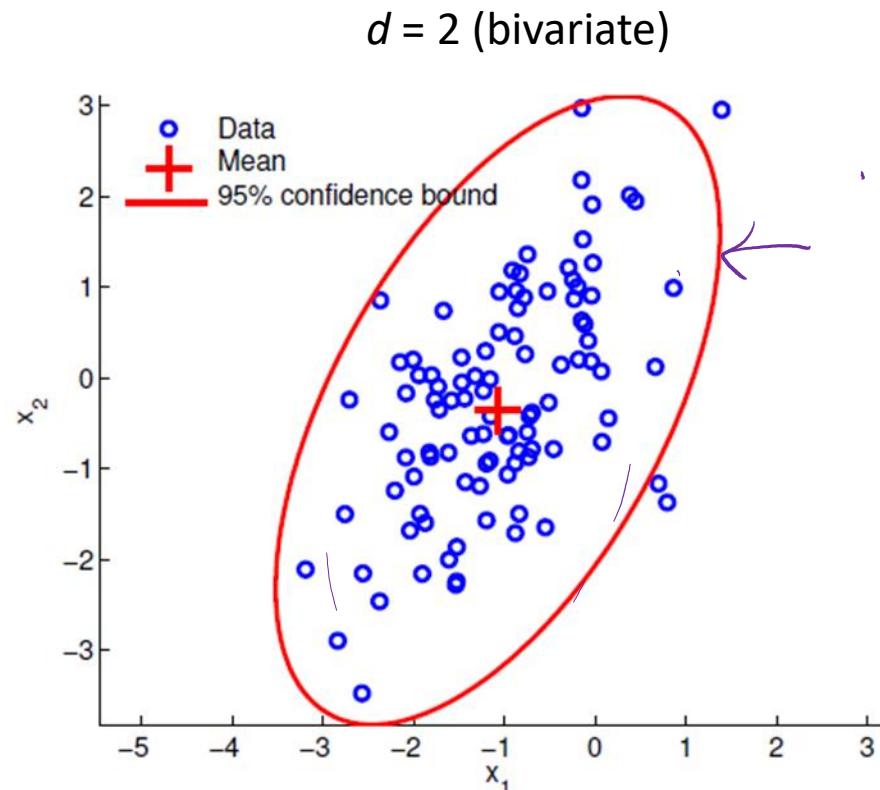
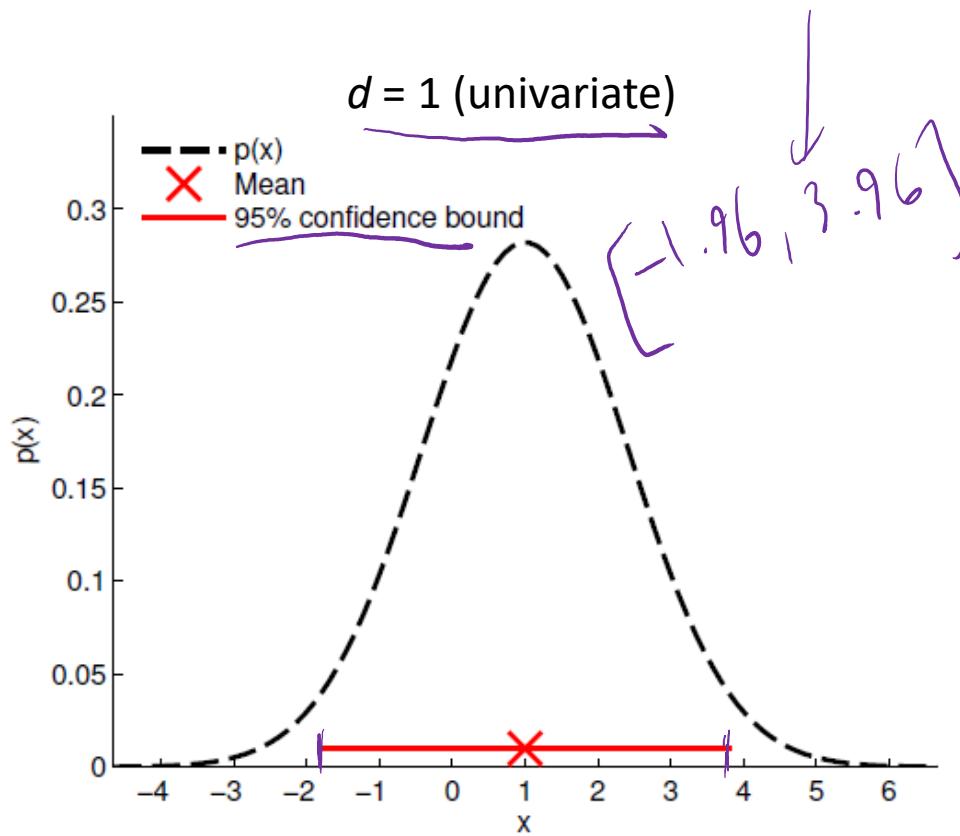


$\begin{bmatrix} 2. & 0.7 \\ 0.7 & 1. \end{bmatrix}$   
0.4949747468305832

# Confidence

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp \left[ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]$$

- Mean vector  $\boldsymbol{\mu}$  ► Average of the data
- Covariance matrix  $\boldsymbol{\Sigma}$  ► Spread of the data



# Example: Calculate the Covariance Matrix

Joint Probability  $P(X_1, X_2)$

		$X_2$		$P(X_1)$
		0	1	
$X_1$	-1	0.24	0.06	0.3
	0	0.16	0.14	0.3
$P(X_2)$	1	0.40	0	0.4
		0.8	0.2	1

compute the mean

$$\vec{\Sigma} = \begin{bmatrix} E_x(X_1 - \mu_1)^2 & E_x(X_1 - \mu_1)(X_2 - \mu_2) \\ E_x(X_2 - \mu_2)(X_1 - \mu_1) & E_x(X_2 - \mu_2)^2 \end{bmatrix}$$

$$\mu_1 = E_x(X_1) = \sum_{\text{all } X_1} x_1 p_1(x_1)$$

$$(-1)(0.3) + (0)(0.3) + (1)(0.4) = 0.1$$

$$\mu_2 = E_x(X_2) = \sum_{\text{all } X_2} x_2 p_2(x_2)$$

$$(0)(0.8) + (1)(0.2) = 0.2$$

# Example: Calculate the Covariance Matrix

		X <sub>2</sub>		P(X <sub>1</sub> )
		0	1	
X <sub>1</sub>	-1	0.24	0.06	0.3
	0	0.16	0.14	0.3
	1	0.40	0	0.4
P(X <sub>2</sub> )		0.8	0.2	1

compute the covariance

$$\vec{\Sigma} = \begin{bmatrix} E_x(X_1 - \mu_1)^2 & E_x(X_1 - \mu_1)(X_2 - \mu_2) \\ E_x(X_2 - \mu_2)(X_1 - \mu_1) & E_x(X_2 - \mu_2)^2 \end{bmatrix}$$

$$\sigma_{11} = E_x(X_1 - \mu_1)^2 = \sum_{\text{all } X_1} (X_1 - 0.1)^2 p_1(X_1)$$

$$(-1-0.1)^2(0.3) + (0-0.1)^2(0.3) + (1-0.1)^2(0.4) = 0.69$$

$$\sigma_{22} = E_x(X_2 - \mu_2)^2 = \sum_{\text{all } X_2} (X_2 - 0.2)^2 p_2(X_2)$$

$$(0-0.2)^2(0.8) + (1-0.2)^2(0.2) = 0.16$$

$$\sigma_{12} = E_x(X_1 - \mu_1)(X_2 - \mu_2)$$

$$= \sum_{\text{all pairs}} (X_1 - 0.1)(X_2 - 0.2) p_{12}(X_1, X_2)$$

$$(-1-0.1)(0-0.2)(0.24) + (-1-0.1)(1-0.2)(0.06) + (0-0.1)(0-0.2)(0.16)$$

$$+ (0-0.1)(1-0.2)(0.14) + (1-0.1)(0-0.2)(0.40) + (1-0.1)(1-0.2)(0) = -0.08$$

# Example: Calculate the Covariance Matrix

Joint Probability $P(X_1, X_2)$		$X_2$		$P(X_1)$
		0	1	
$X_1$	-1	0.24	0.06	0.3
	0	0.16	0.14	0.3
	1	0.40	0	0.4
$P(X_2)$		0.8	0.2	1

$$\vec{\Sigma} = \begin{bmatrix} E_x(X_1 - \mu_1)^2 & E_x(X_1 - \mu_1)(X_2 - \mu_2) \\ E_x(X_2 - \mu_2)(X_1 - \mu_1) & E_x(X_2 - \mu_2)^2 \end{bmatrix}$$

putting it all together

$$\vec{\Sigma} = \begin{bmatrix} 0.69 & -0.08 \\ -0.08 & 0.16 \end{bmatrix}$$

Covariance matrix

# Example: Calculate the Covariance Matrix

Joint Probability $P(X_1, X_2)$		$X_2$		$P(X_1)$
		0	1	
$X_1$	-1	0.24	0.06	0.3
	0	0.16	0.14	0.3
	1	0.40	0	0.4
$P(X_2)$		0.8	0.2	1

$$\sigma_{21} = \sigma_{12}$$
$$\Sigma = \begin{bmatrix} 0.69 & -0.08 \\ -0.08 & 0.16 \end{bmatrix}$$

Covariance matrix



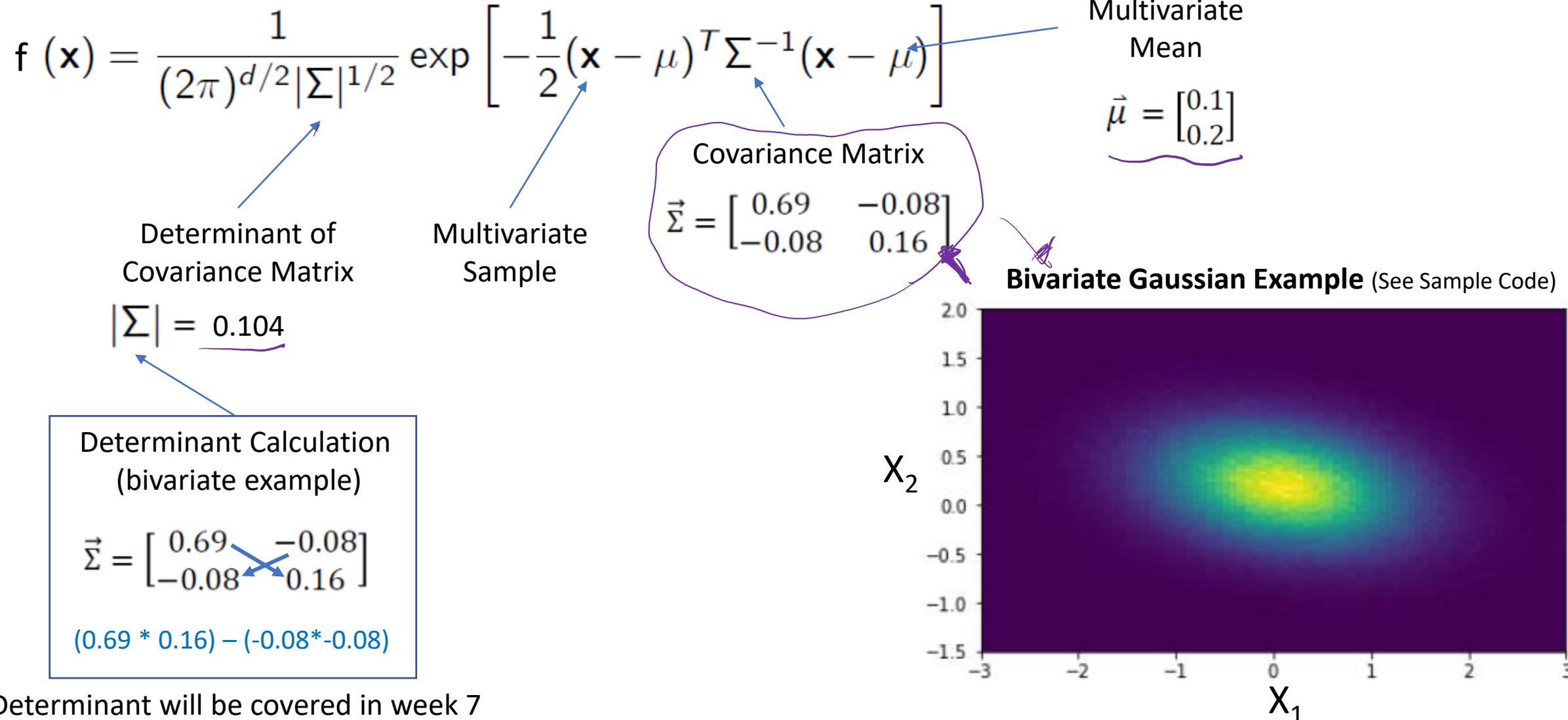
$$\rho_{ik} = \frac{\sigma_{ik}}{\sqrt{\sigma_{ii}\sigma_{kk}}}$$
$$\rho_{12} = \frac{\sigma_{12}}{\sqrt{\sigma_{11}\sigma_{22}}} = -0.24$$

Population correlation coefficient

➤ Normalizing the covariance matrix will give you the population correlation coefficient, a measure of how correlated two variables are.

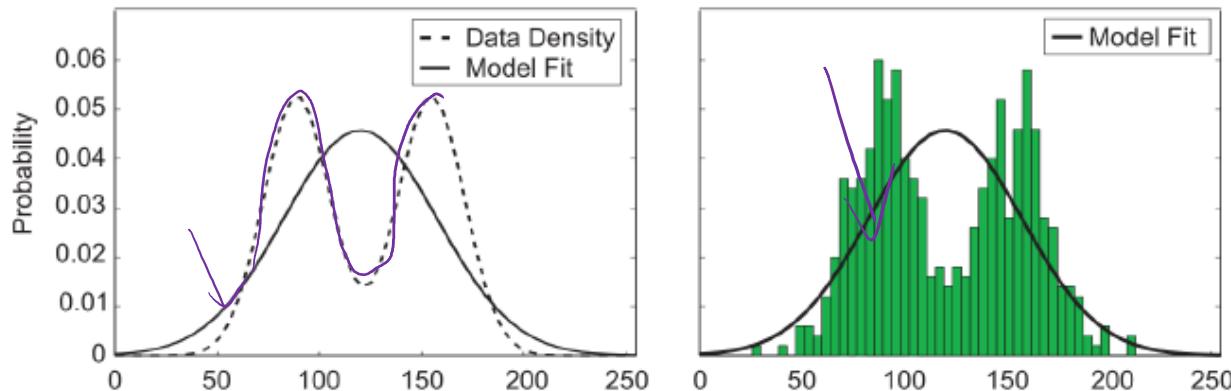
(-1, 1)

# Example: Visualization



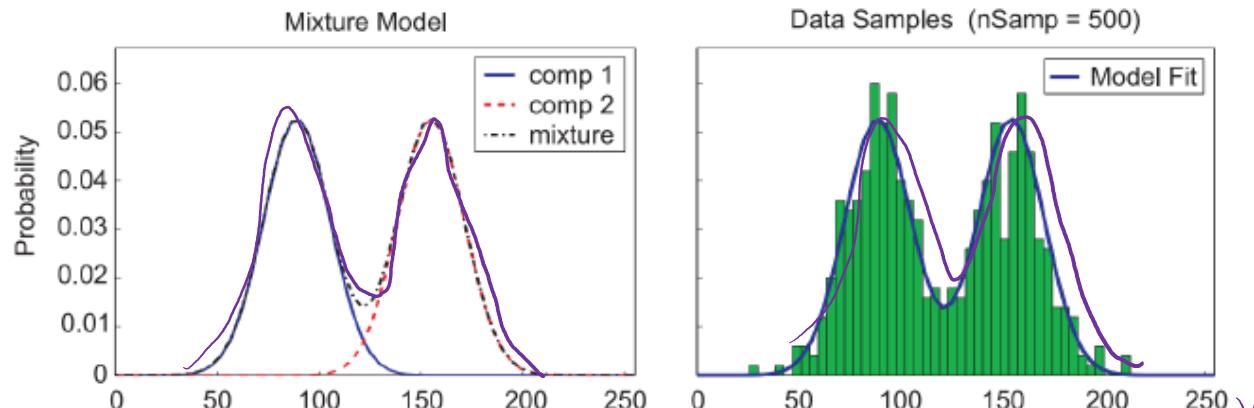
# Gaussian Mixture Models (GMM)

- If you fit a Gaussian to data:



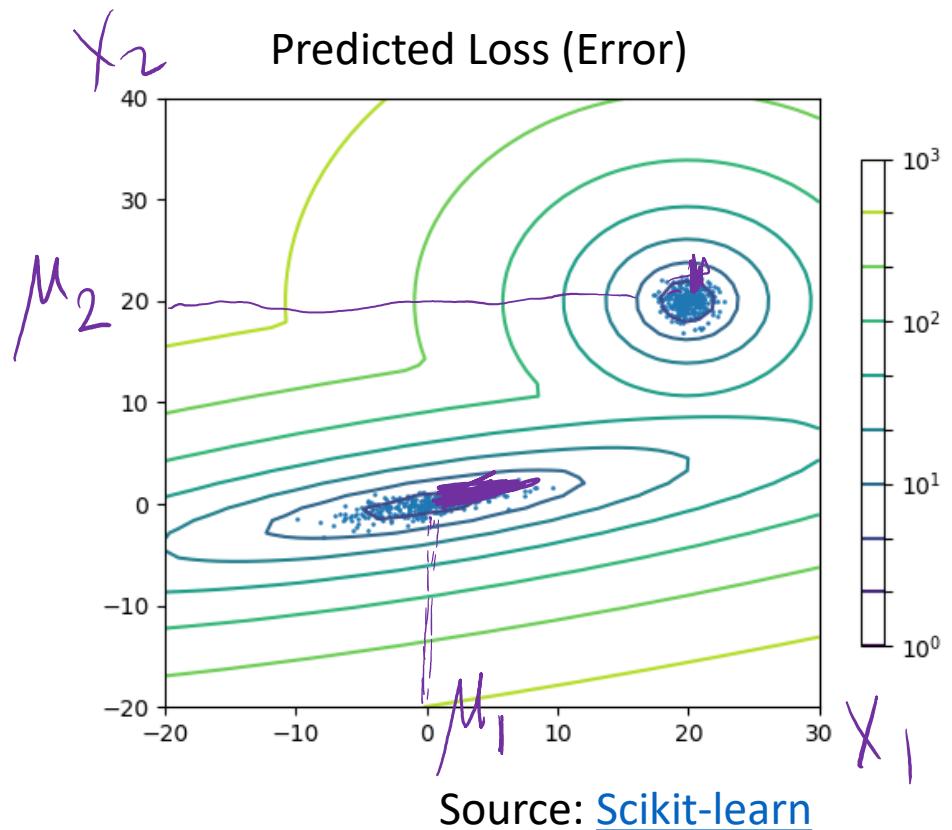
Univariate data  
multiple comp

- Now, we are trying to fit a GMM (with  $K = 2$  in this example):



[Slide credit: K. Kutulakos]

# Gaussian Mixture Models (GMM)

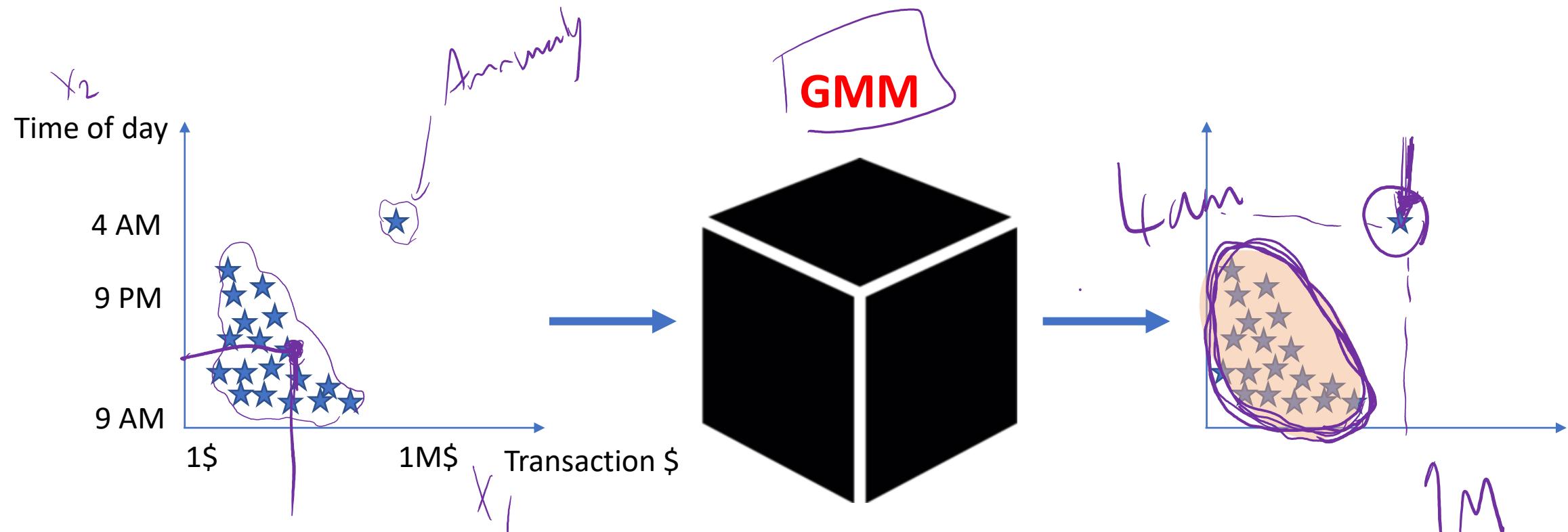


multivariate data  
multiple components

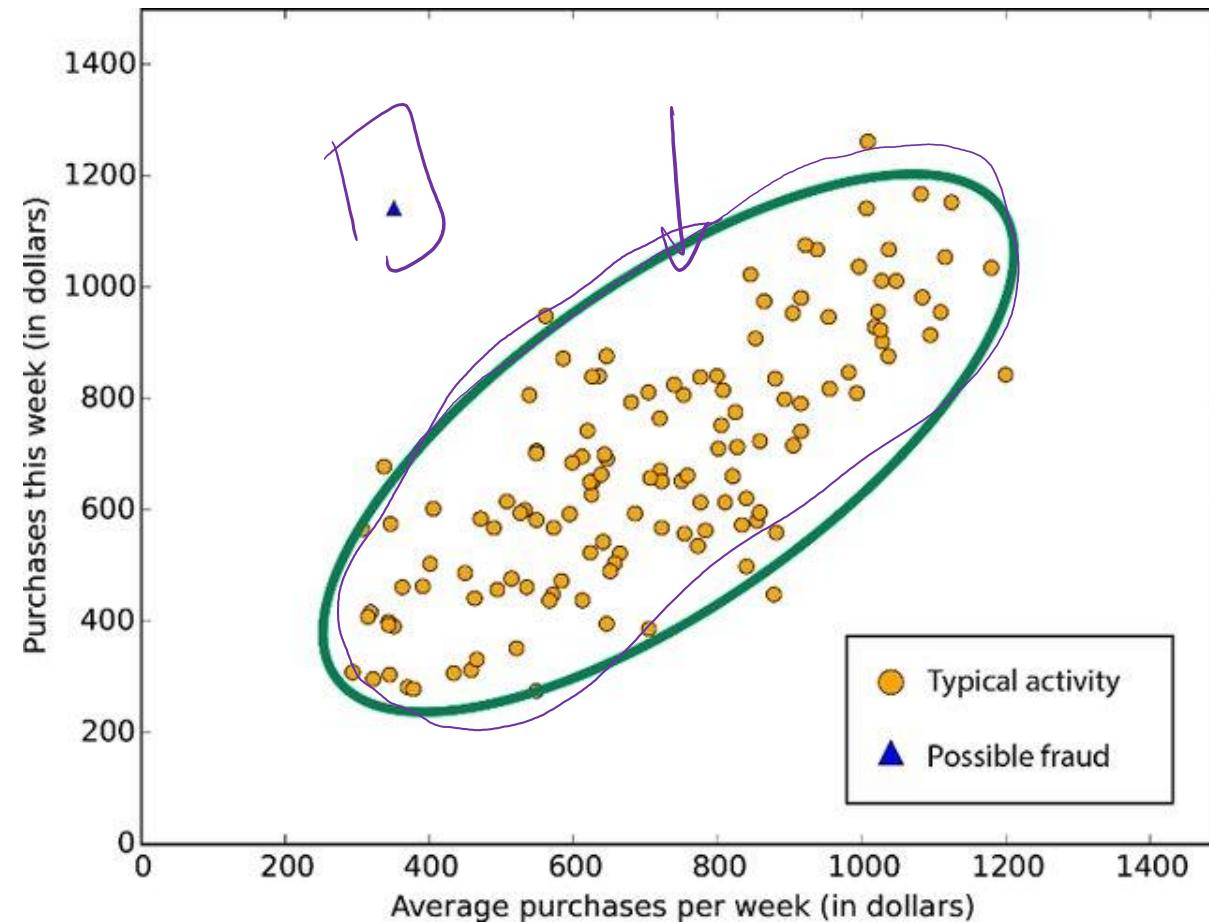
➤ Can be extended to multivariate data (e.g., bivariate GMM)

# Anomaly Detection (Semi-Supervised)

# What is the goal?



# Outlier



# Google Colab (Code Example)

# Part 3

# Performance Metrics

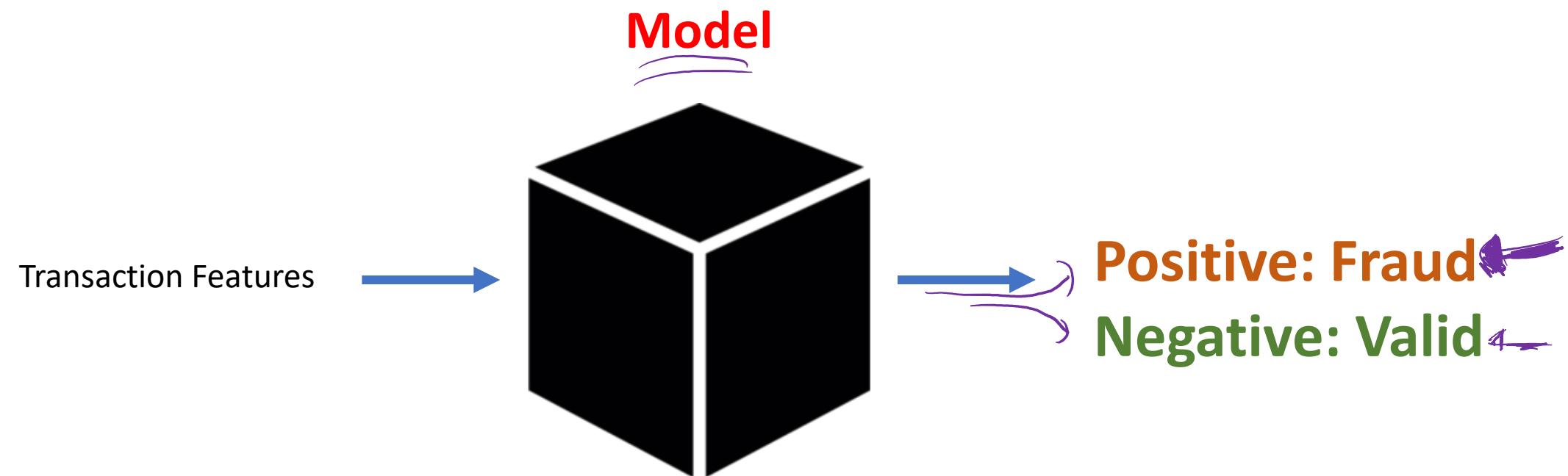
# Why Performance?

- Q: Why do we care about performance?
- Identifying how well our models are doing is not trivial. Easy to fall into the trap of believing (or wanting to believe) our models are working based on weak assessment.

# How to measure the performance of a model?

- Assume a case where we want to detect outliers and we know:
  - Dataset has 100 points
  - 98 are non-outlier
  - 2 are outliers
- If we detect all the points as non-outliers:
  - $98 \text{ True predictions} / 100 = \text{98\% accuracy for a model that is not working}$
- Q: How can we improve our performance measurements?

# Fraud Detection System



# Fraud Detection System

**(Positive = Fraud)**

- If transaction is **Valid**:
  - Prediction : **Valid** (True Negative) 😊
  - Prediction : **Fraud** (False Positive) 😞
- If transaction is **Fraud**:
  - Prediction : **Fraud** (True Positive) 😊
  - Prediction : **Valid** (False Negative) 😞

# Precision and Recall

➤ If transaction is **Valid**:

➤ Prediction : **Valid** (True Negative) **OK!**

➤ Prediction : **Fraud** (False Positive) **Not that bad!**

➤ If transaction is **Fraud**:

➤ Prediction : **Fraud** (True Positive) **GOOD!**

➤ Prediction : **Valid** (False Negative) **Super BAD!**

How many selected items are relevant?

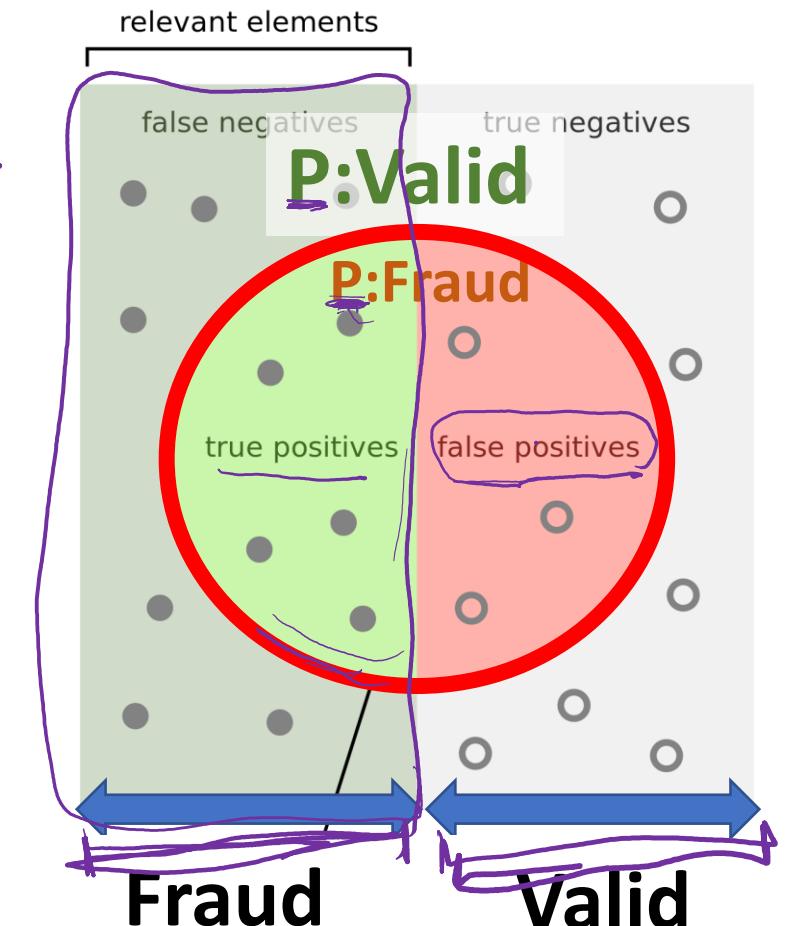
(MISTAKE)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

(MISS)



# Confusion Matrix

		Actual Value (as confirmed by experiments)	
		positives	negatives
Predicted Value (predicted by the test)	positives	TP True Positive	FP False Positive
	negatives	FN False Negative	TN True Negative

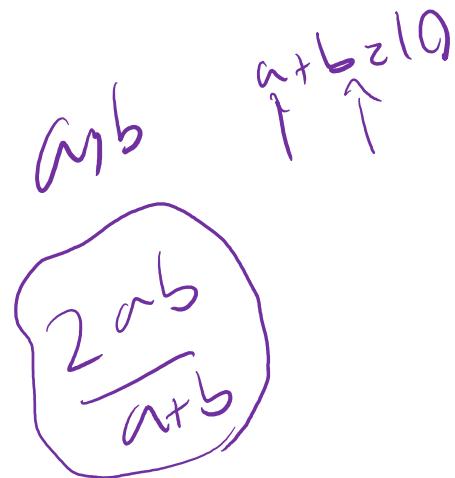
➤ Table used to describe prediction performance on a set of test data

$$\frac{TP}{TP+FP} = \text{Pre}$$

$$ACC = \frac{TP+TN}{All}$$

$$Rec = \frac{TP}{TP+FN}$$

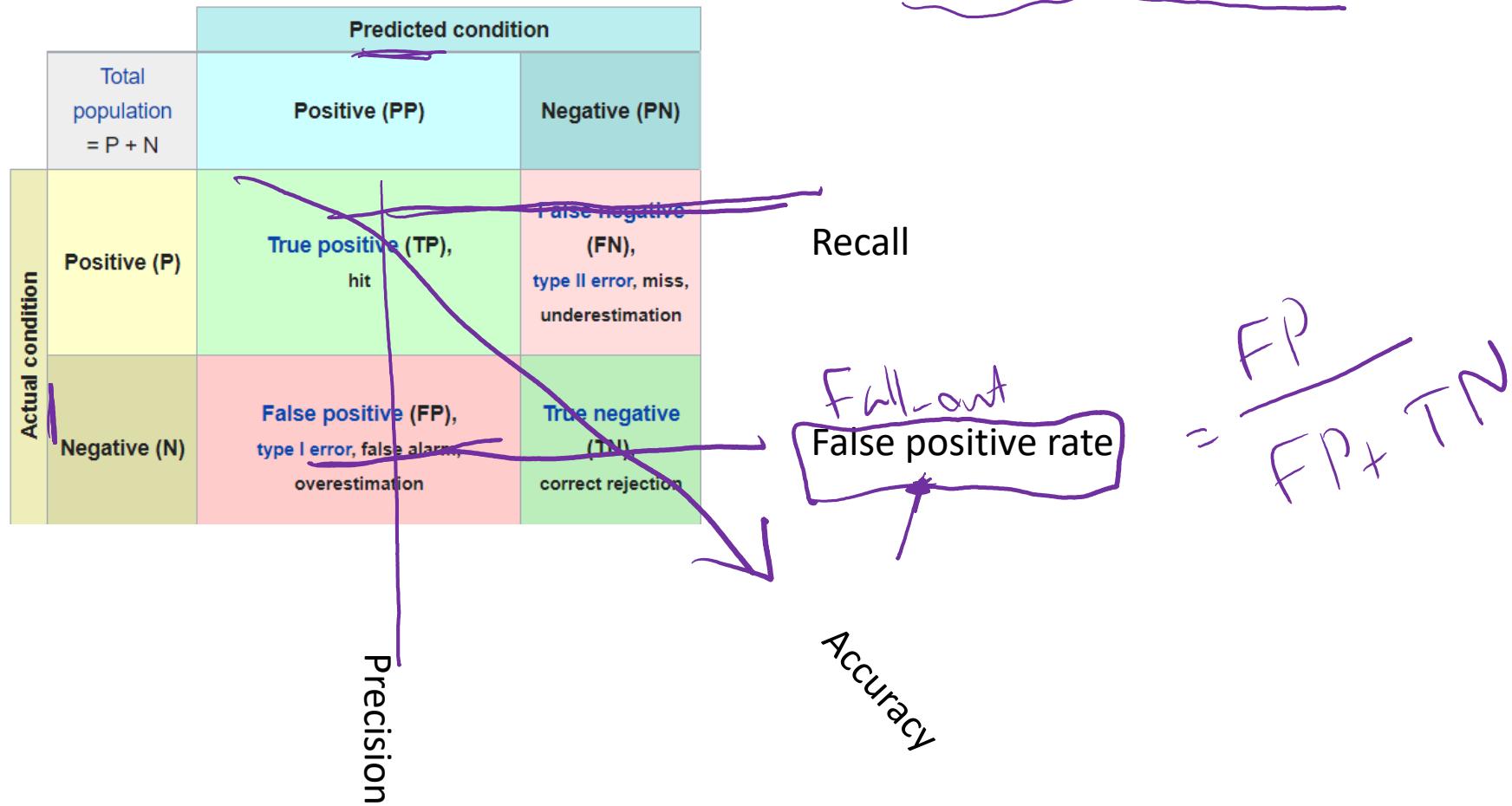
# F<sub>1</sub> score



$$F_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

- A balanced measure of accuracy giving equal importance to recall and precision
- The highest possible value of F<sub>1</sub> is 1, and lowest possible value is 0

# Confusion Matrix (by heart)



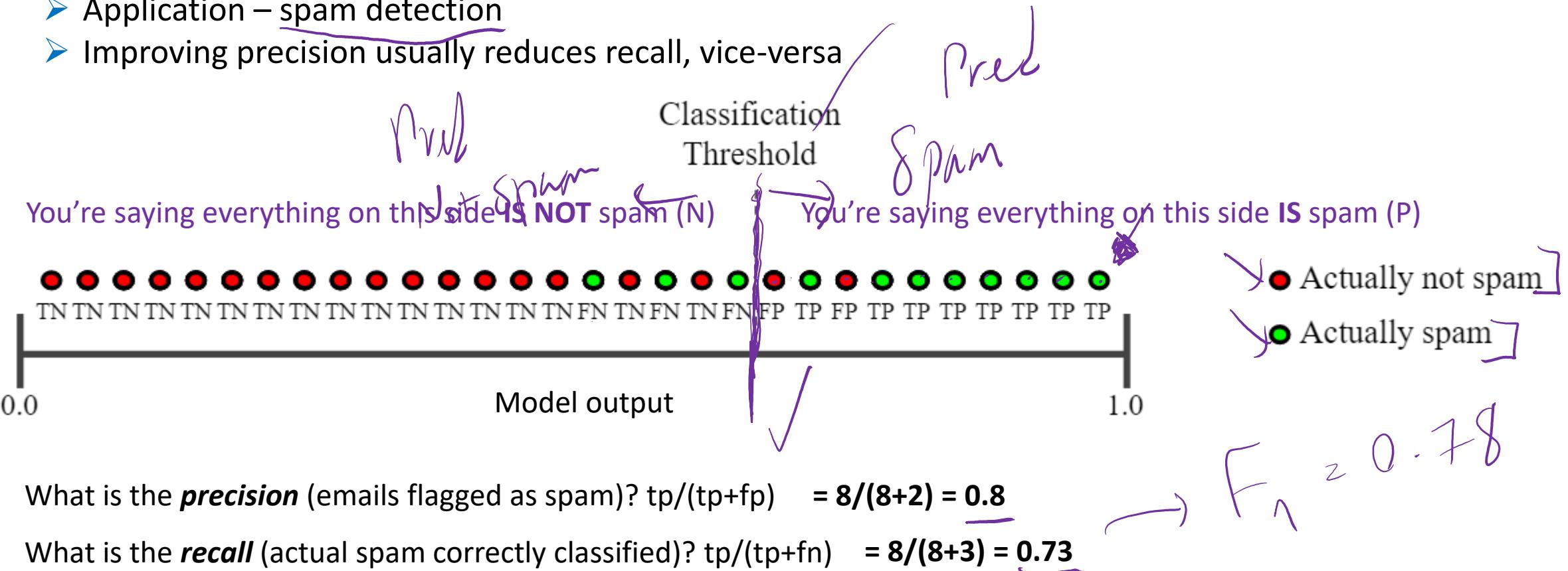
$$\text{False positive rate} = \frac{FP}{FP + TN}$$

# Confusion Matrix

		Predicted condition		Sources: [13][14][15][16][17][18][19][20] view · talk · edit		
		Total population $= P + N$	Positive (PP)	Negative (PN)	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold $(PT) = \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	Positive (P)	True positive (TP), hit	False negative (FN), type II error, miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate $= \frac{FN}{P} = 1 - TPR$	
	Negative (N)	False positive (FP), type I error, false alarm, overestimation	True negative (TN), correct rejection	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$	
Prevalence $= \frac{P}{P+N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR-) $= \frac{FNR}{TNR}$		
Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN} = 1 - FOR$	Markedness (MK), deltaP $(\Delta p) = PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$		
Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	$F_1$ score $= \frac{2PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times DFR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$		

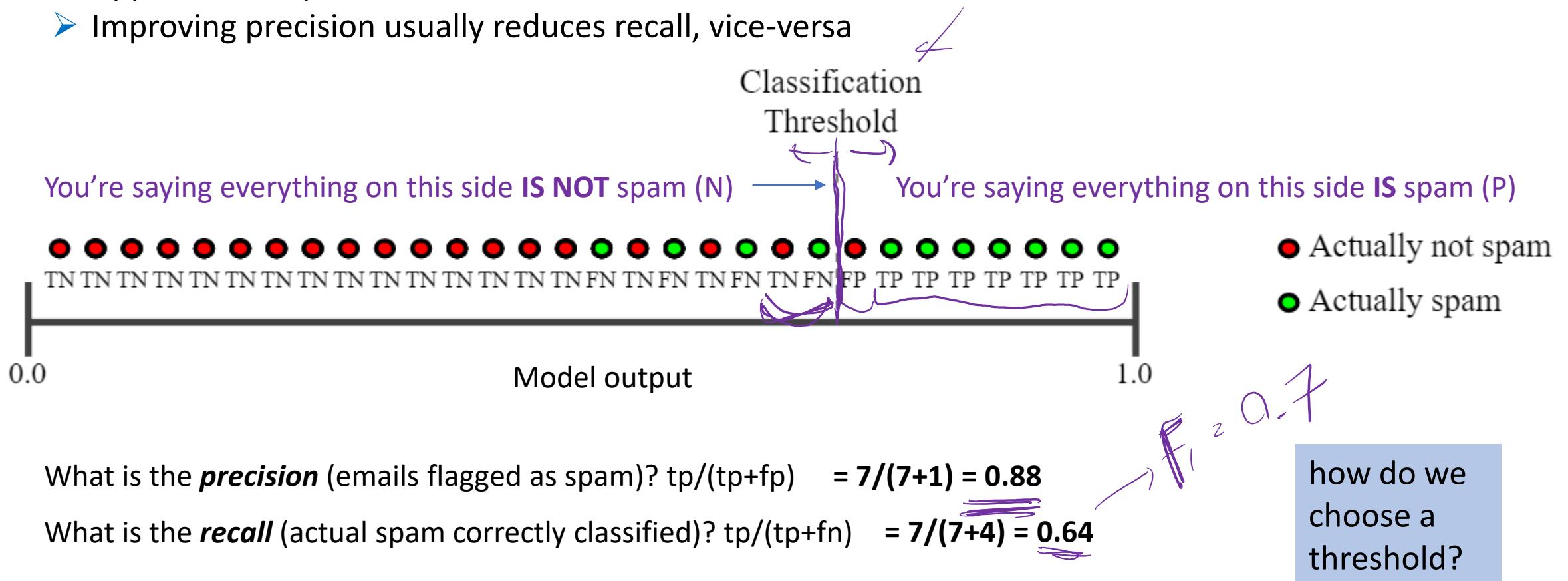
# Precision and Recall – tug of war

- Application – spam detection
- Improving precision usually reduces recall, vice-versa

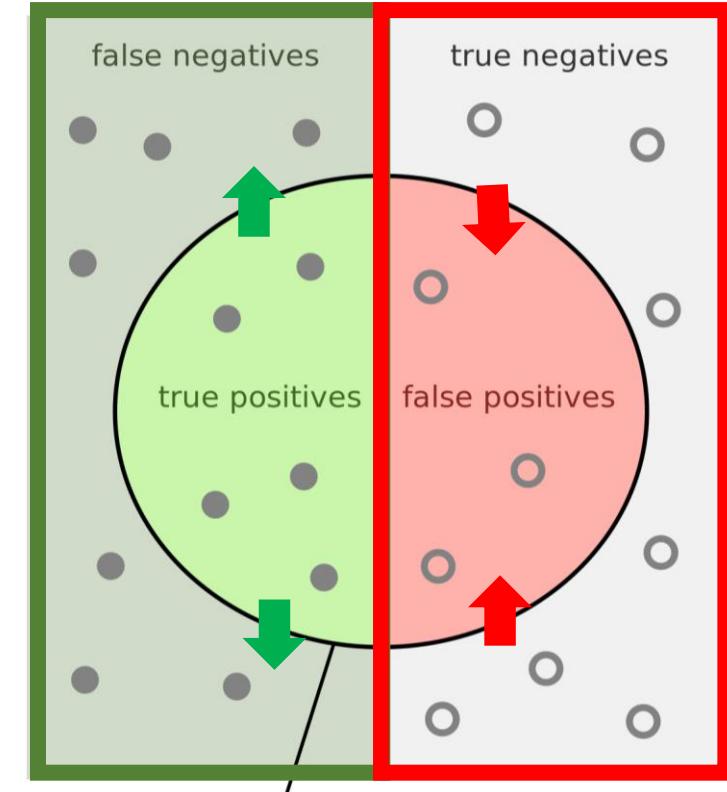
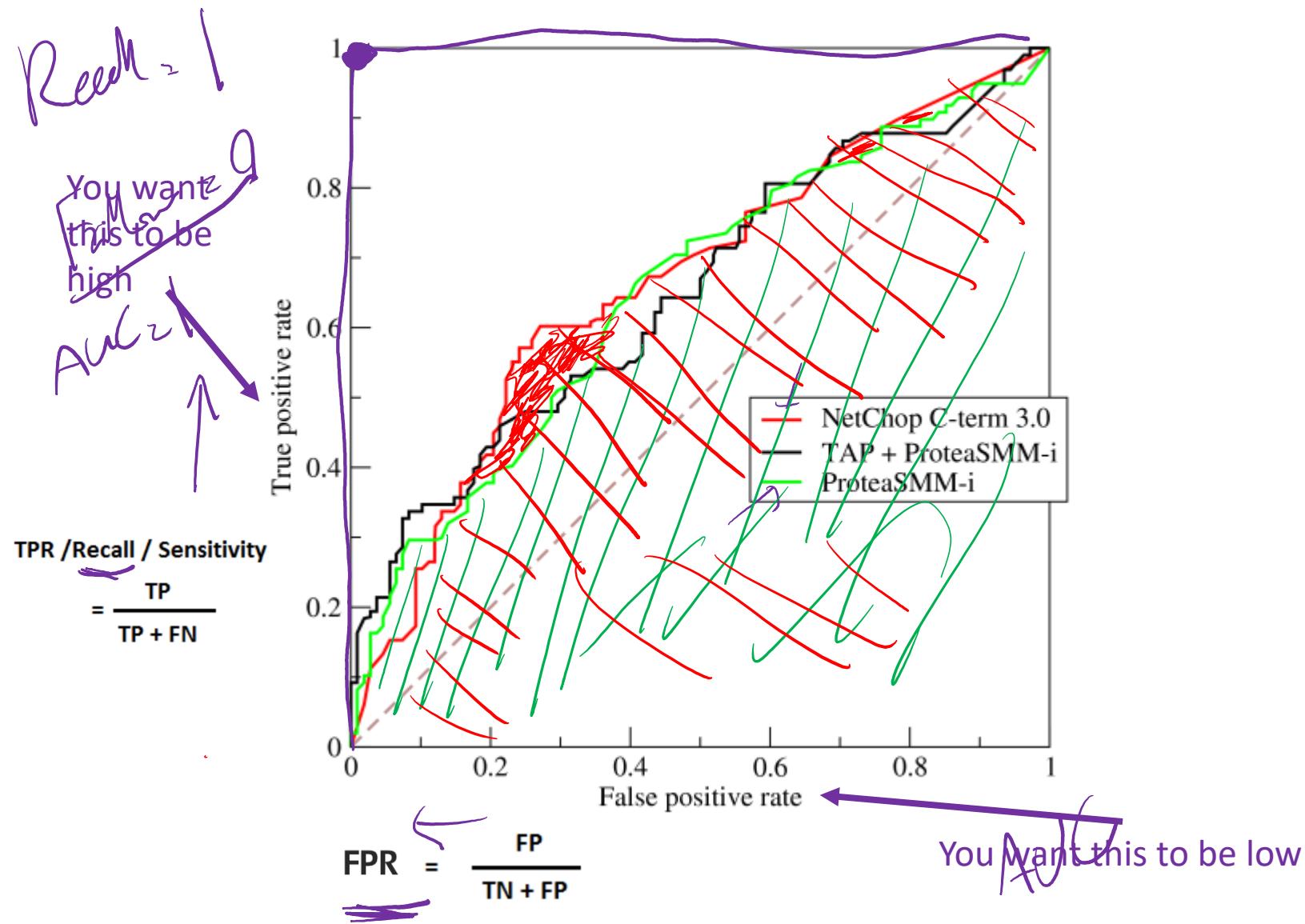


# Precision and Recall – tug of war

- Application – spam detection
- Improving precision usually reduces recall, vice-versa

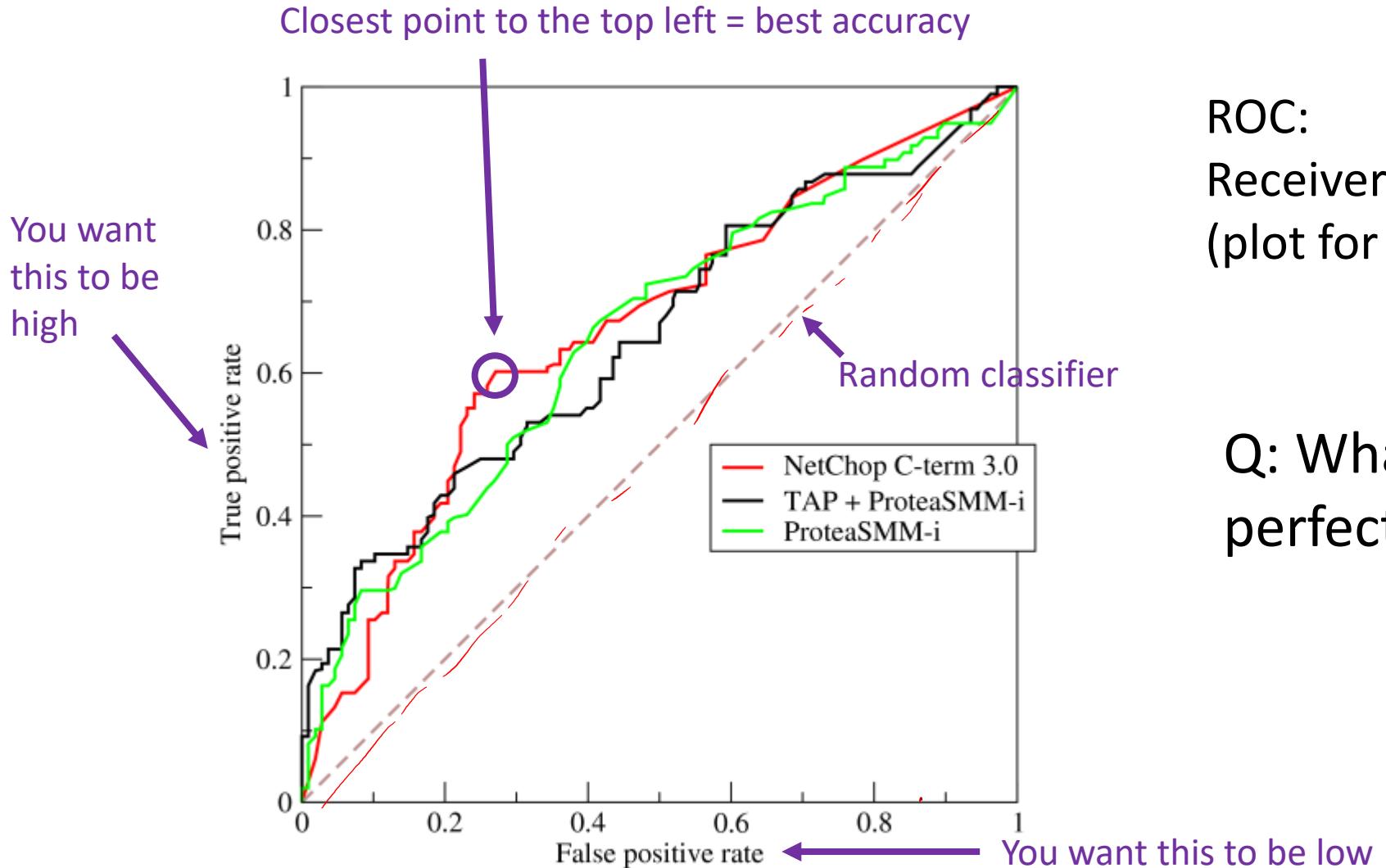


# ROC (Receiver Operating Characteristic) Curve



Source: [Wikipedia](#)

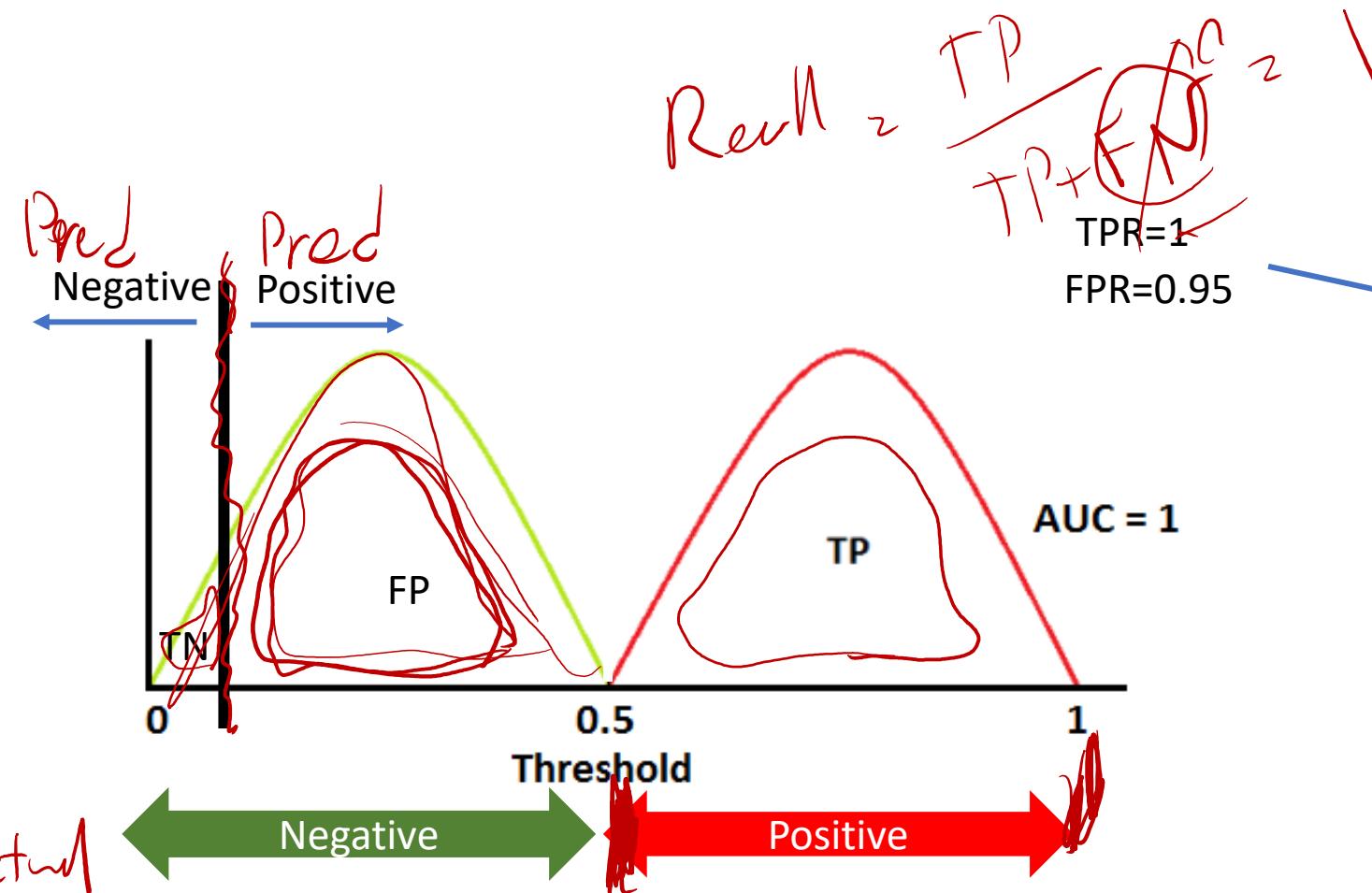
# ROC Curve



ROC:  
Receiver Operating Characteristic  
(plot for binary classifiers)

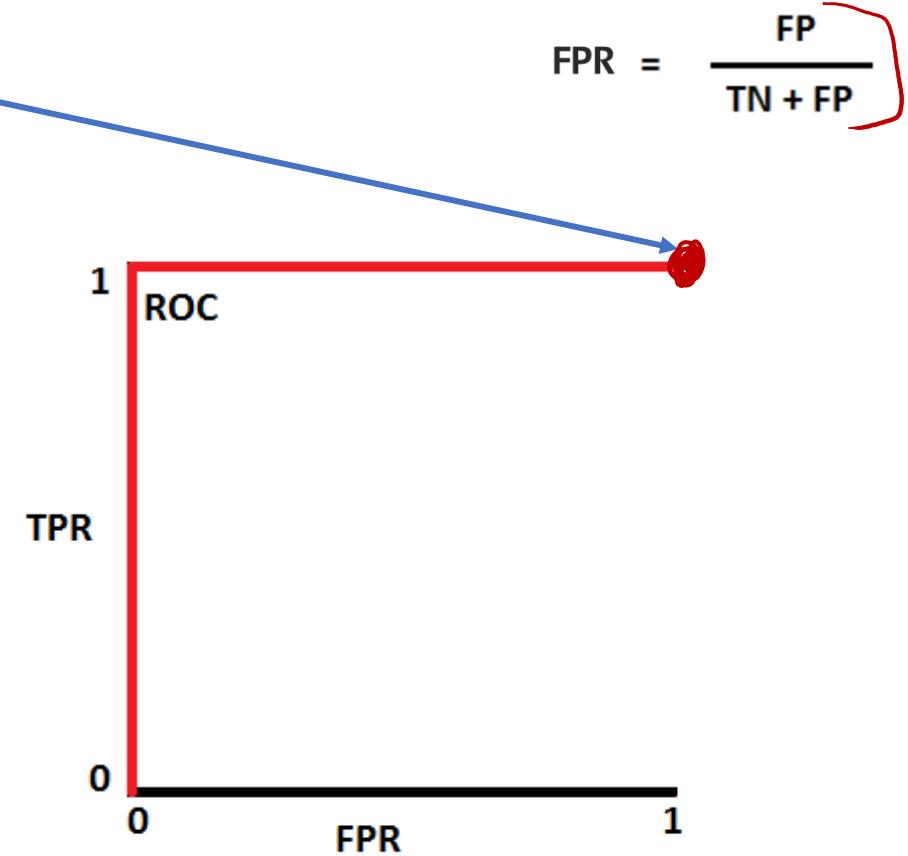
Q: What would be a perfect classifier?

# ROC Curve

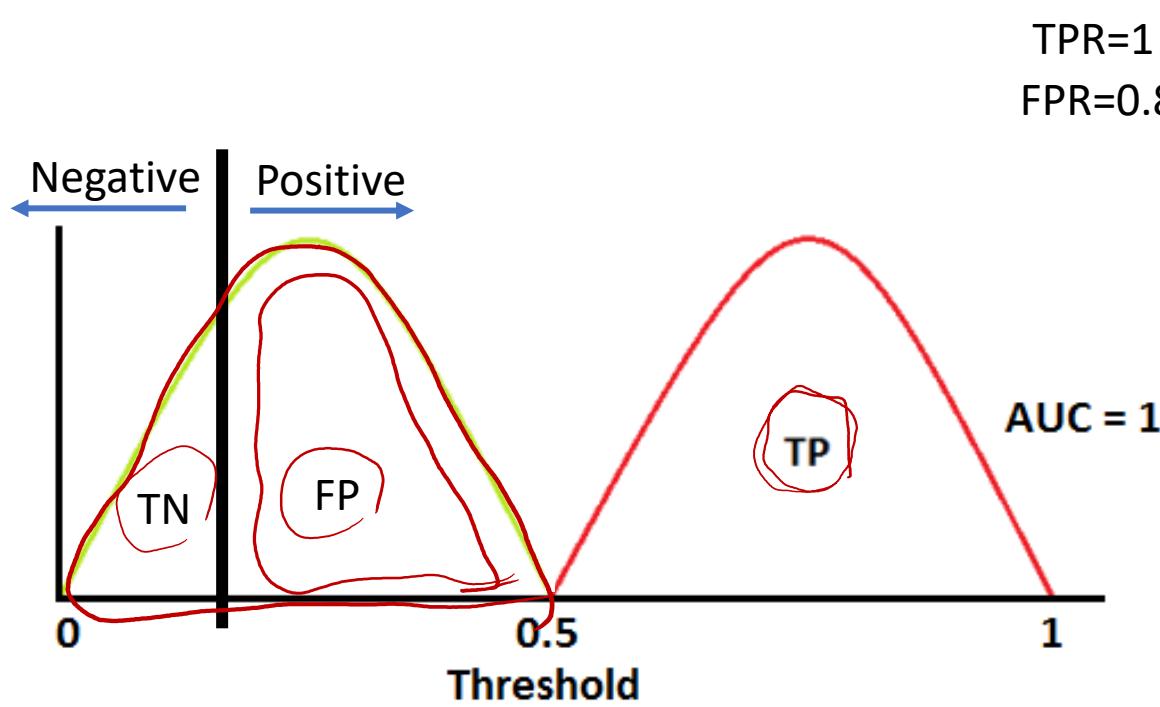


$$\text{TPR / Recall / Sensitivity} = \frac{TP}{TP + FN}$$

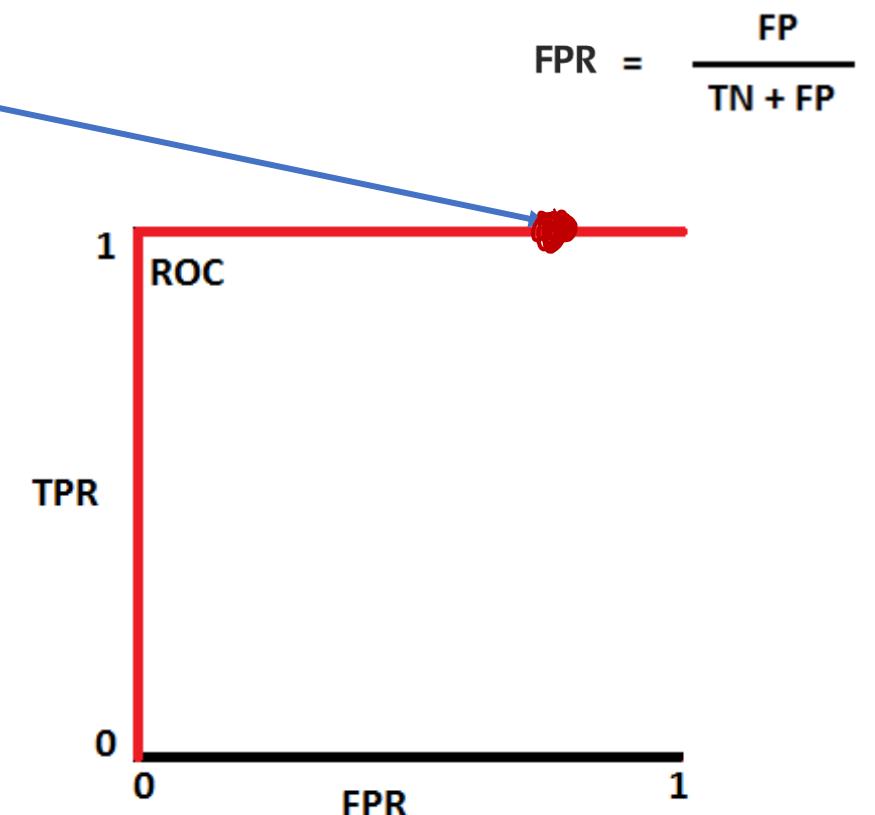
$$\text{FPR} = \frac{FP}{TN + FP}$$



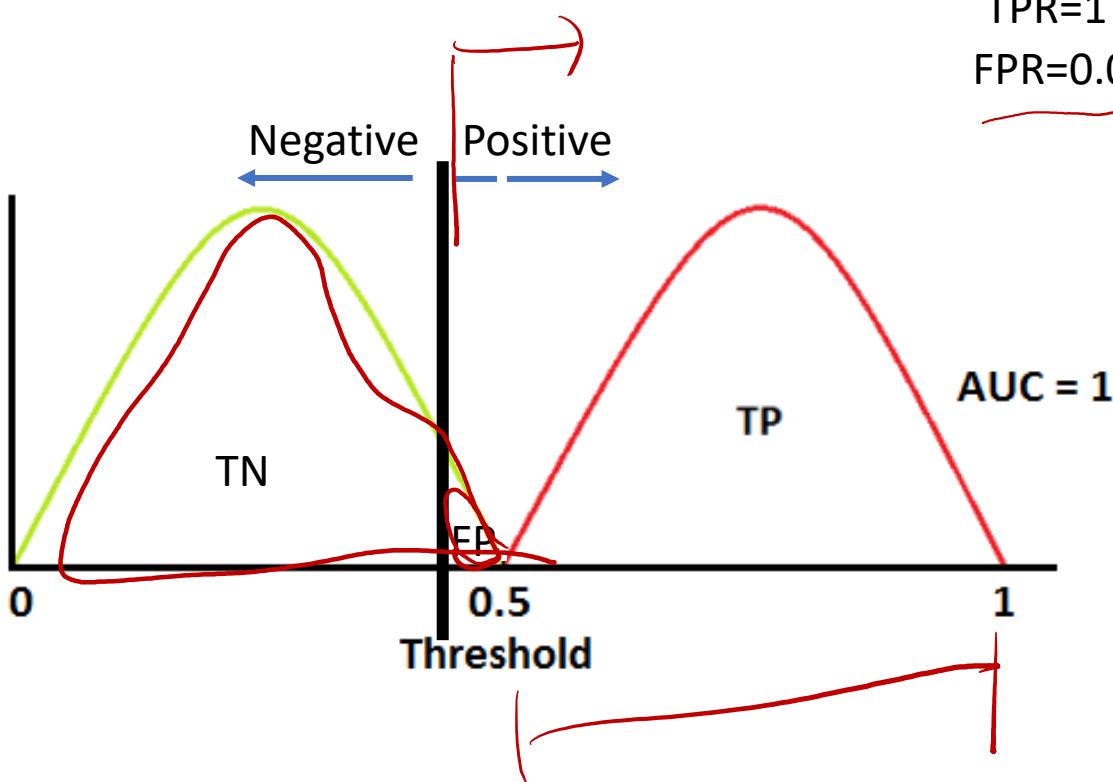
# ROC Curve



$$\text{TPR / Recall / Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$



# ROC Curve



TPR=1  
FPR=0.05

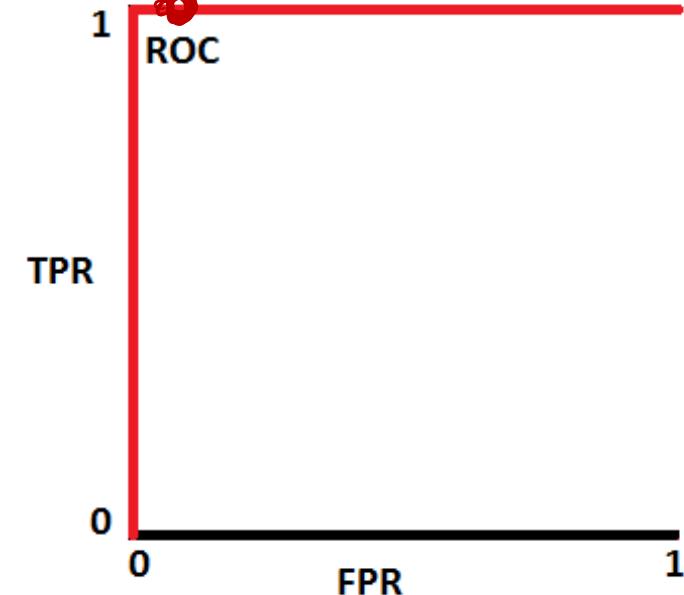
AUC = 1

0 0.5 1

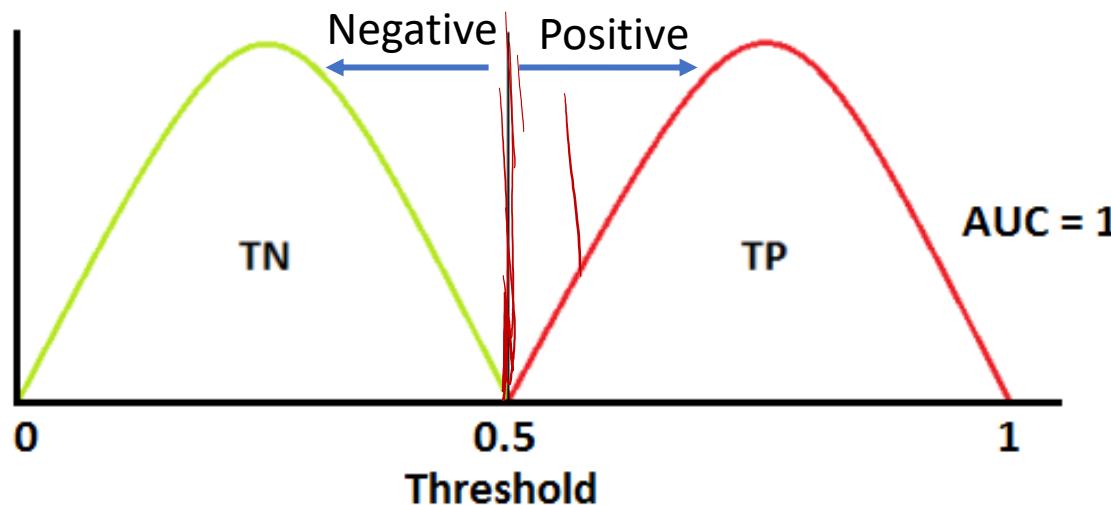
Threshold

$$\text{TPR / Recall / Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

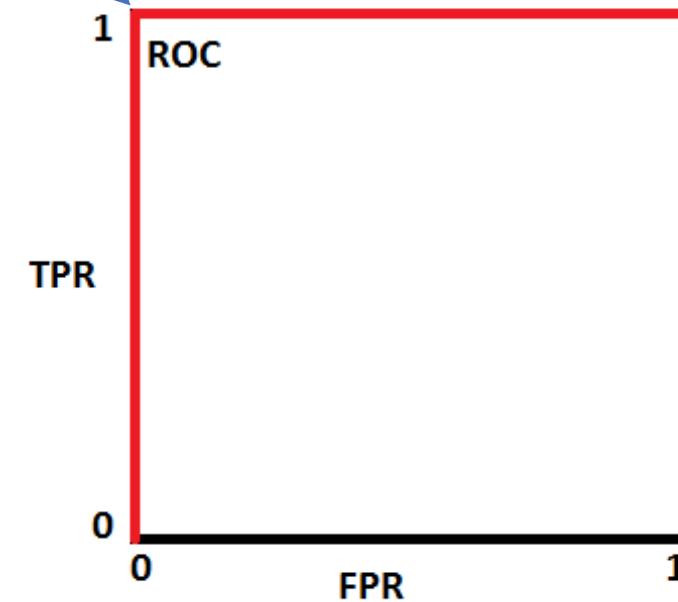
$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$



# ROC Curve



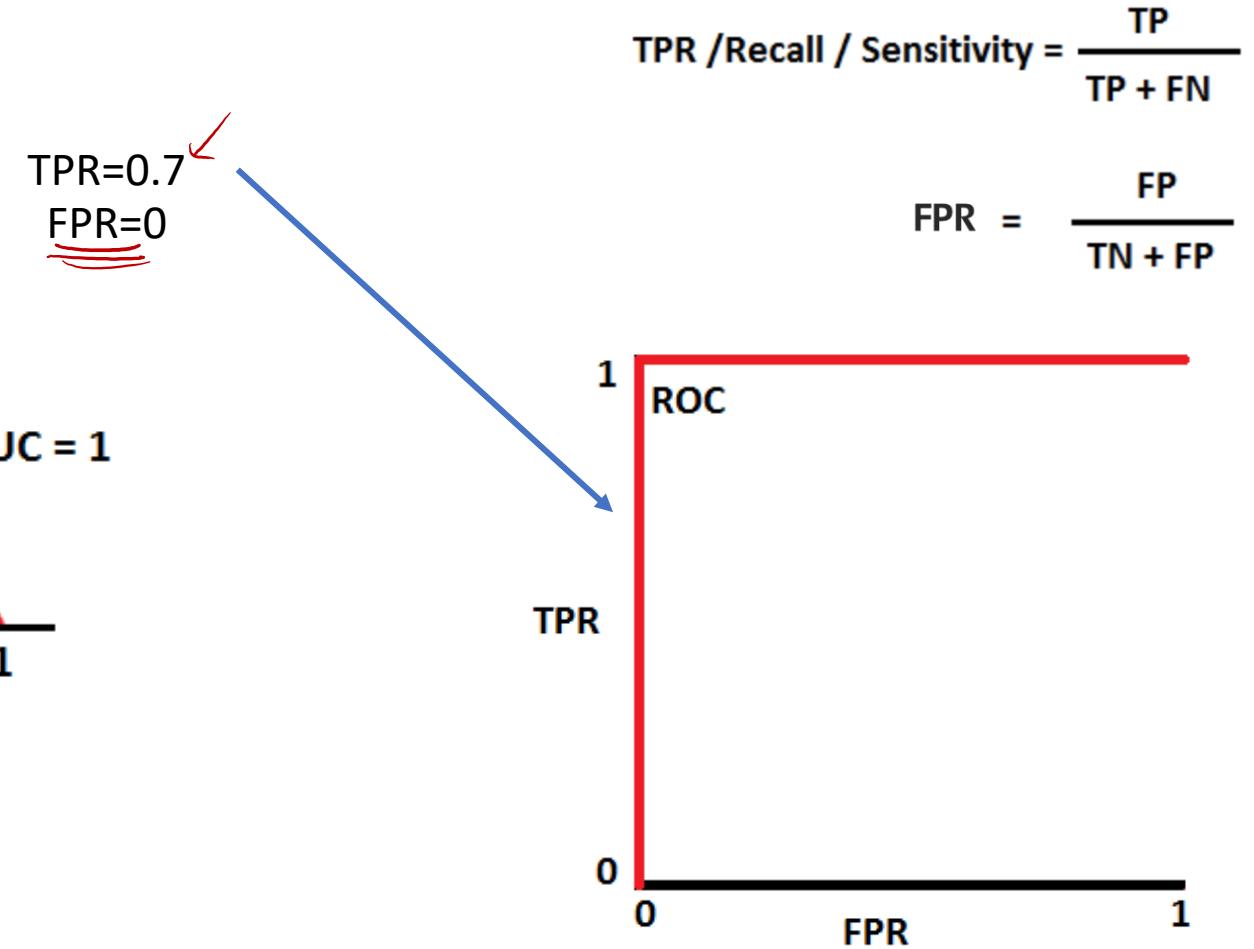
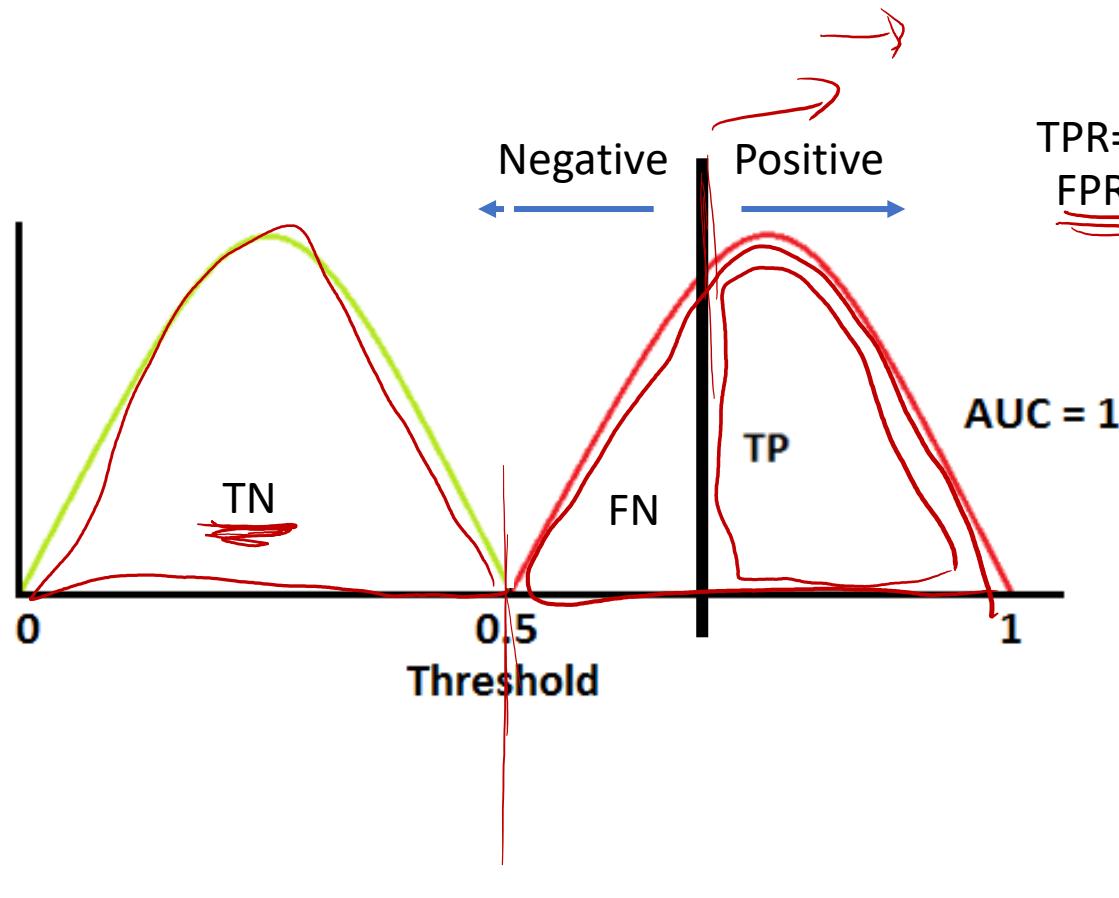
TPR=1  
FPR=0



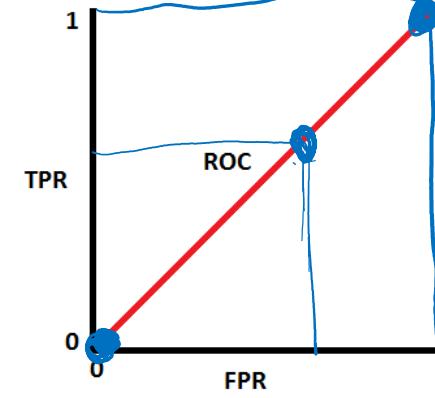
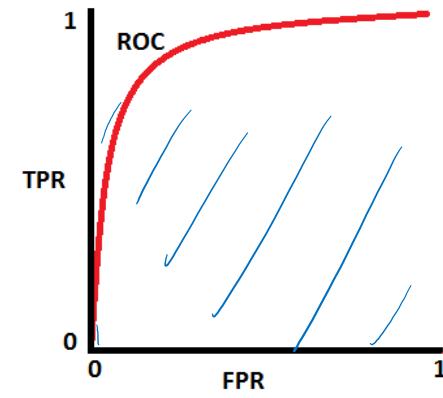
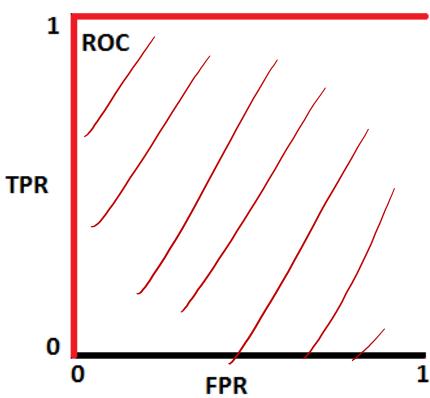
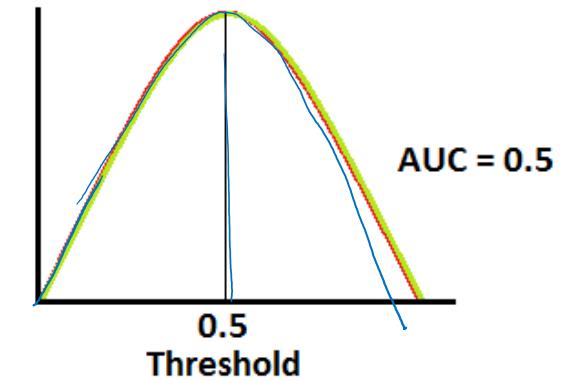
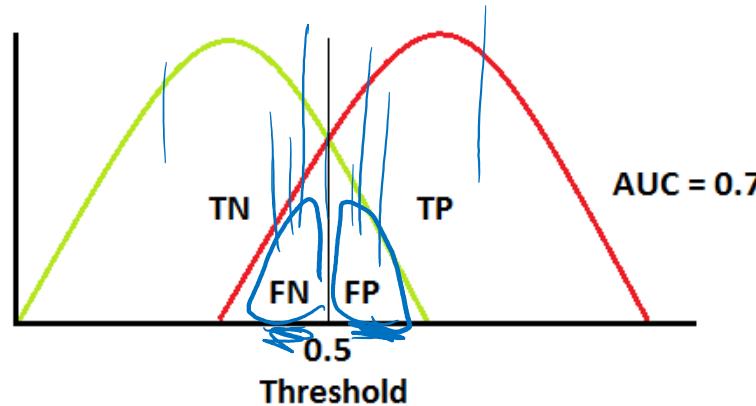
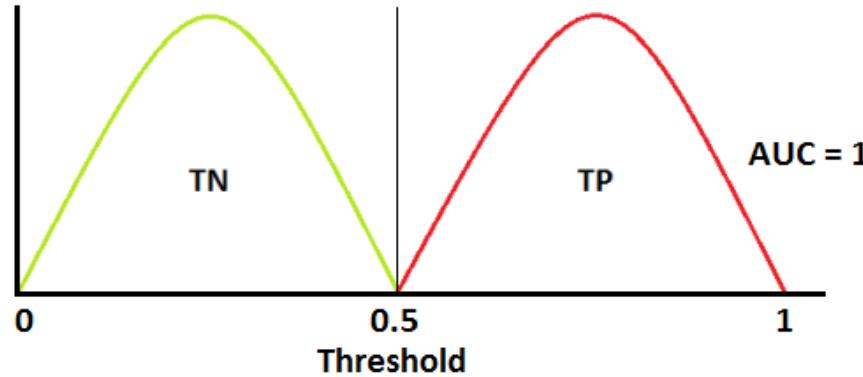
$$\text{TPR / Recall / Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

# ROC Curve



# AUC (Area Under the Curve)



Source: [Sarang Narkhede](#)

*AUC > 1*

*AUC < 1*

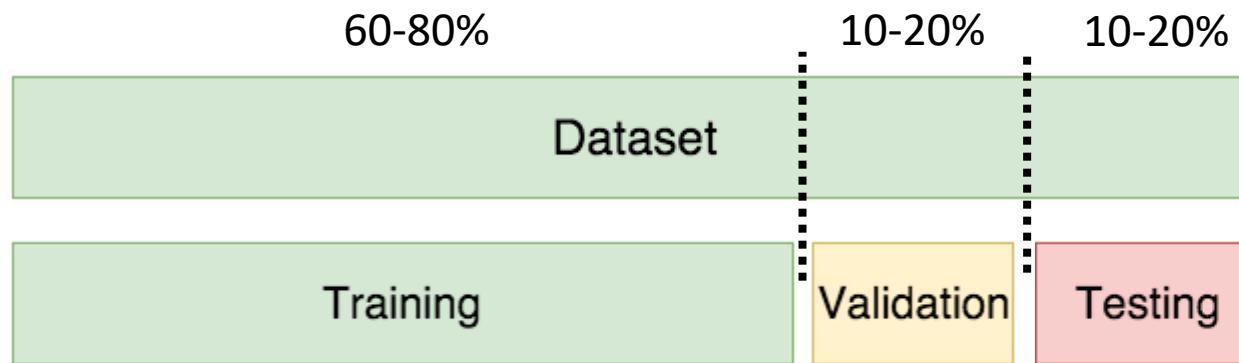
*Recall = Fallout*

# Data Selection

- How we select our data is another aspect that is often overlooked.
- Can have serious effects on the prediction's performance.
- Q: What are some issues with arbitrary data selection?

# Splitting the Dataset

## ➤ Training, Validation and Testing



➤ More training data => better model

➤ More testing data => more confidence in assessment

➤ Cross-Validation is an attempt to have both...

# Independent and Identically Distributed

- It is important that the generative process of data is the same for all data and the process has no memory of past generated samples.
- Sampling of data needs to be independent

# Limited Data

- There are often situations where it is difficult to obtain sufficient data to train a machine learning algorithm.
- For example: A common practice with medical data is to **increase the sample size by obtaining multiple samples from the same patient.**
- Q: What are some concerns resulting from this?

# Generalization

- We have seen previously that correctly predicting on new data is what we're interested in.
- **Golden Rule:** No model selection decisions should come from the test data!

# Next Time

- Week 4 Q/A Support Session on Thursday and Friday
  - Project Questions
- Project 1 is due on February 4<sup>th</sup>
- Reading assignment 4 (due on Monday)
- Week 5 Lecture - Data Processing
  - Linear Algebra
  - Analytical Geometry
  - Data Augmentation