

**GLADSTONE**  
INSTITUTES



Bakar Computational Health  
Sciences Institute

# Introductory Machine Learning for Biologists

• Min-Gyoung Shin

March 2, 2025

# Table of contents

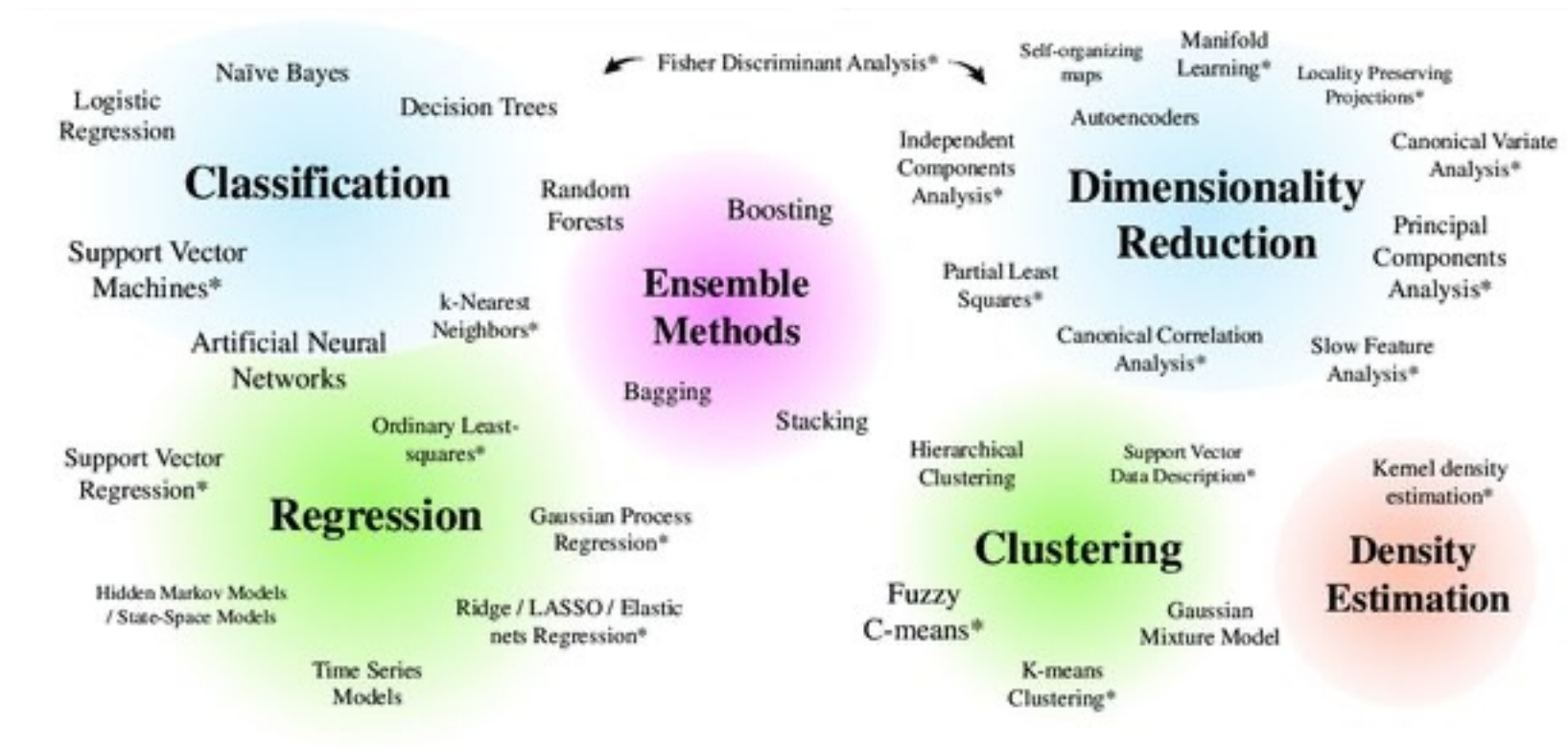
## Introduction

- Supervised and Unsupervised
- Bias and Variance
- Cross Validation

## Methods to learn

- K-Nearest Neighbors (+ Hands on practice )
- K-means clustering (+ Hands on practice )
- Decision Tree (+ Hands on practice )

# Examples of Machine Learning Methods





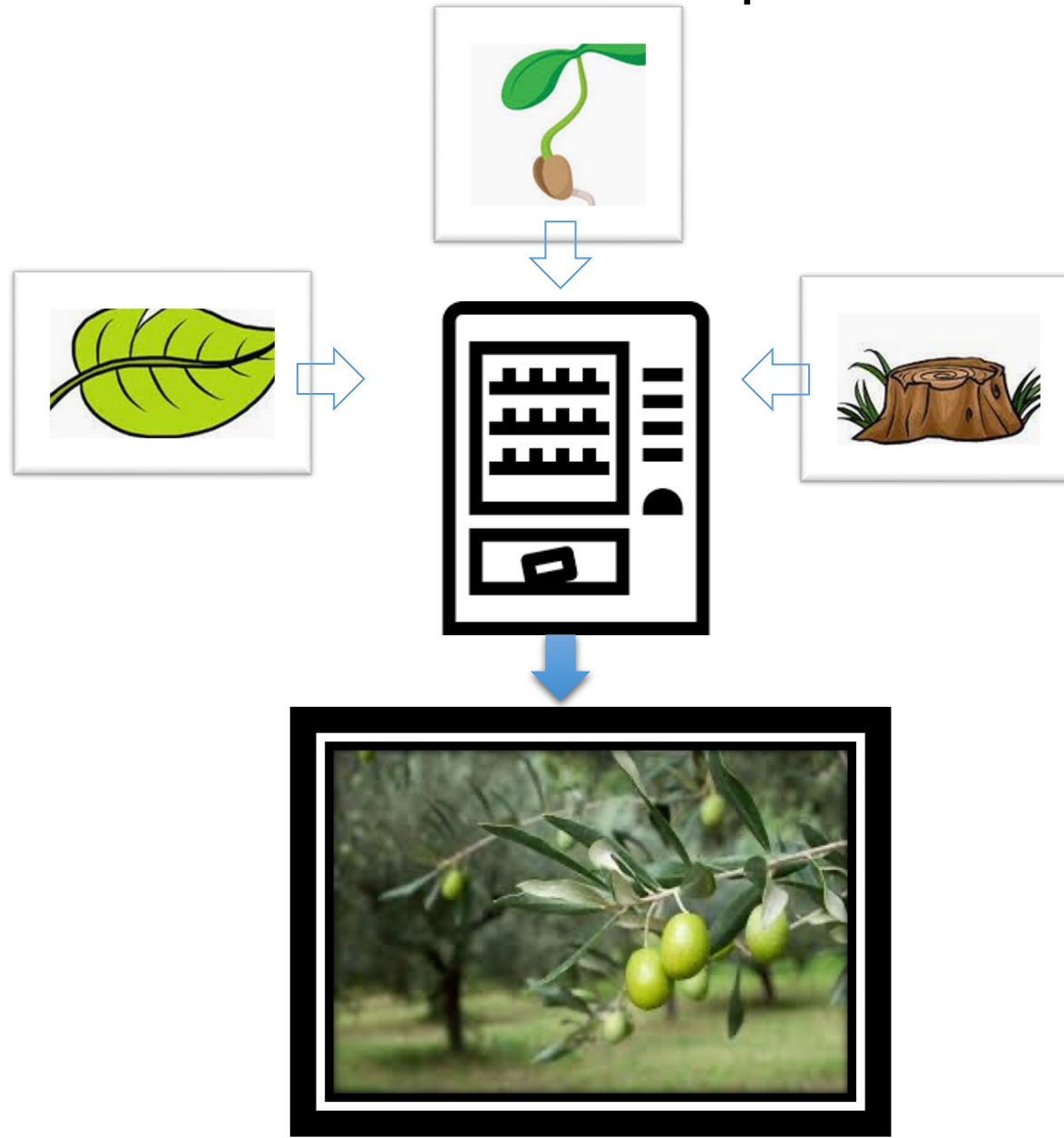
# What we are going to learn

K-Nearest Neighbor

K-Means

Decision Tree

# It's all about prediction!



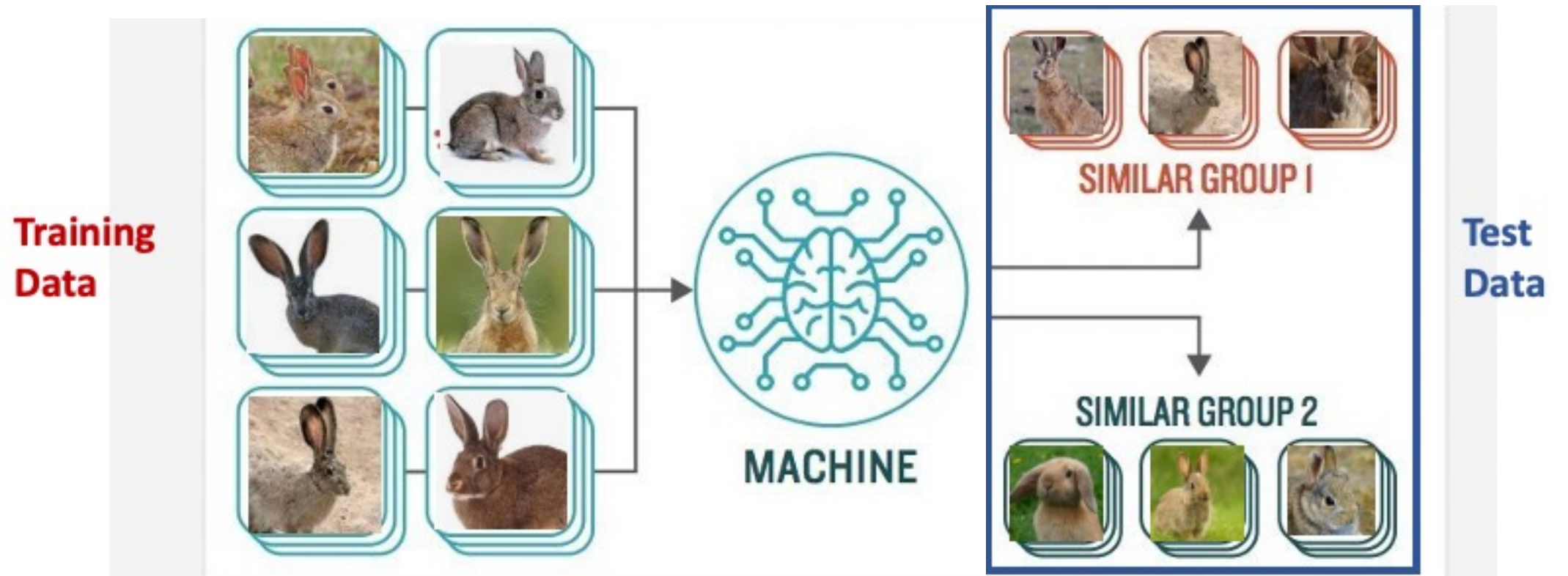
# Example data

Label

Attributes / Features

	Disease	Heart rate	Respiratory rate	temperature	color	age	Behavior X	weight	...
Mouse 1	T	310	90	37	grey	1	Y	12	
Mouse 2	F	400	200	36.5	grey	1	N	13	
Mouse 3	T	430	100	36.5	black	1	N	11	
Mouse 4	F	300	190	37.2	grey	1	N	10	
Mouse 5	T	550	221	38	black	1	Y	9	
Mouse 6	F	700	130	37.7	grey	2	N	11.5	
...									

# Training Data and Test Data



# How **SUPERVISED** Machine Learning Works

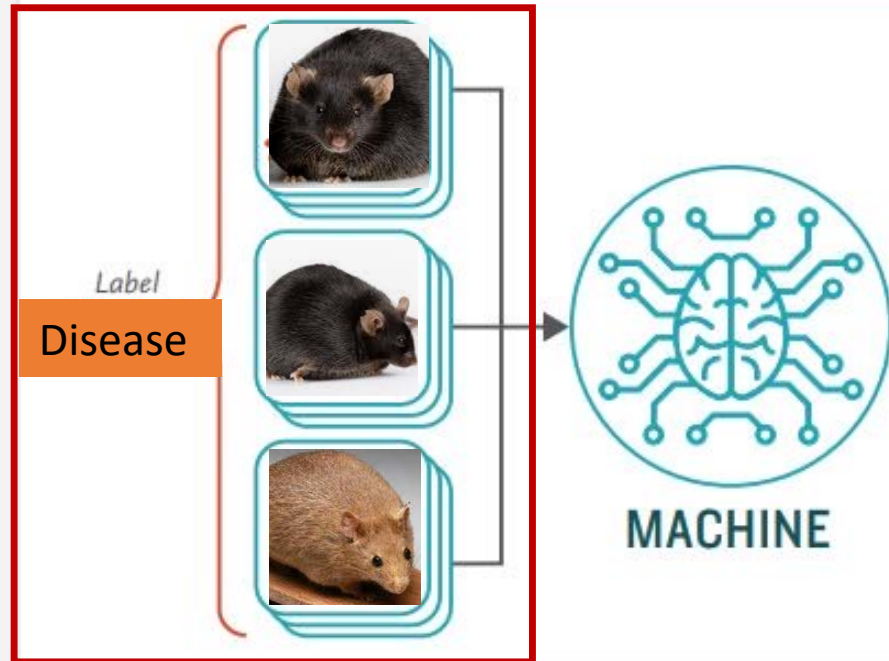
## STEP 1

Provide the machine learning algorithm categorized or "labeled" input and output data from to learn

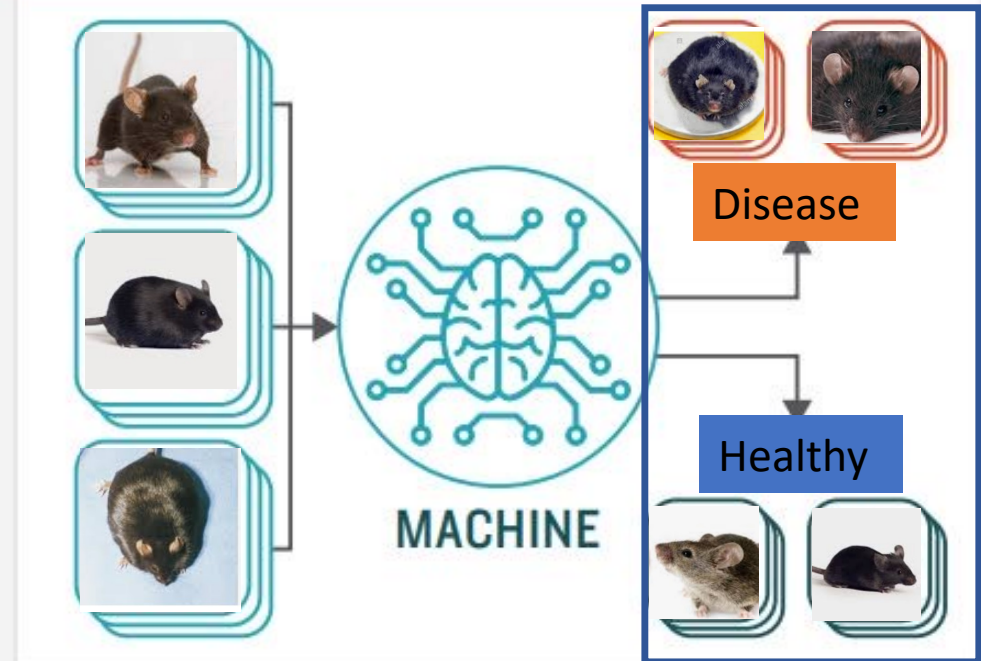
## STEP 2

Feed the machine new, unlabeled information to see if it tags new data appropriately. If not, continue refining the algorithm

Training Data



Test Data



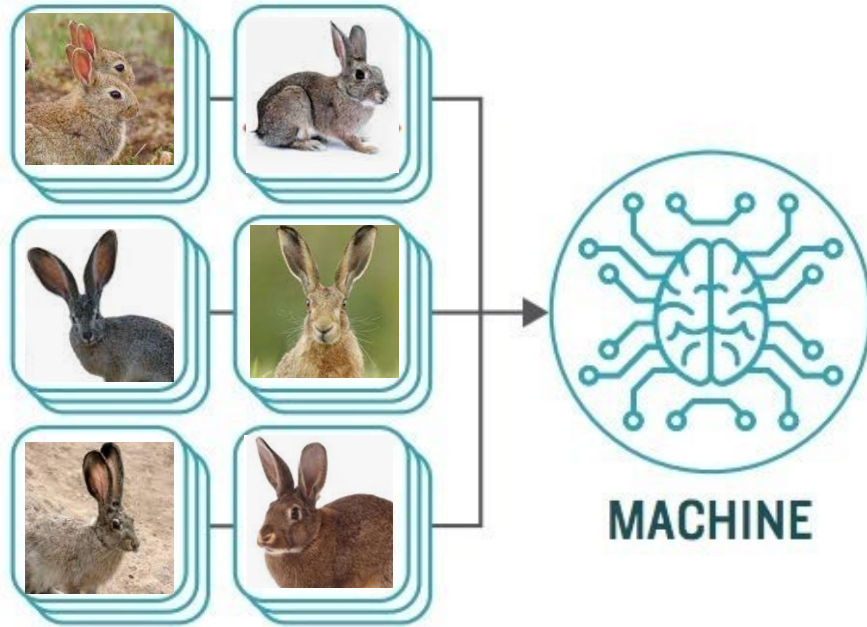
The model requires user input of known values for training



# How **UNSUPERVISED** Machine Learning Works

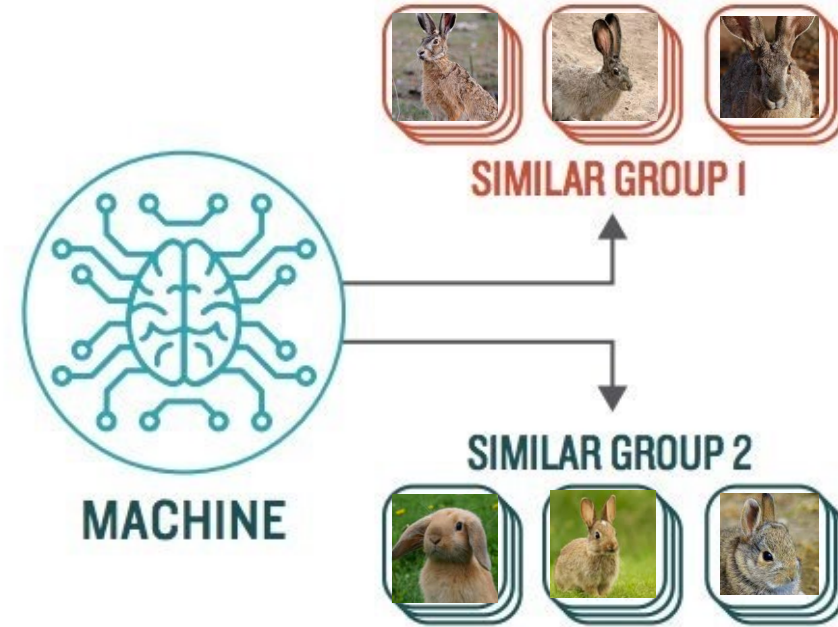
## STEP 1

Provide the machine learning algorithm uncategorized, unlabeled input data to see what patterns it finds



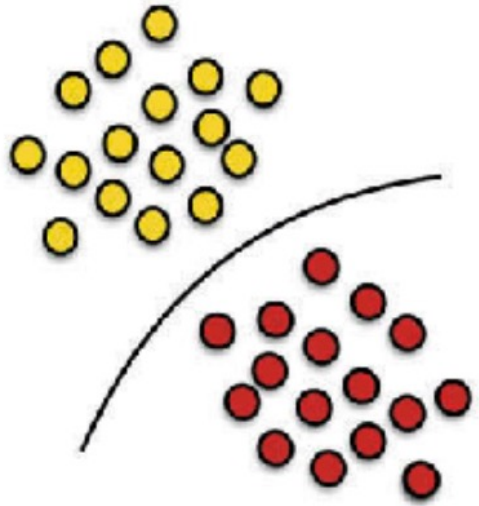
## STEP 2

Observe and learn from the patterns the machine identifies



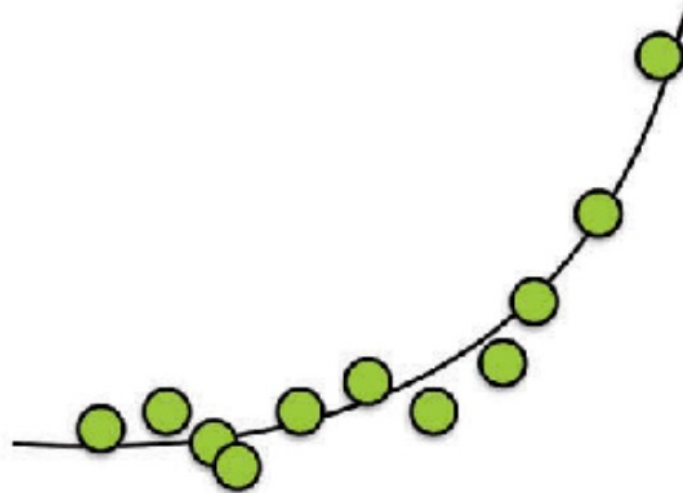
- The models are given unlabeled in order to identify relevant pattern
  - The machine finds the hidden structure

**a**



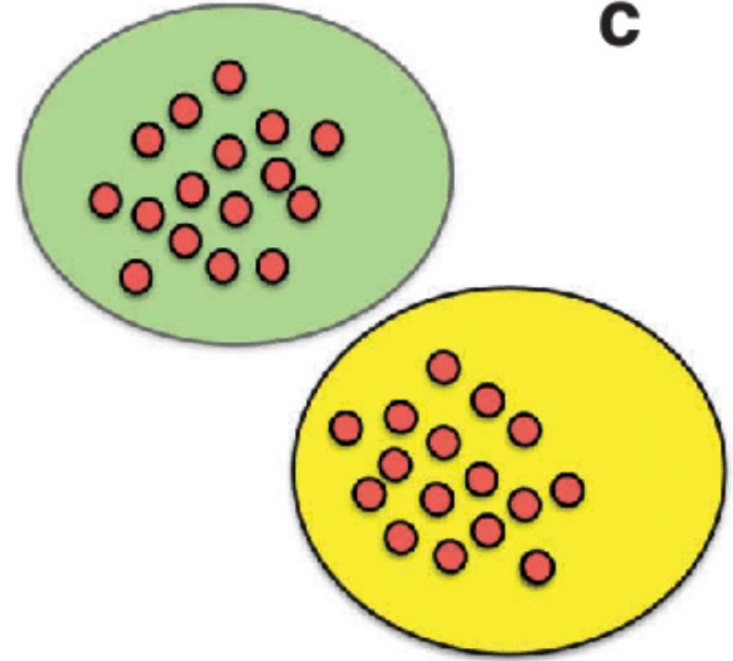
Classification

**b**



Regression

**c**



Clustering

Q1. I performed a knockout experiment. I want to cluster genes based on gene expression similarity.

- Supervised
- Unsupervised

Q1. I performed a knockout experiment. I want to cluster genes based on gene expression similarity.

- Supervised
- Unsupervised



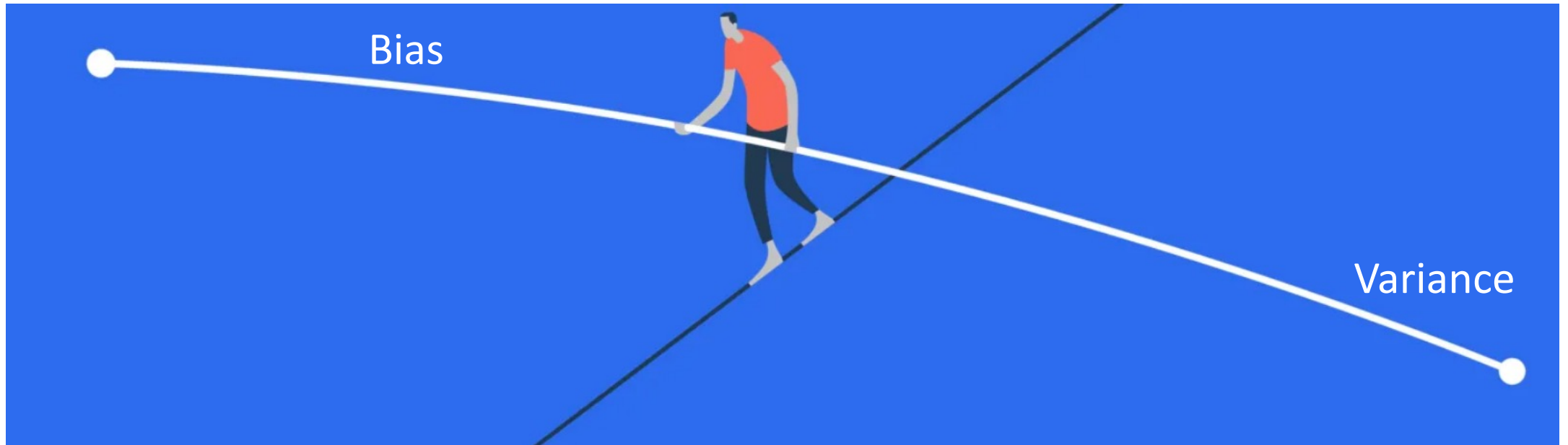
Q2. I have a cohort with variant information and disease states. I want to make a machine learning model to predict disease states.

- Supervised
- Unsupervised

Q2. I have a cohort with variant information and disease states. I want to make a machine learning model to predict disease states.

- Supervised
- Unsupervised

# Finding a good prediction model: Balance of Bias and Variance

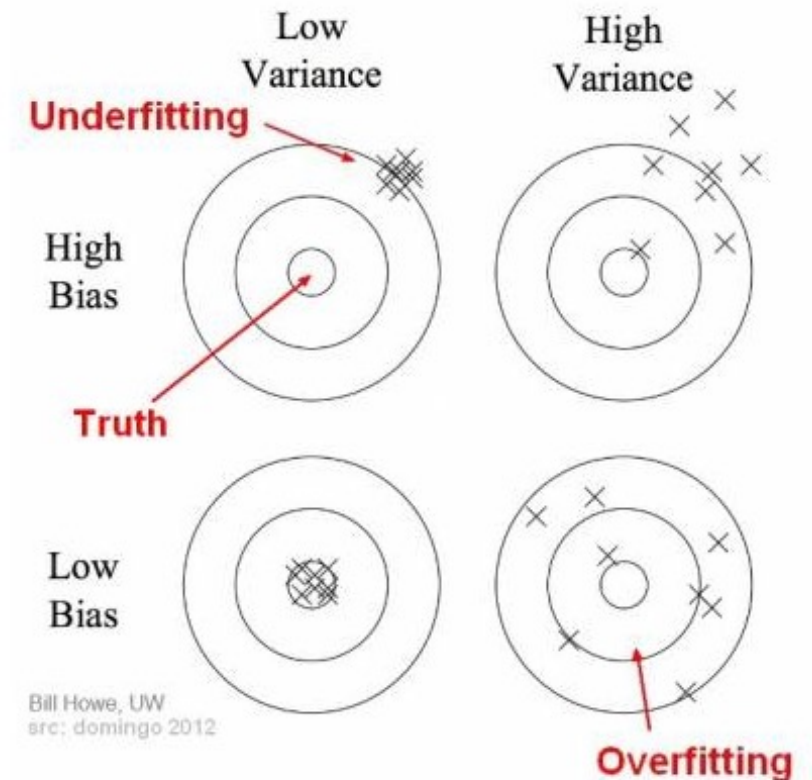


# Bias and Variance

$$Err(x) = \left(E[\hat{f}(x)] - f(x)\right)^2 + E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] + \sigma_e^2$$

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

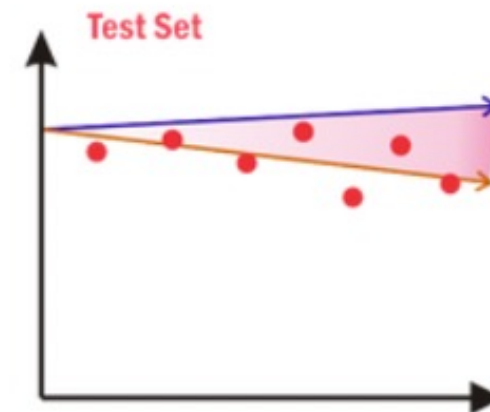
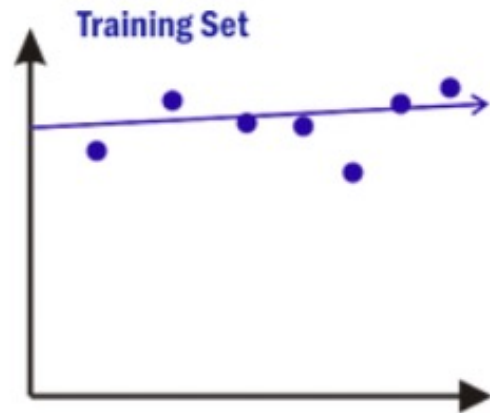
$f(x)$ : prediction function  
 $\hat{f}(x)$ : estimate of  $f(x)$



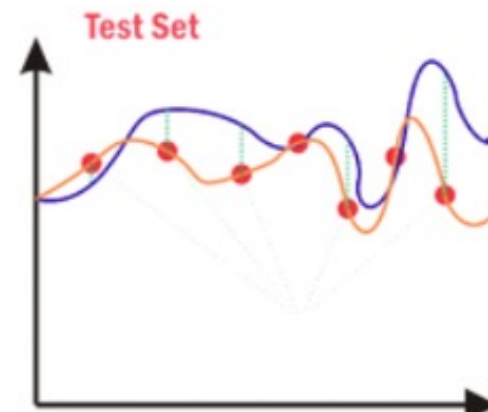


# Bias and Variance

High bias  
Low variance



Low bias  
High variance



Q3. Predict variance level and bias level



Q3. Predict variance level and bias level



Bias error	High
Variance error	Low

Q4. Predict variance level and bias level





Q4. Predict variance level and bias level



Bias error

Low

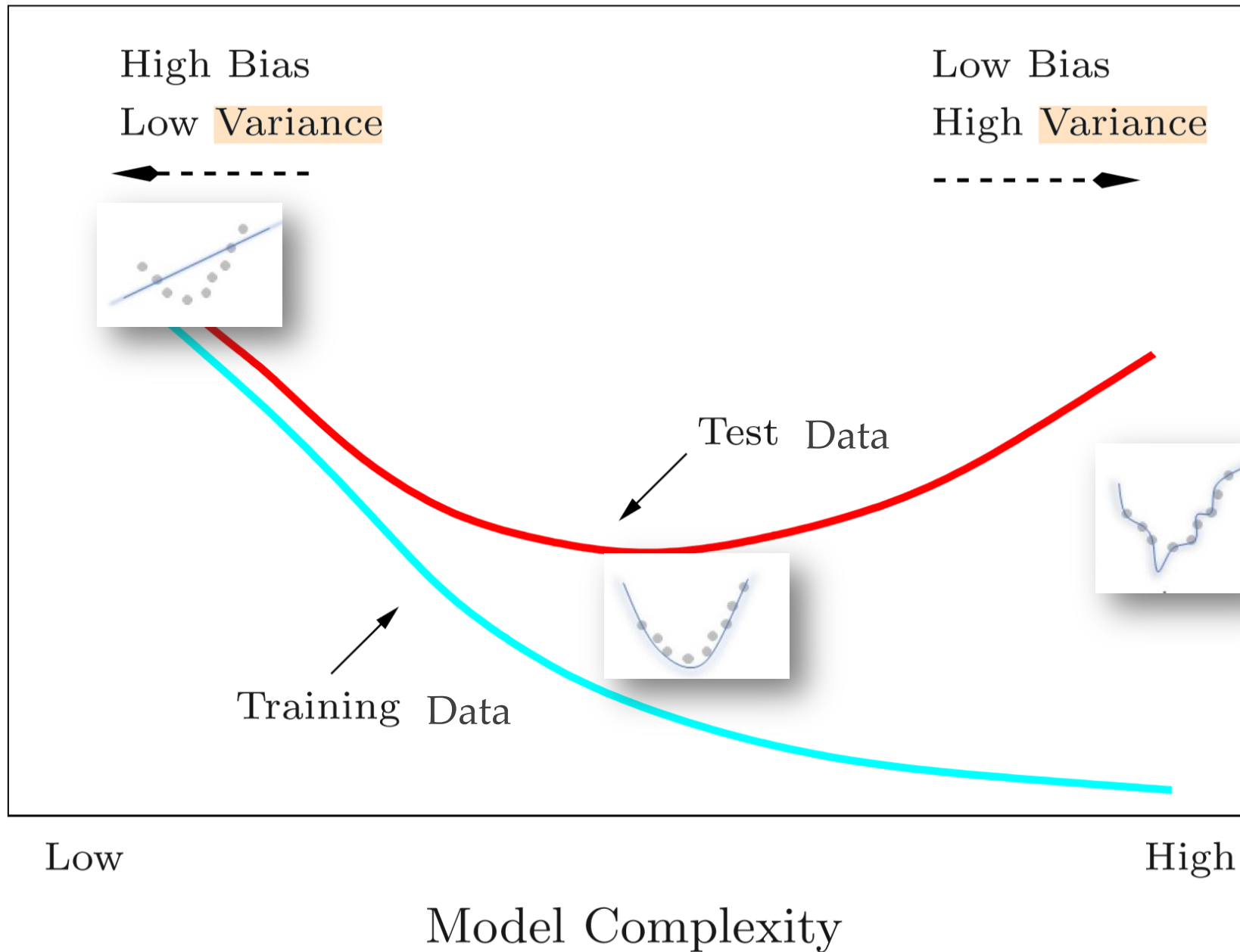
Variance error

High

What we want to make:

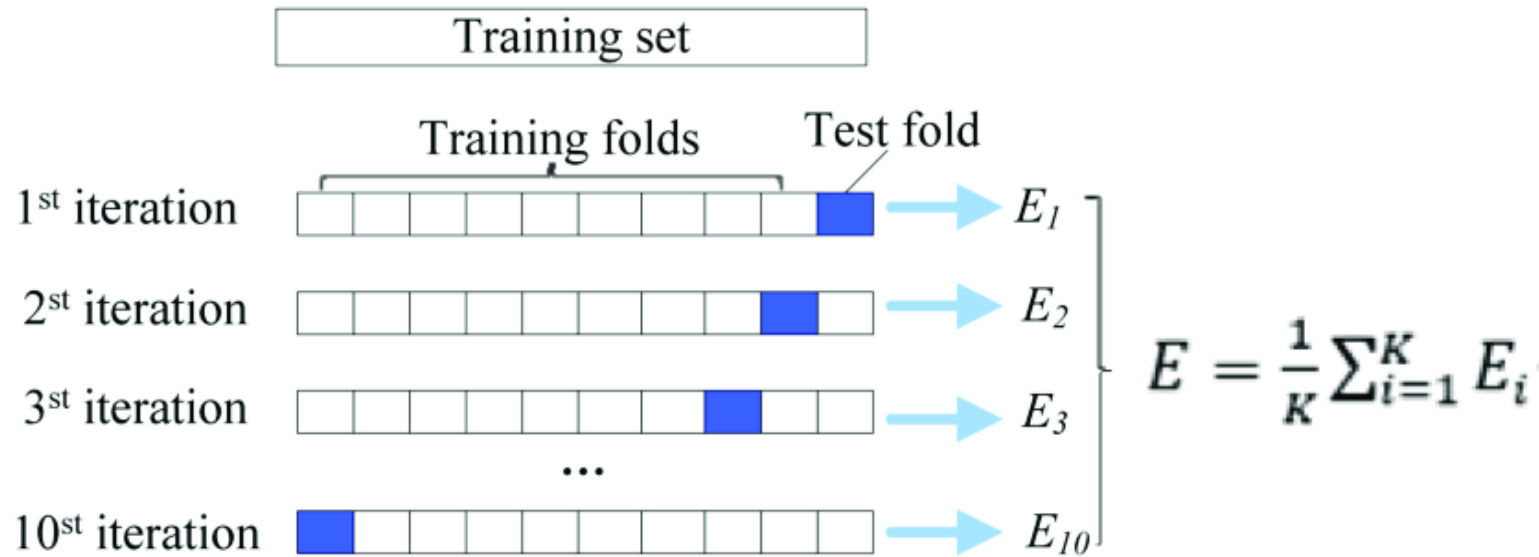


Prediction Error



# How can we select a model that balances variance & bias?

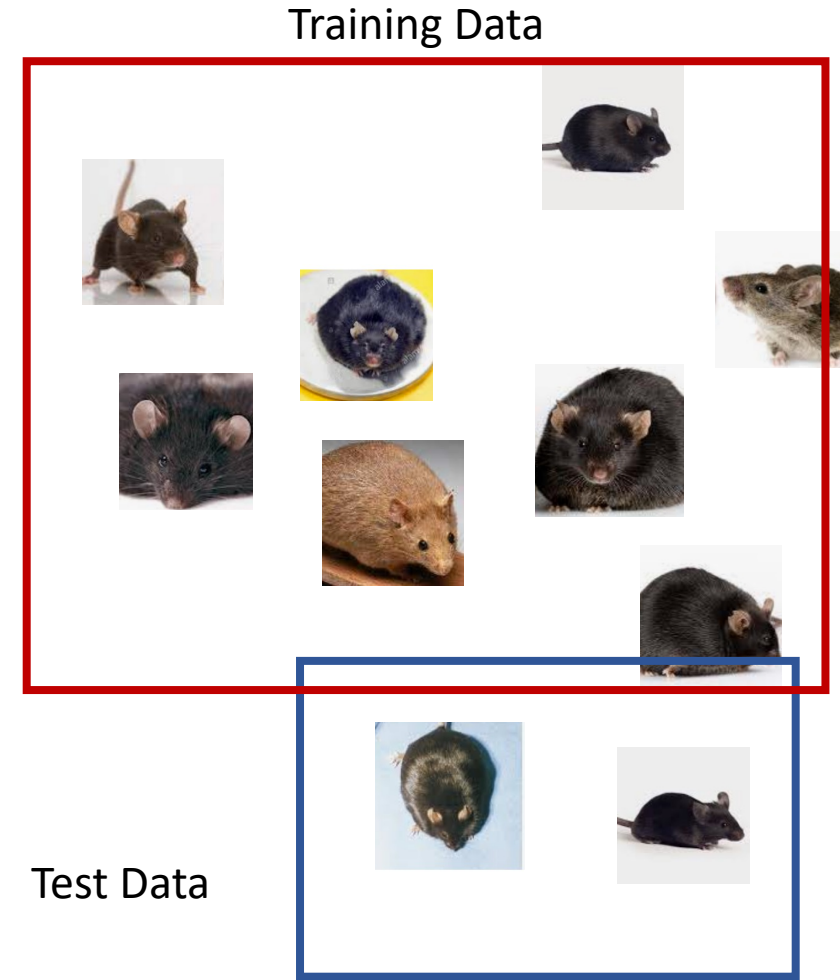
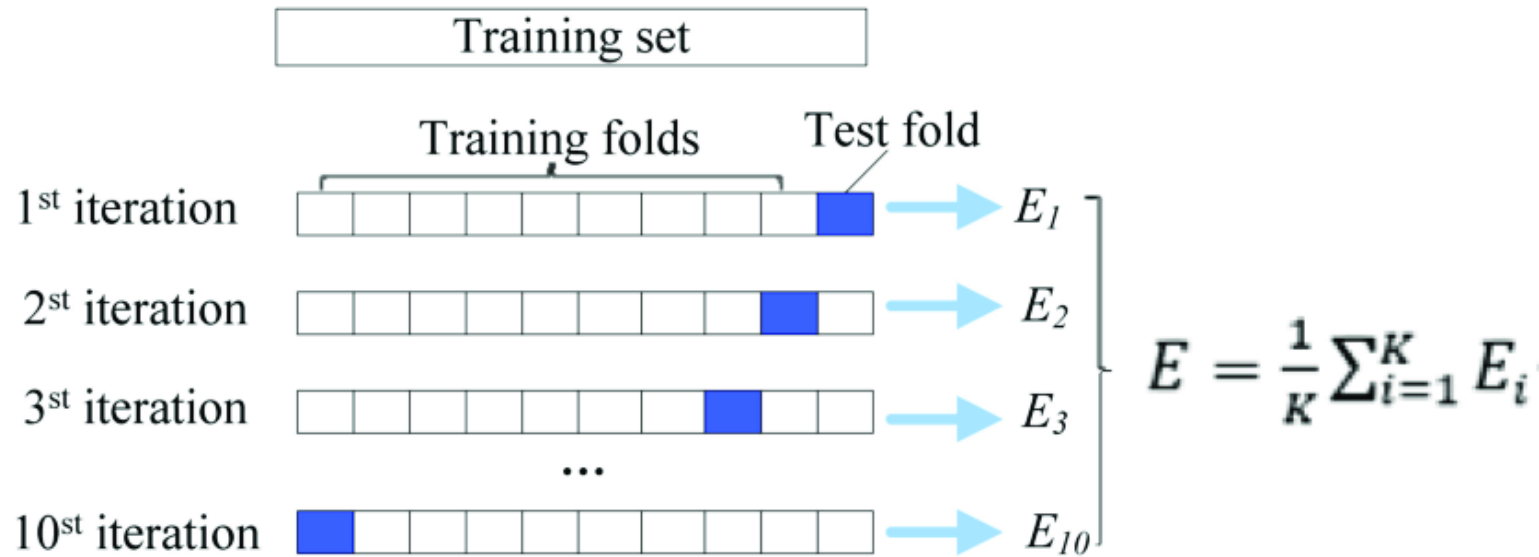
- Use cross-validation!
- Bias and variance can be balanced





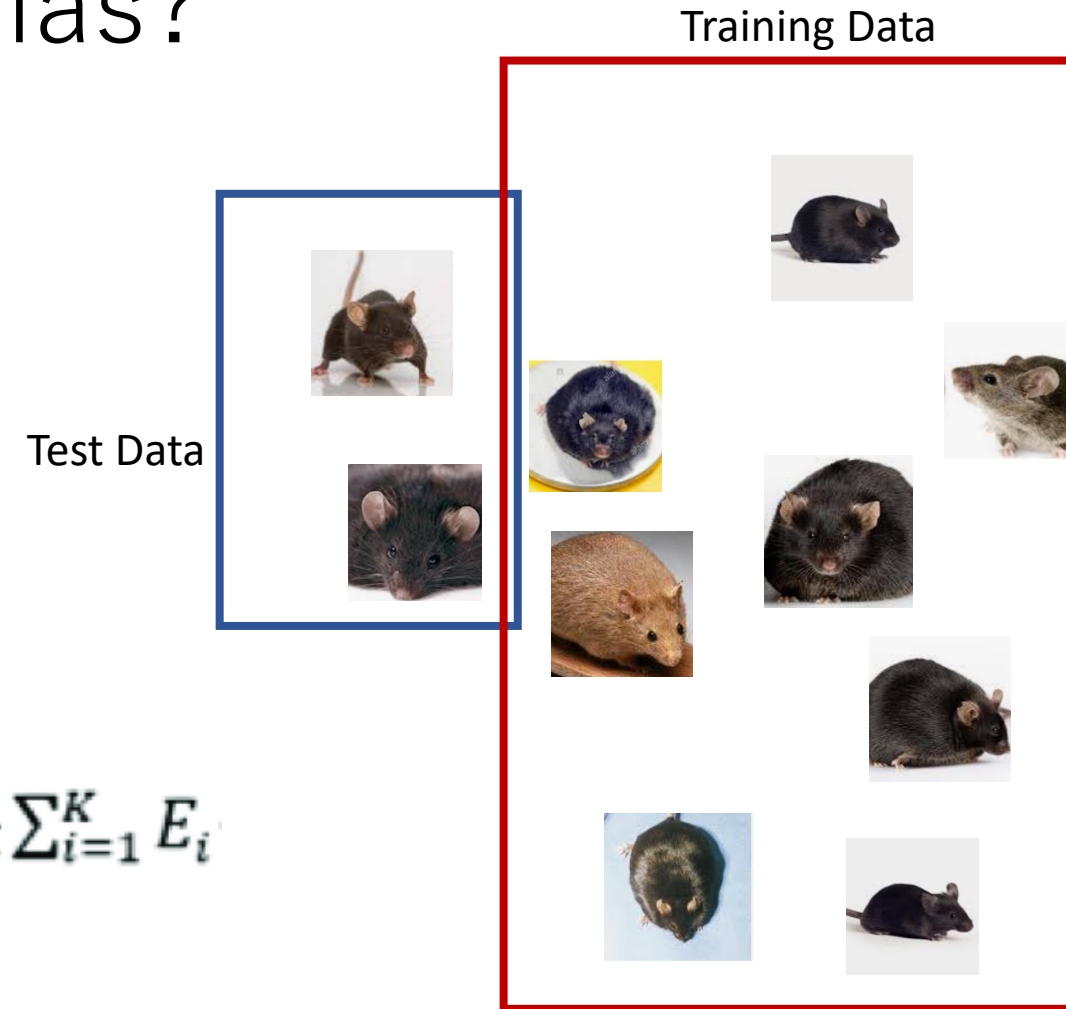
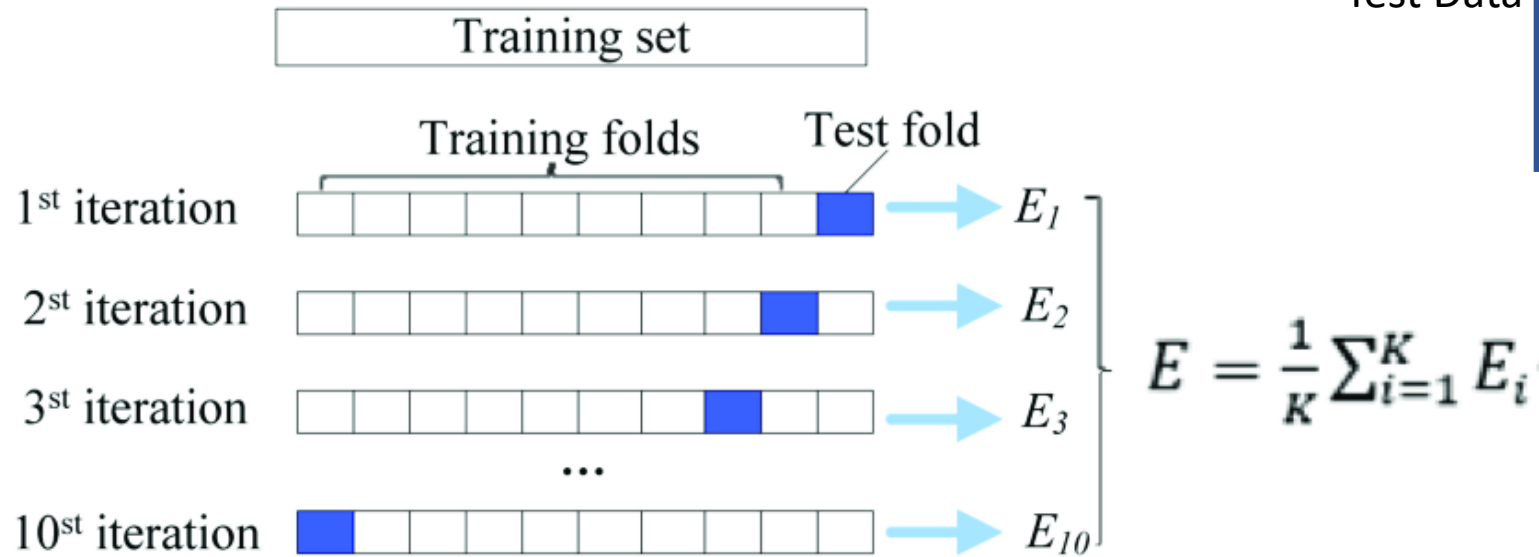
# How can we select a model that balances variance & bias?

- Use cross-validation!
- Bias and variance can be balanced



# How can we select a model that balances variance & bias?

- Use cross-validation!
- Bias and variance can be balanced



# High variance

- Overfitting
- Good performance only for the training data set
- Little generalization

## **Remedy**

- Remove attributes from the model
- More data

# High bias

- Underfitting
- Overly simple model

## **Remedy**

- Try making a more complex model
- Add more attributes to the model
- Train the model for a longer time

# What we are going to learn

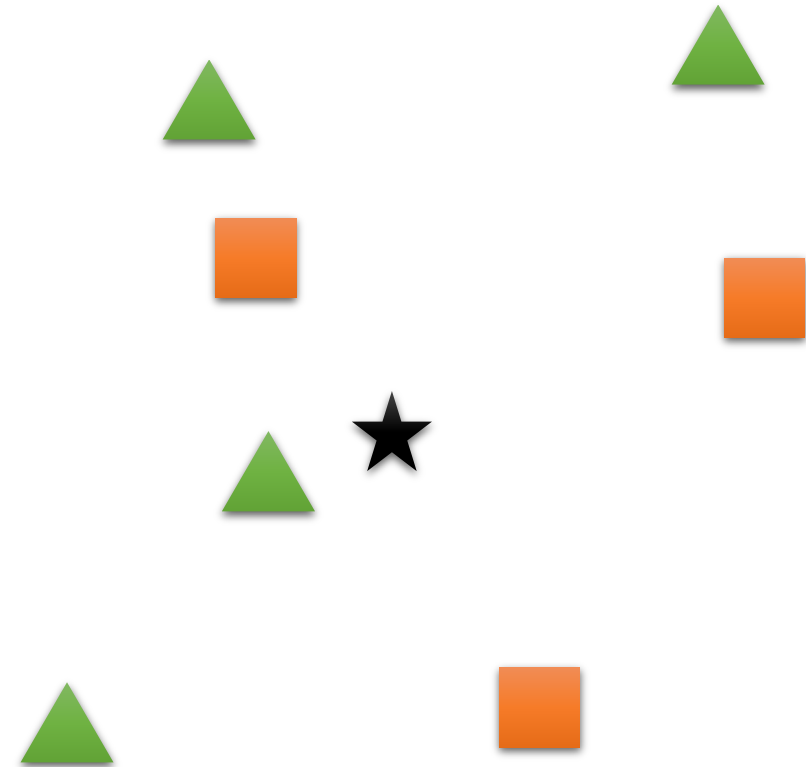
K-Nearest Neighbor

K-Means

Decision Tree

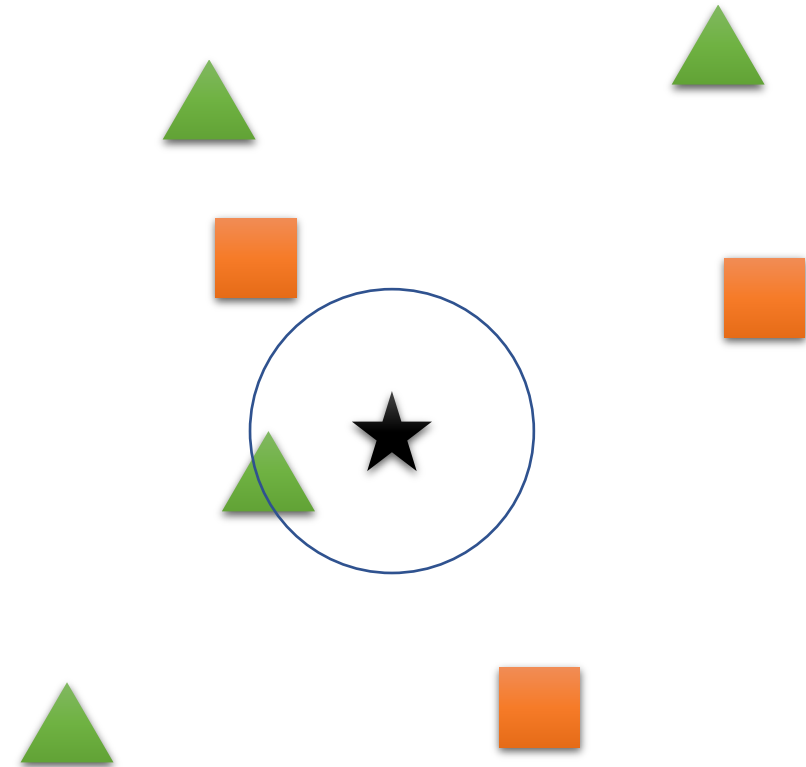
# K-nearest neighbor (kNN)

- K: #neighbor data points to consider
- Majority vote: collect cluster associations from K (user defined) neighbors and identify the most relevant association



# K-nearest neighbor (kNN)

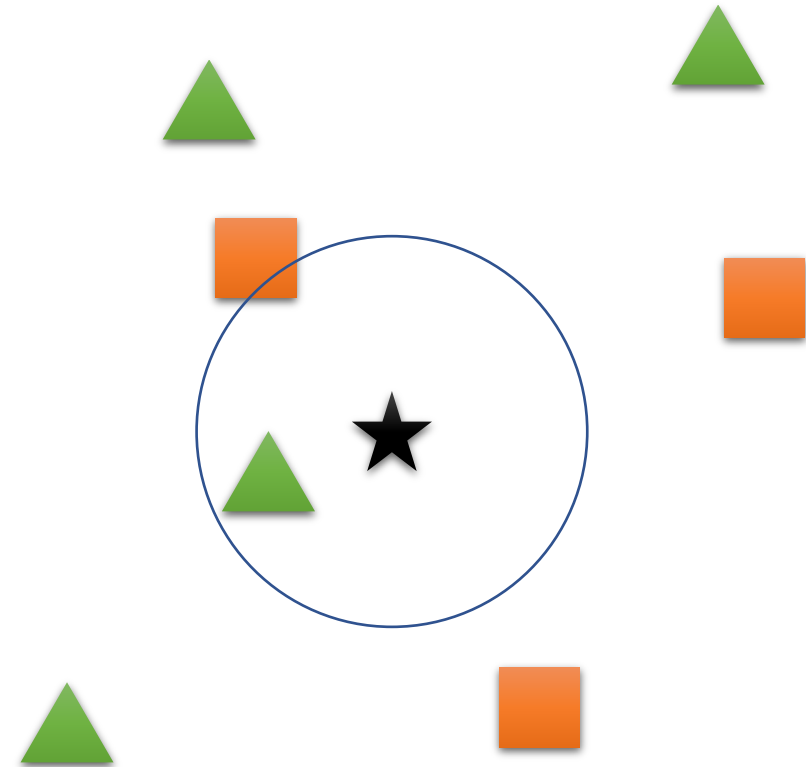
- K: #neighbor data points to consider
- Majority vote: collect cluster associations from k (User defined) neighbors and identify the most relevant association





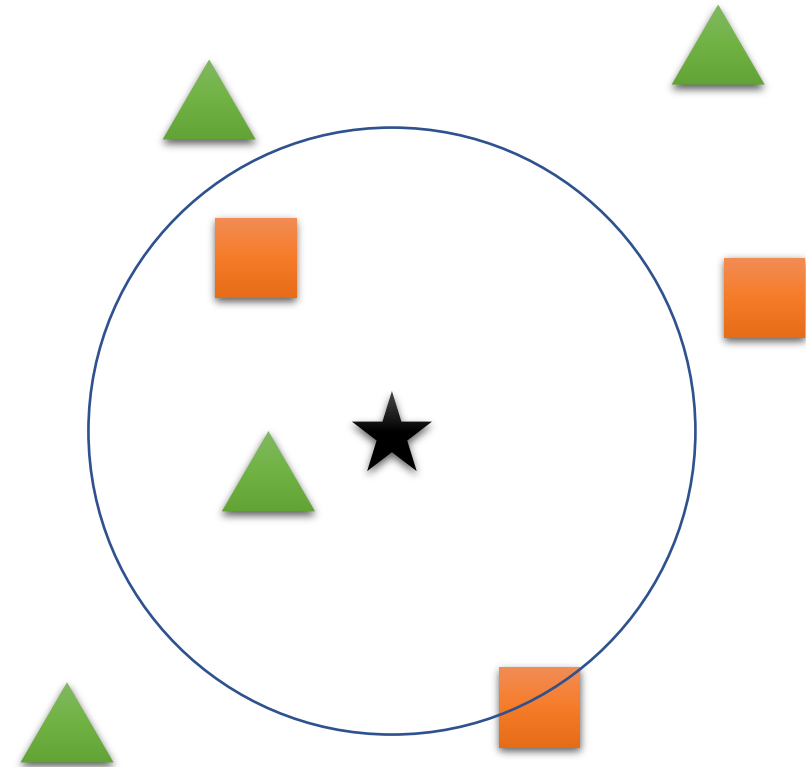
# K-nearest neighbor (kNN)

- K: #neighbor data points to consider
- Majority vote: collect cluster associations from k (User defined) neighbors and identify the most relevant association



# K-nearest neighbor (kNN)

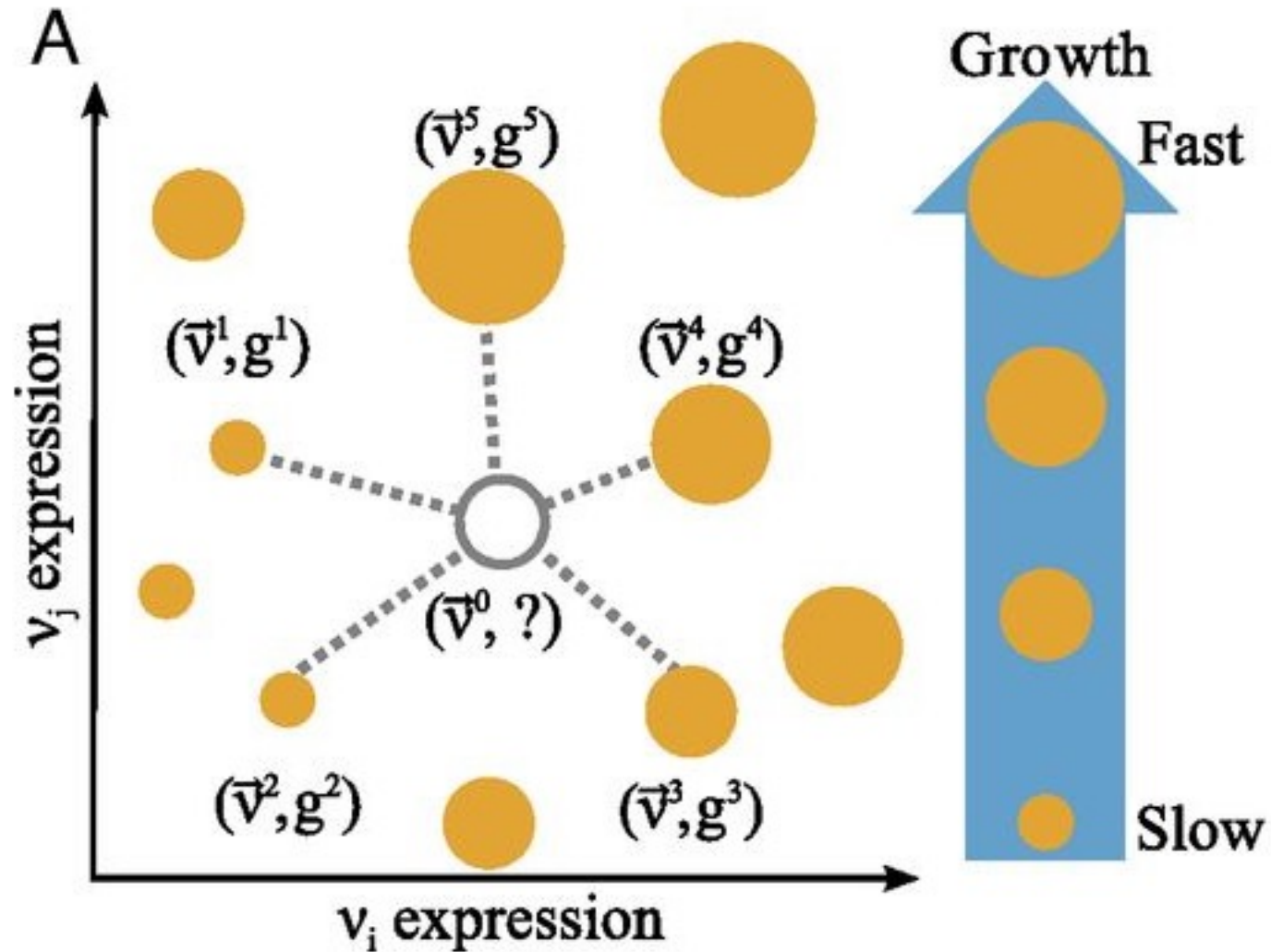
- K: #neighbor data points to consider
- Majority vote: collect cluster associations from k (User defined) neighbors and identify the most relevant association



# K-nearest neighbor (kNN) steps:

- Calculate the distance between a new data point and preexisting data points
- Sort the results in ascending order
- Choose the first K rows
- Take the majority vote from the data points within K

# kNN application example



# K-nearest neighbor (kNN)

## Pros

- Easy to implement

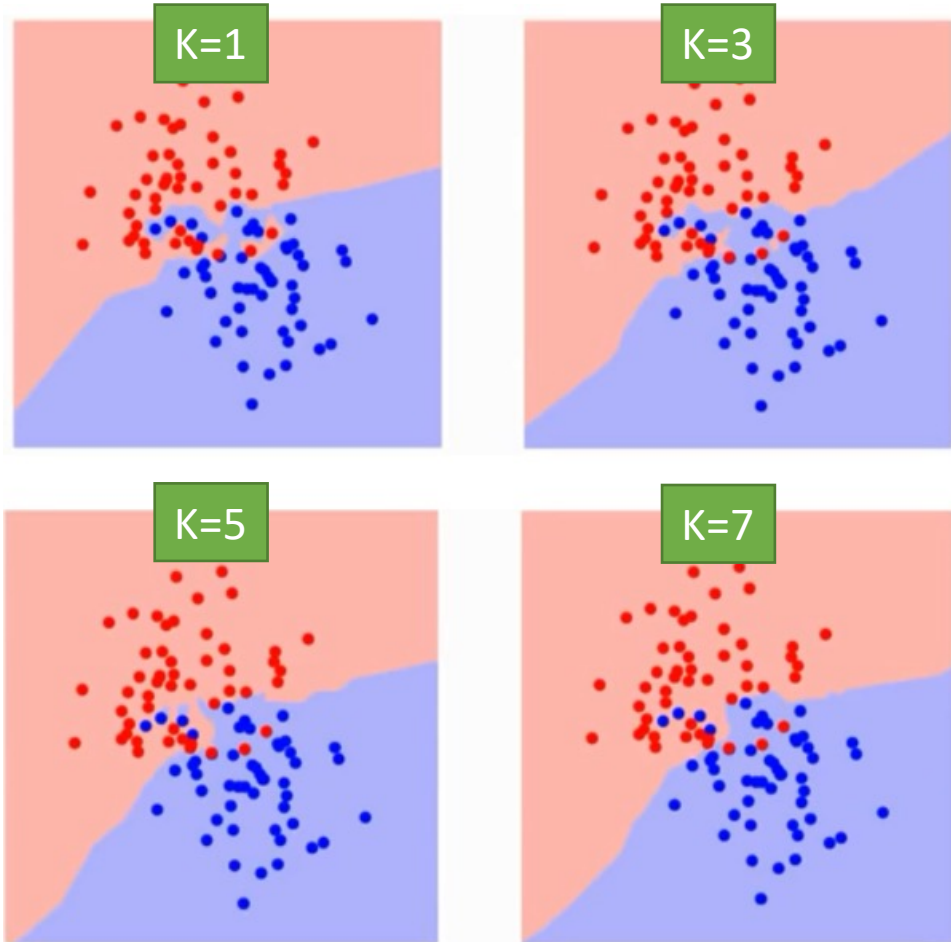
## Cons

- Computationally expensive
- Sensitive to the scale of data and outliers

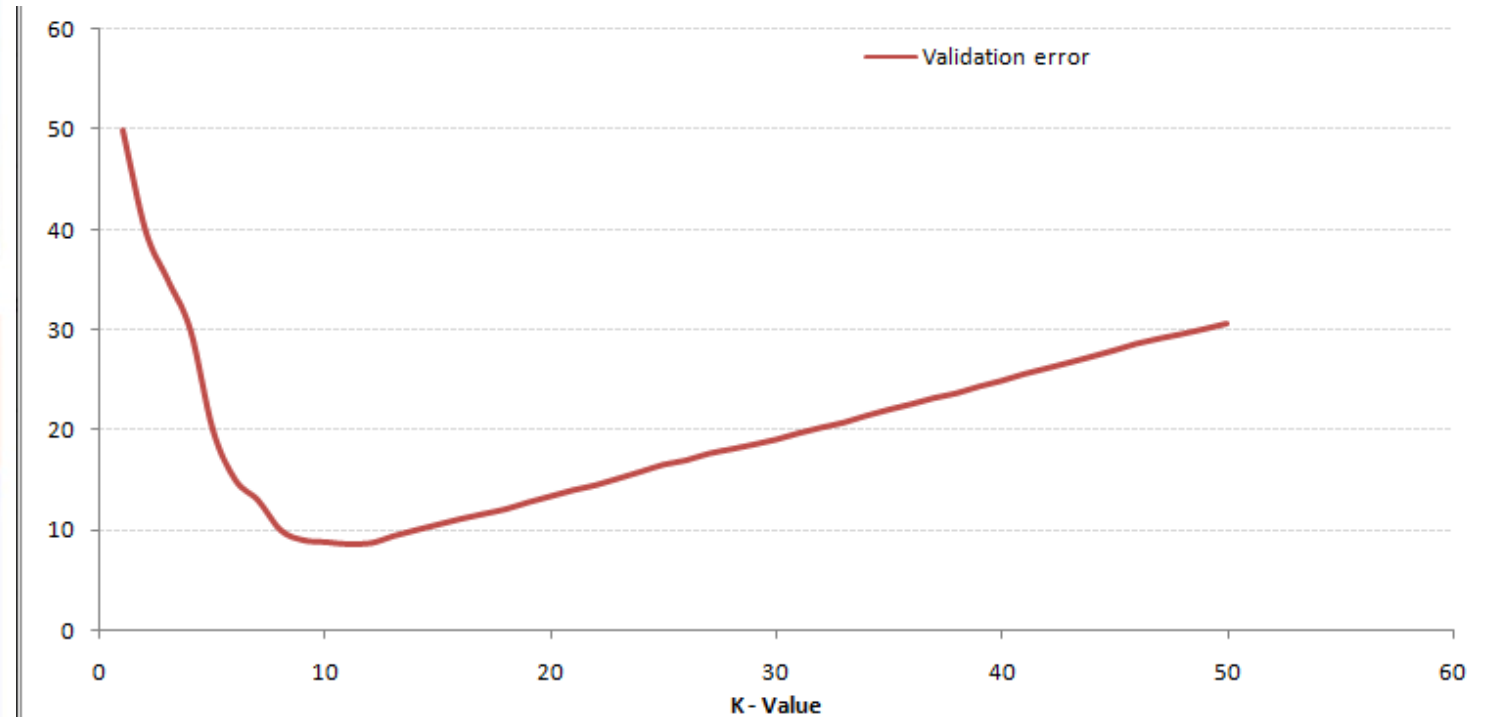
# K-nearest neighbor (kNN): Preprocessing

- Rescale data (ex:  $\frac{x-\mu}{\sigma}$ )
- Impute (substitute) missing data
- Reduce the dimension (# attributes) if the dimension is too high

# K-nearest neighbor (kNN)



Test different Ks and find the one with lowest error





# kNN practice

# What we are going to learn

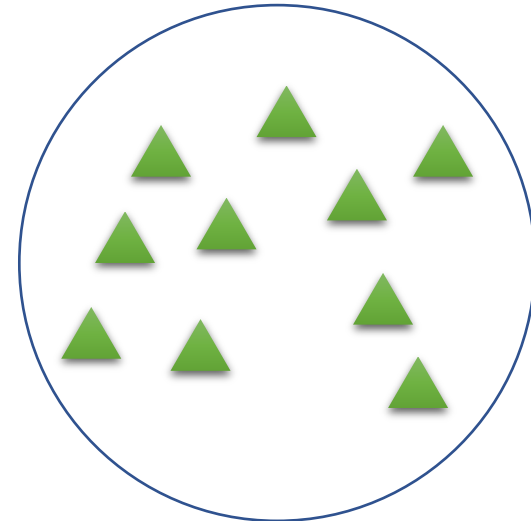
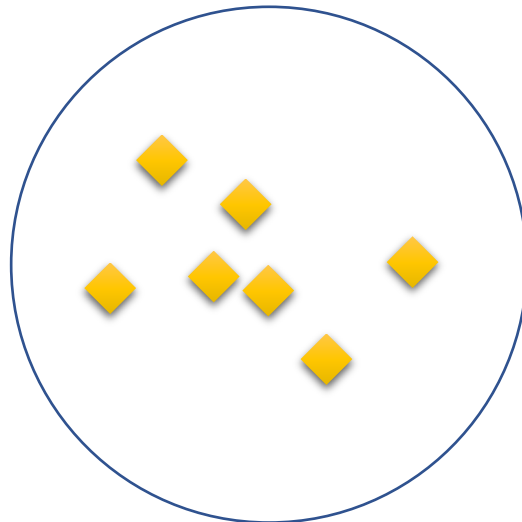
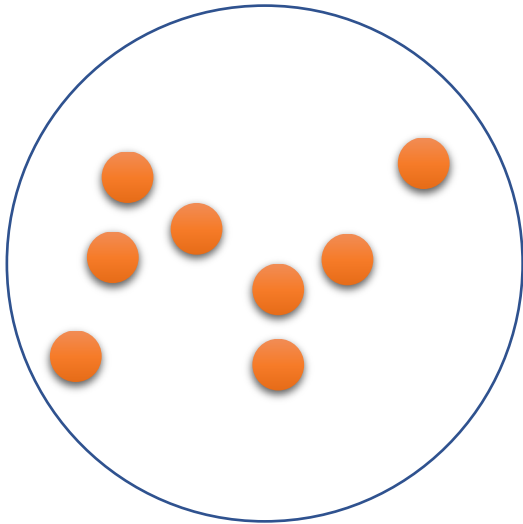
K-Nearest Neighbor

K-Means

Decision Tree

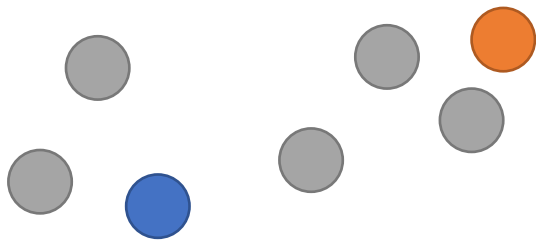
# K-means clustering

- Assign a cluster to a data point based on distance from each center (centroid) of each cluster



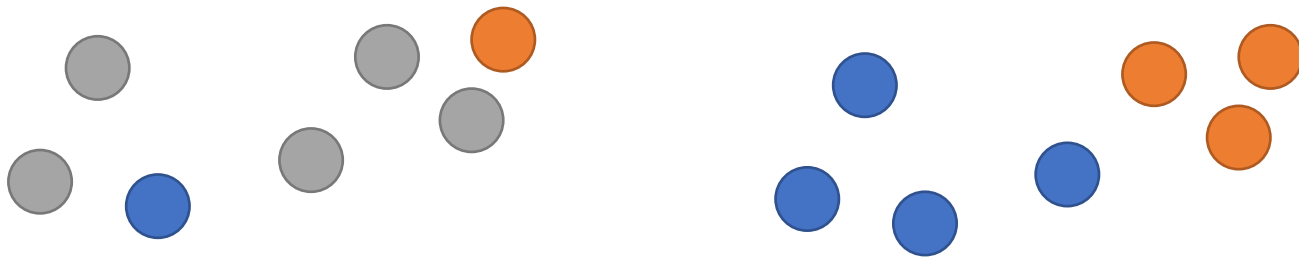
# K-means clustering steps (Lloyd):

- Start: Random centers
- Take the sum of the squared distance between data points and all centers
- Assign a cluster membership to each data point
- Compute new centers
- End: if the centers don't change OR meet iteration threshold



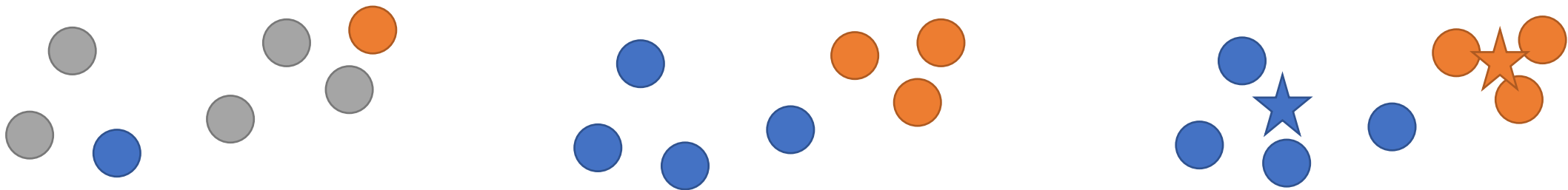
# K-means clustering steps (Lloyd):

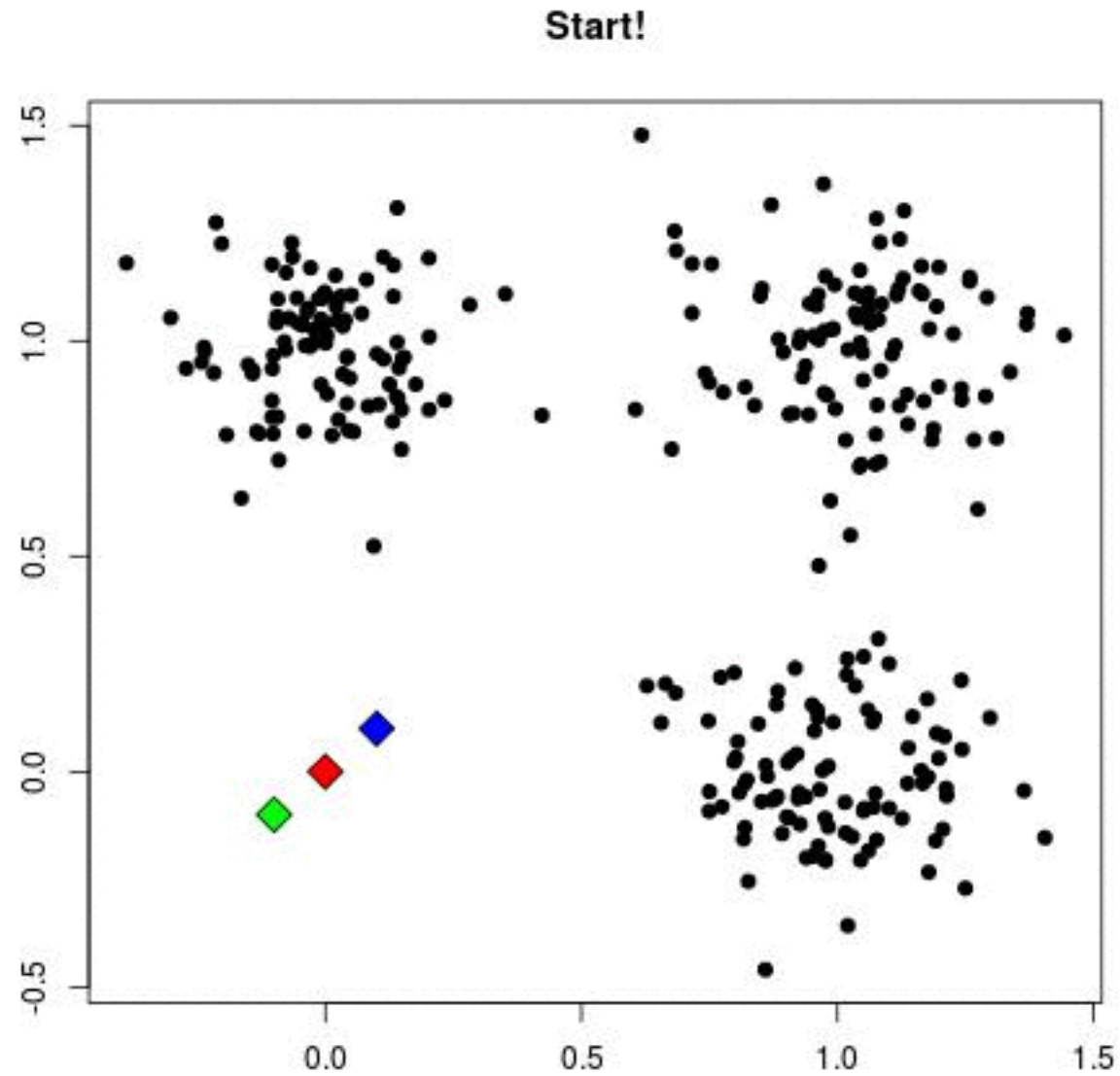
- Start: Random centers
- Take the sum of the squared distance between data points and all centers
- Assign a cluster membership to each data point
- Compute new centers
- End: if the centers don't change OR meet iteration threshold



# K-means clustering steps (Lloyd):

- Start: Random centers
- Take the sum of the squared distance between data points and all centers
- Assign a cluster membership to each data point
- Compute new centers
- End: if the centers don't change OR meet iteration threshold

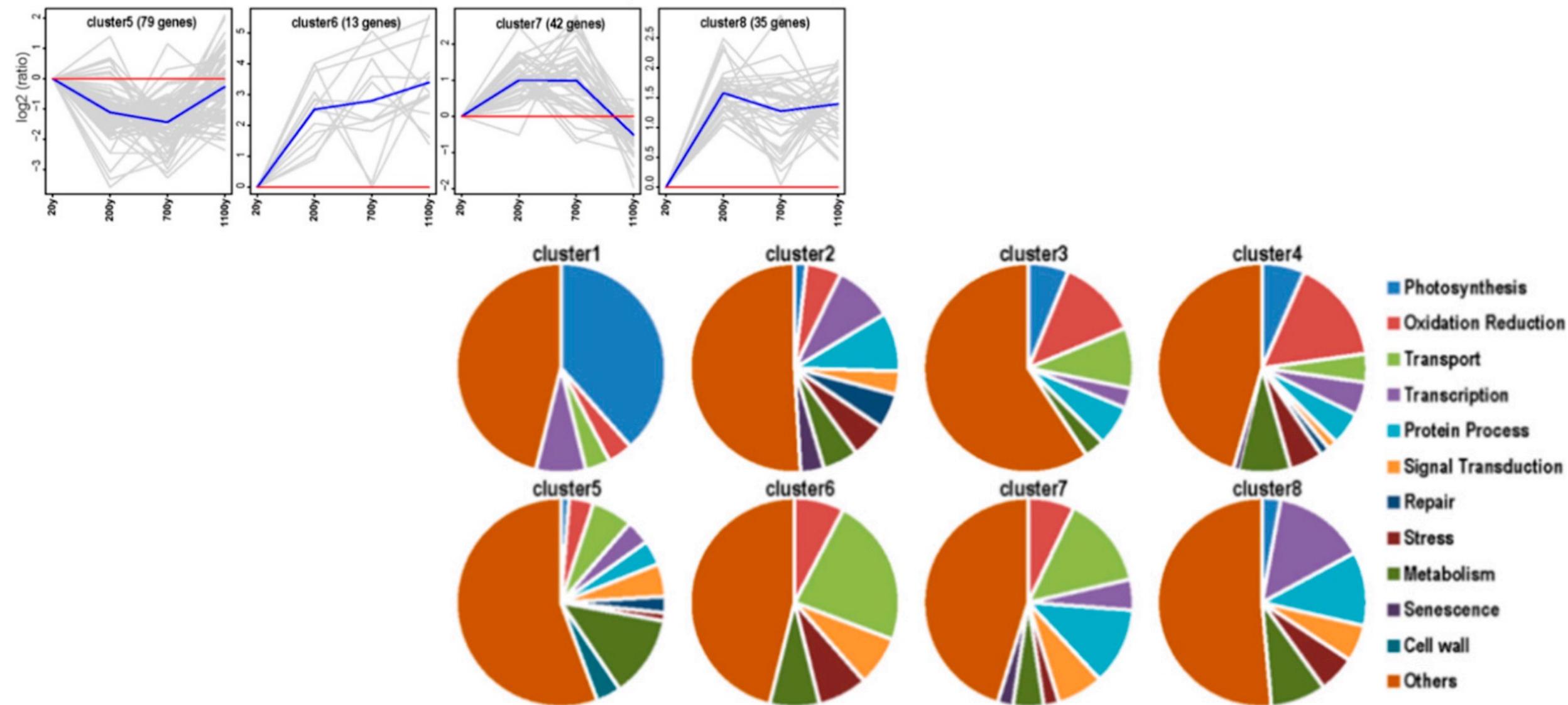




<https://raw.githubusercontent.com/andrewxiechina/DataScience/master/K-Means/k4Xcap1.gif>



# K-means cluster application example



# K-means clustering

## Pros:

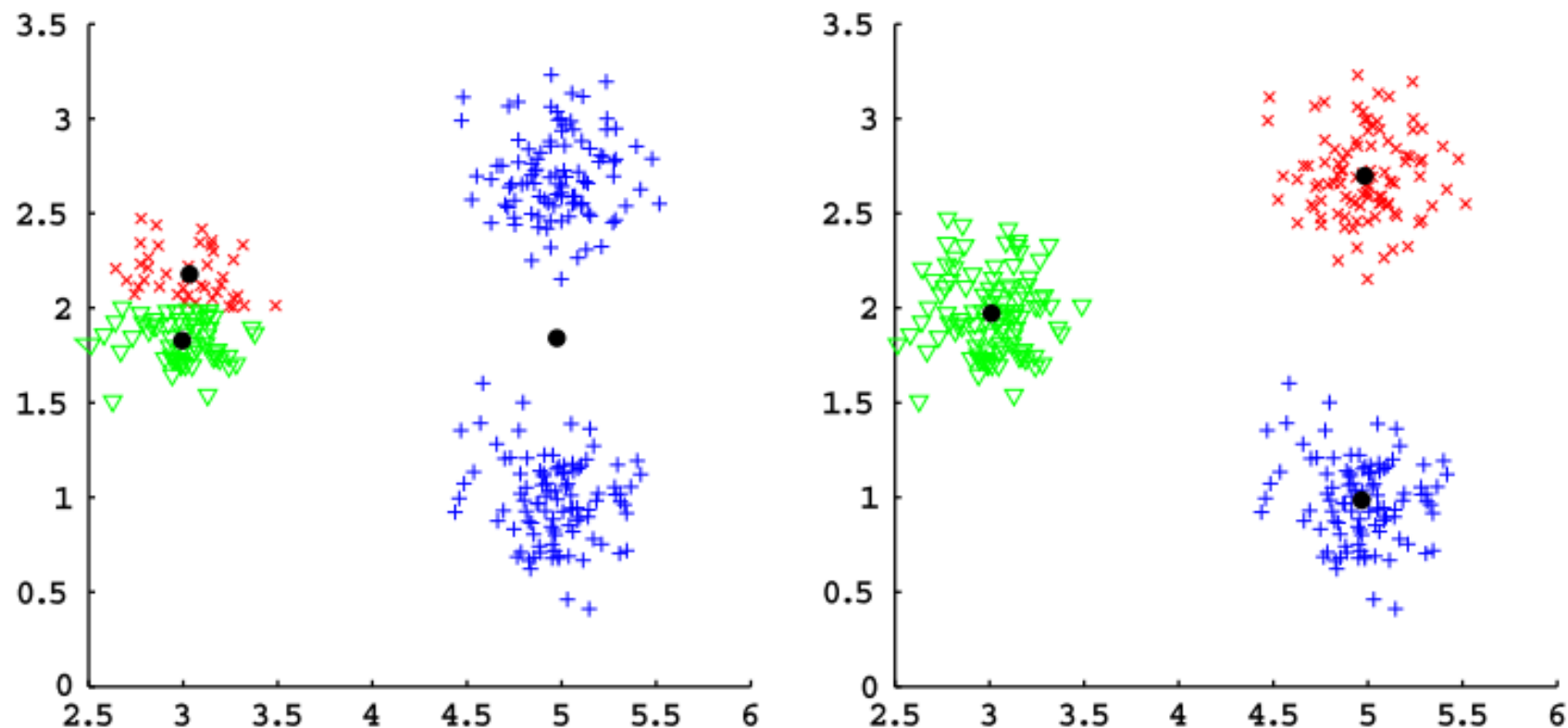
- Simple and fast

## Cons:

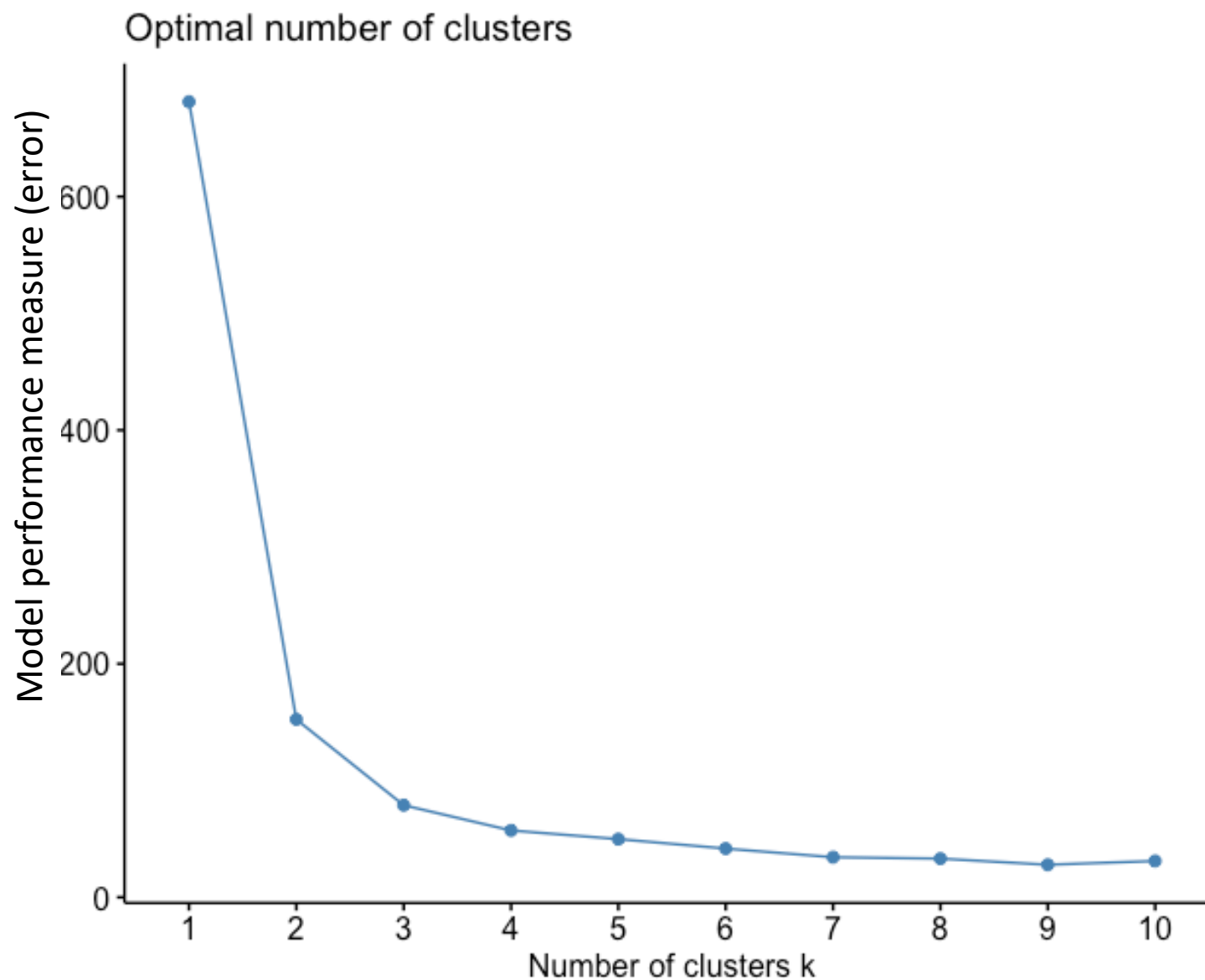
- Sensitive to the initial random selection of centers
- Computationally expensive
- Sensitive to the scale of data and outliers

# K-means clustering

- Initialization matters!



# K-means clustering, determine k



within-cluster sum of squares

$$\sum_{k=1}^K \sum_{i \in S_k} \sum_{j=1}^P (x_{ij} - \bar{x}_{kj})^2$$

where  $k$  is the  $k$ th cluster,  $S_k$  is the cluster set of the  $k$ th cluster, and  $j$  is the  $j$ th element of each data point

# Survey time & Coffee break

<https://www.surveymonkey.com/r/F75J6VZ>

# K-means practice

# What we are going to learn

K-Nearest Neighbor

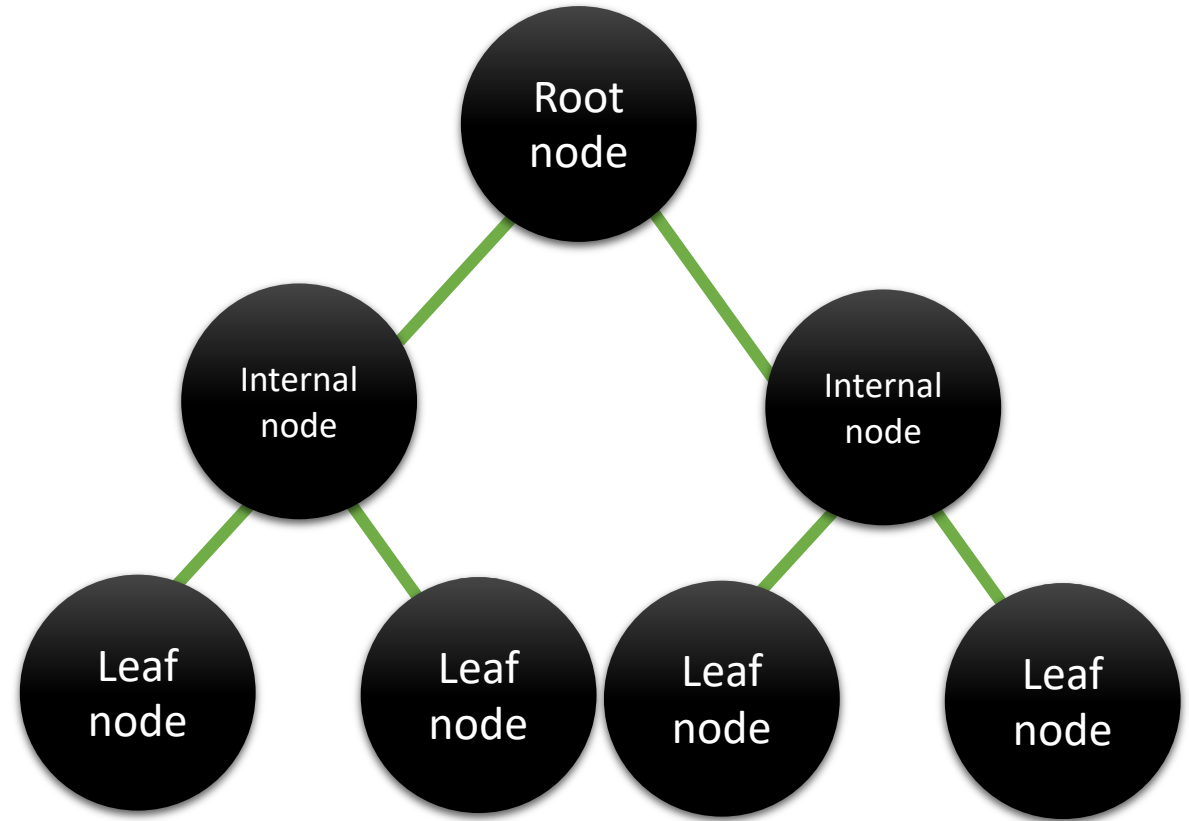
K-Means

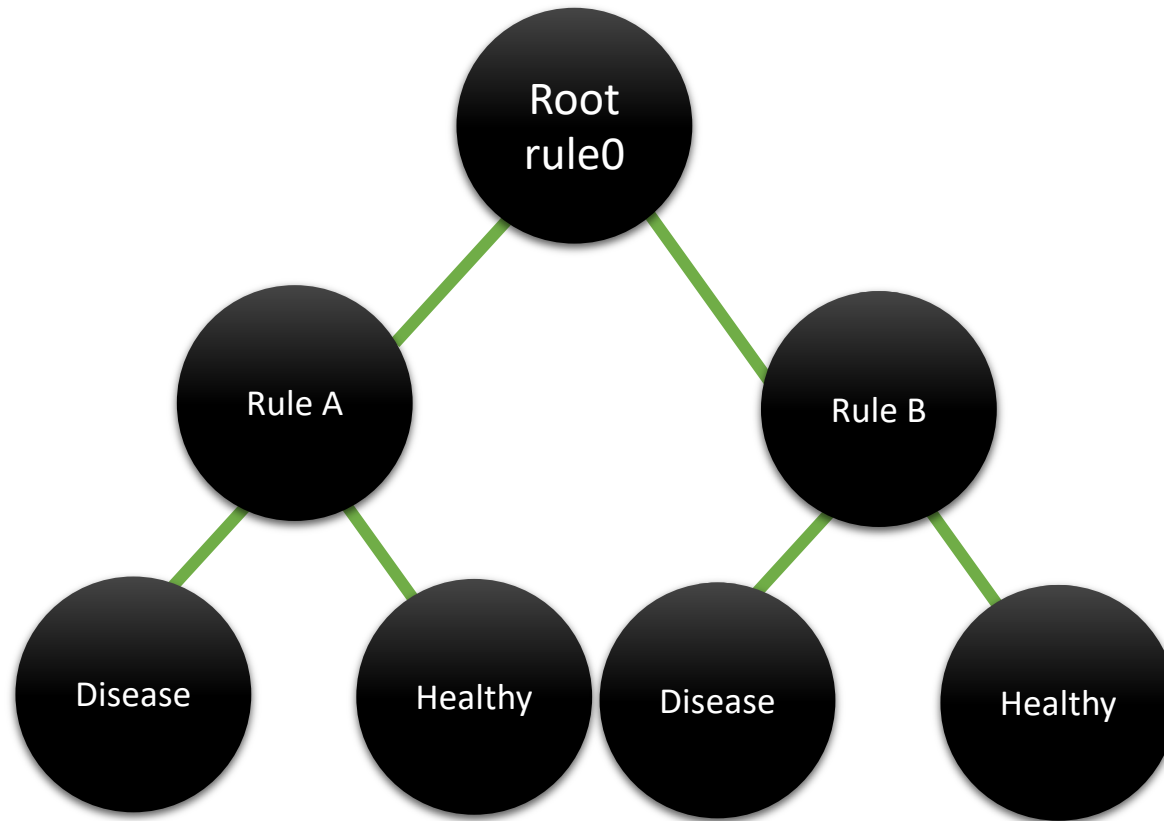
Decision Tree



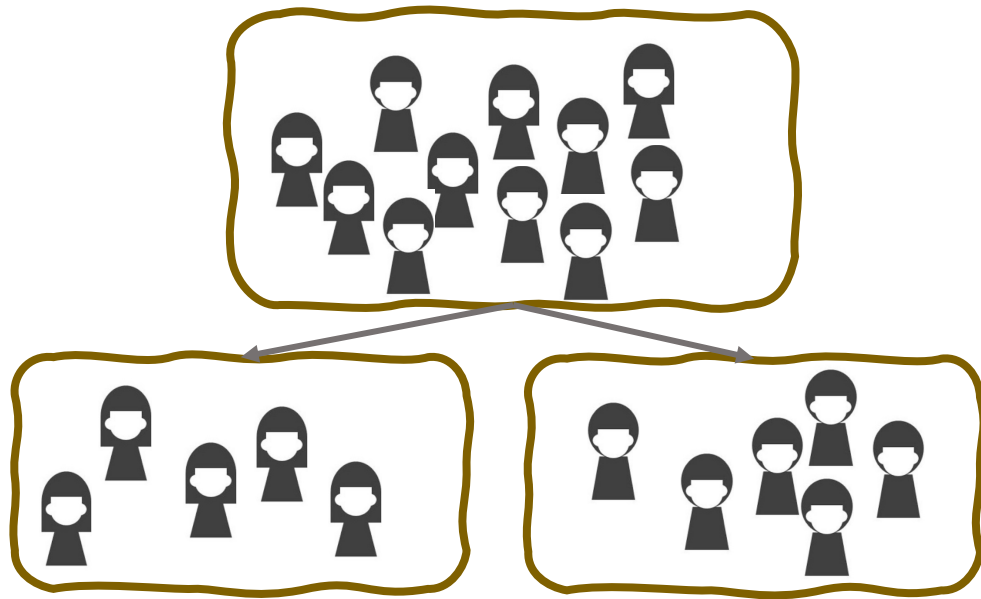
# Decision Tree

- Node: attributes
- Leaf node: label
- Edge: connection between nodes





# Decision Tree: which feature to use?

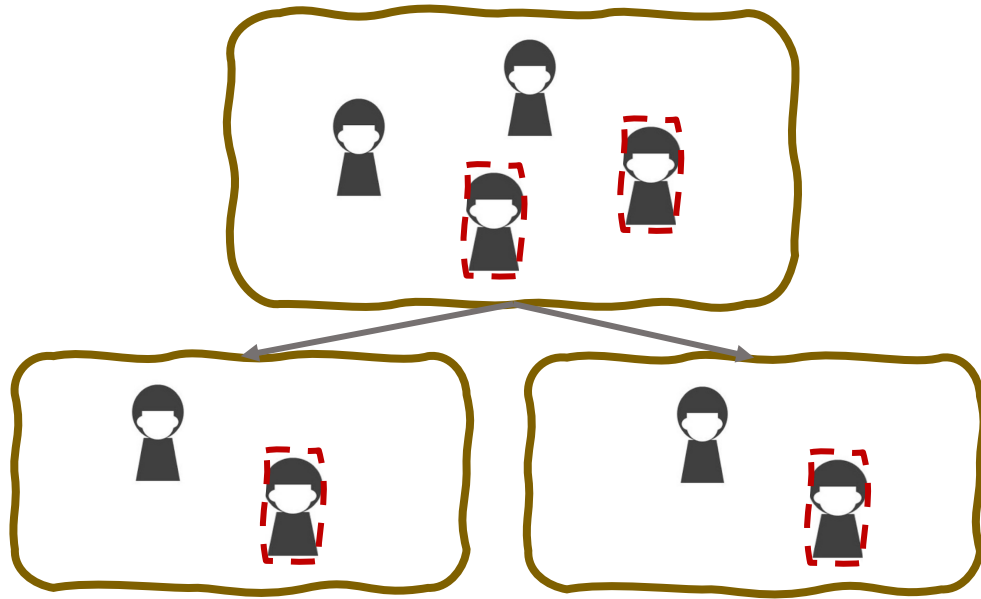


## Evaluation

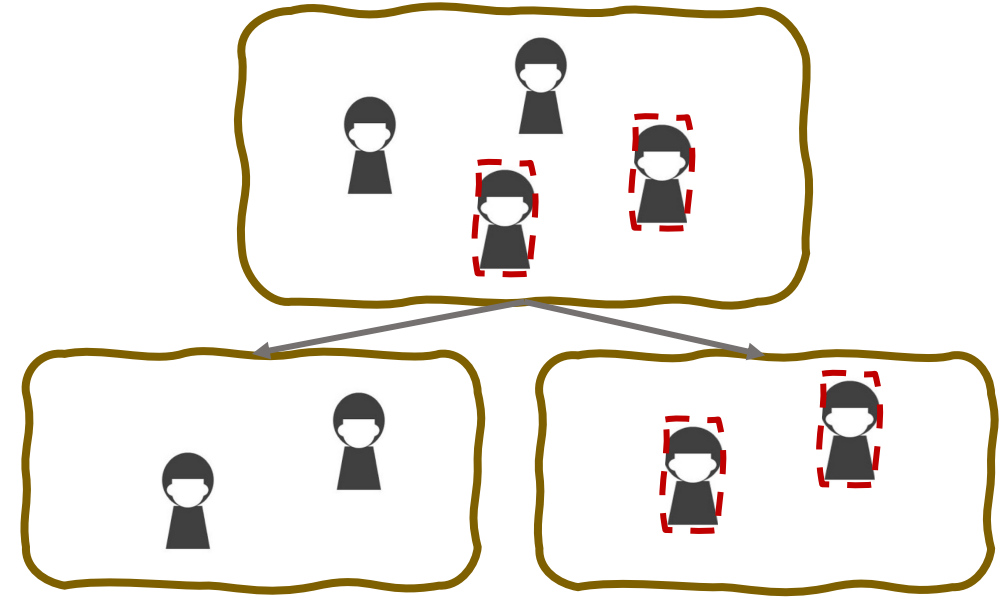
**Gini Index:**  $1 - \sum_{i=1}^m p_i^2$

**Information Gain :** Entropy(before split) – Entropy(after split)  
Entropy:  $-\sum_{i=1}^m p_i \log_2(p_i)$

# Decision Tree: which feature to use?



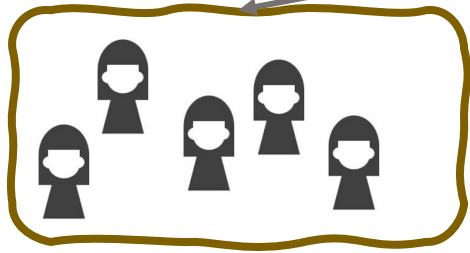
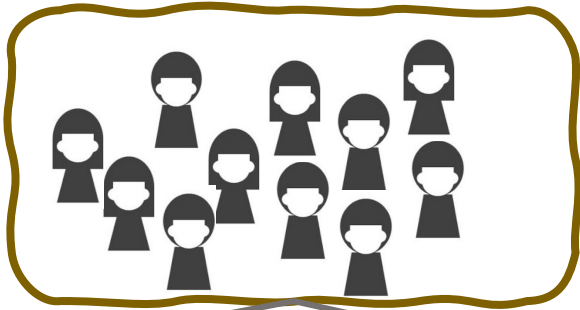
Gini index: High  
Information Gain: Low



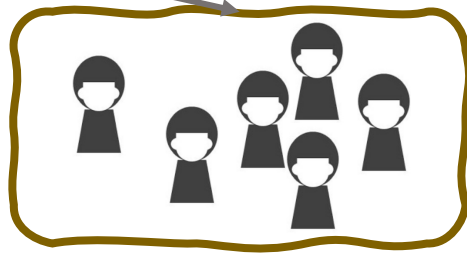
Gini index: Low  
Information Gain: High

# Decision Tree: which feature to use?

Gender  
Total: 11  
Case: 5



F  
Total: 5  
Case: 2



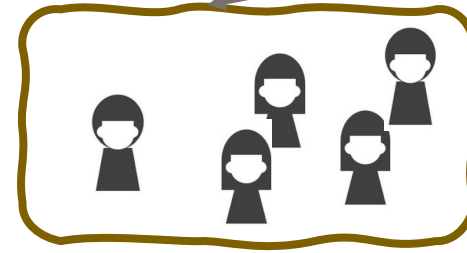
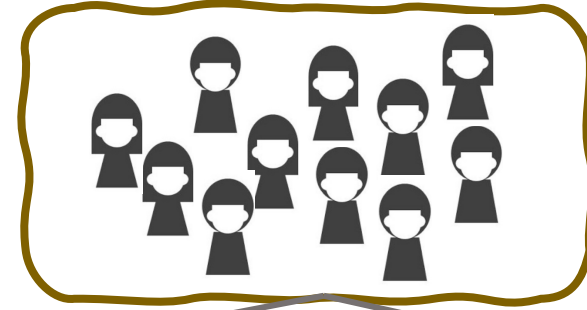
M  
Total: 6  
Case: 3

Gini index:  $1 - (2/5)^2 - (3/5)^2$

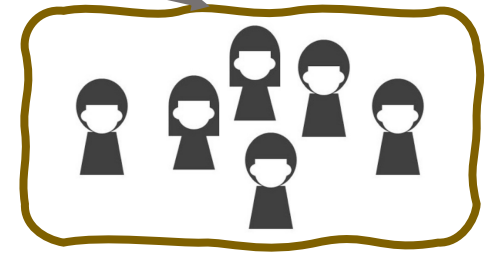
$1 - (3/6)^2 - (3/6)^2$

Weighted sum: 0.49

Age



40 >  
Total: 5  
Case: 1



40 <=  
Total: 6  
Case: 4

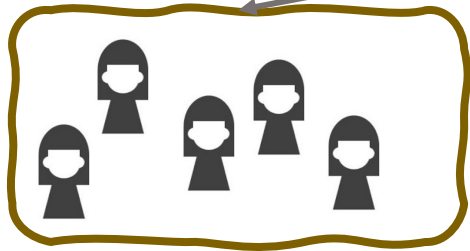
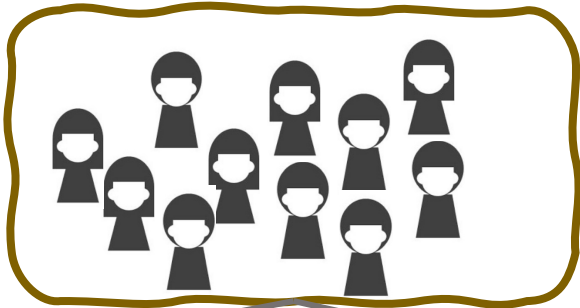
$1 - (1/5)^2 - (4/5)^2$

$1 - (4/6)^2 - (2/6)^2$

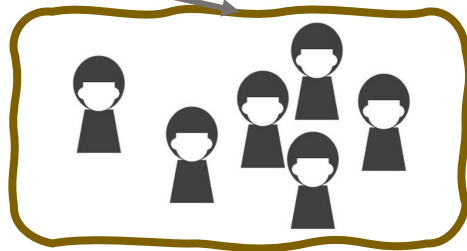
Weighted sum: 0.39

# Decision Tree: which feature to use?

Gender  
Total: 11  
Case: 5



F  
Total: 5  
Case: 2

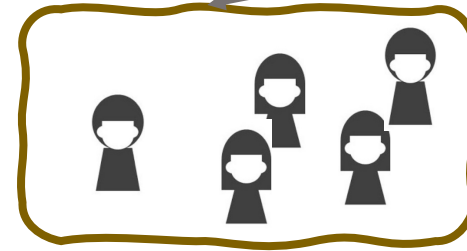
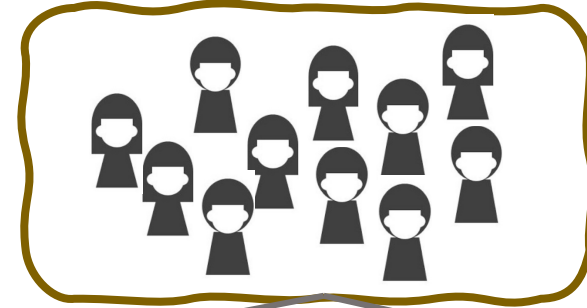


M  
Total: 6  
Case: 3

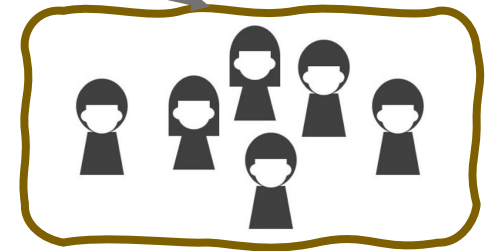
Gini index:  $1 - (2/5)^2 - (3/5)^2$       $1 - (3/6)^2 - (3/6)^2$

Weighted sum: 0.49

Age



40 >  
Total: 5  
Case: 1

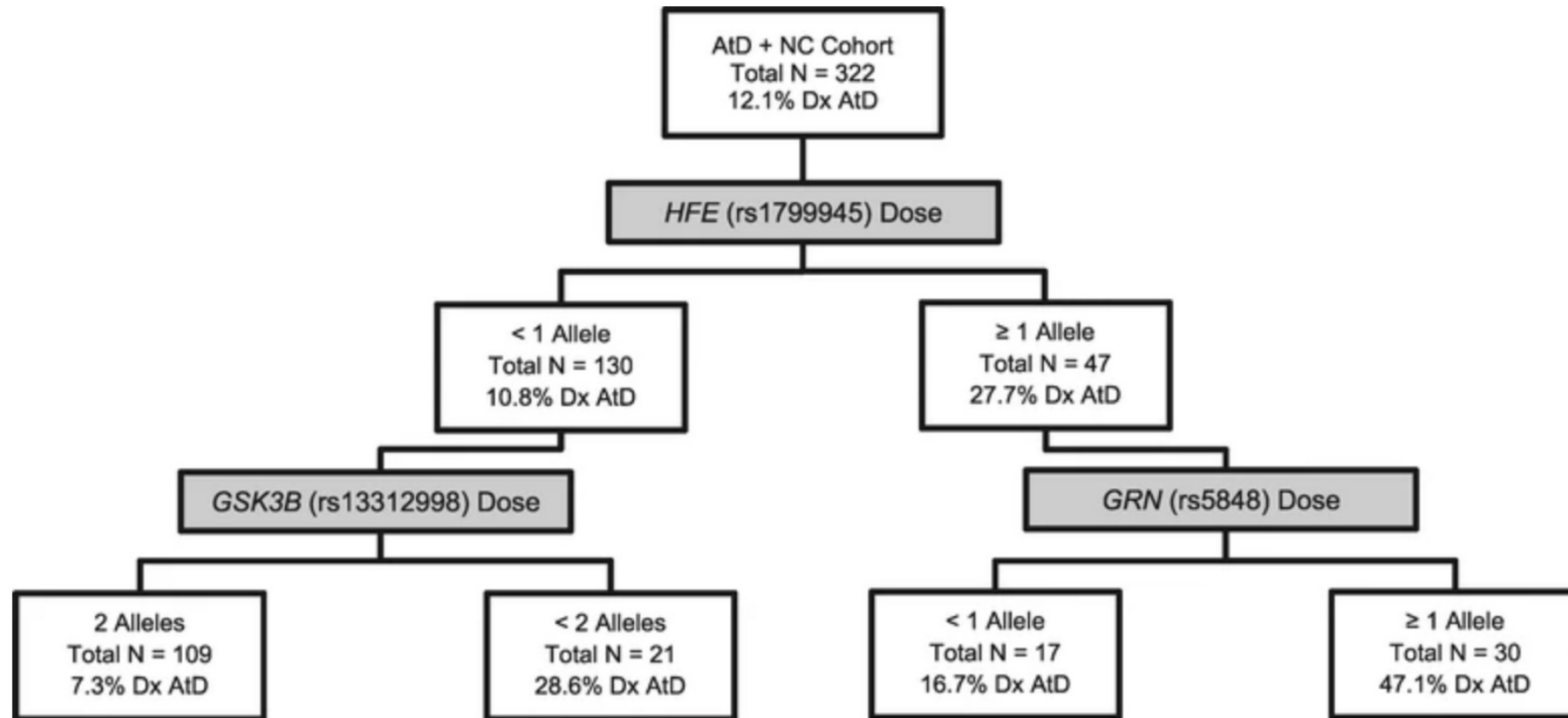


40 ≤  
Total: 6  
Case: 4

$1 - (1/5)^2 - (4/5)^2$       $1 - (4/6)^2 - (2/6)^2$

Weighted sum: 0.39

# Decision Tree application example



# Decision Tree

## Pros

- Easy interpretation
- Not influenced by outliers

## Cons

- Higher computational time for training and to process data
- Overfitting => pruning
- Selection bias toward attributes with many possible splits  
(Use Gain Ratio)



# Complexity parameter (cp)

Governs the minimum “benefit” that must be gained at each split of the decision tree in order to make a split worthwhile.

Williams G. (2011) Decision Trees. In: Data Mining with Rattle and R. Use R. Springer, New York, NY

Trims off least important splits at each run.  
Penalizes models that are too complex.

Muhammad Azam et al. (2017) Simulation and Computation, 46:4, 2924-2934

# Decision Tree practice

# More advanced methods

Support Vector Machine

Random Forest

Neural Network

The background is a solid dark teal color. It features abstract, wavy, light teal lines that create a sense of movement and depth. Overlaid on these waves is a pattern of small, light teal dashes or grid lines, which are more densely packed in some areas and more sparse in others, adding a textured, digital feel to the overall design.

Thank you!