ARTIFICIAL INTELLIGENCE NEEDS REAL INTELLIGENCE

Clustering I

Professor: Tianyuan ZHANG
tianyuan.zhang@kedgebs.com

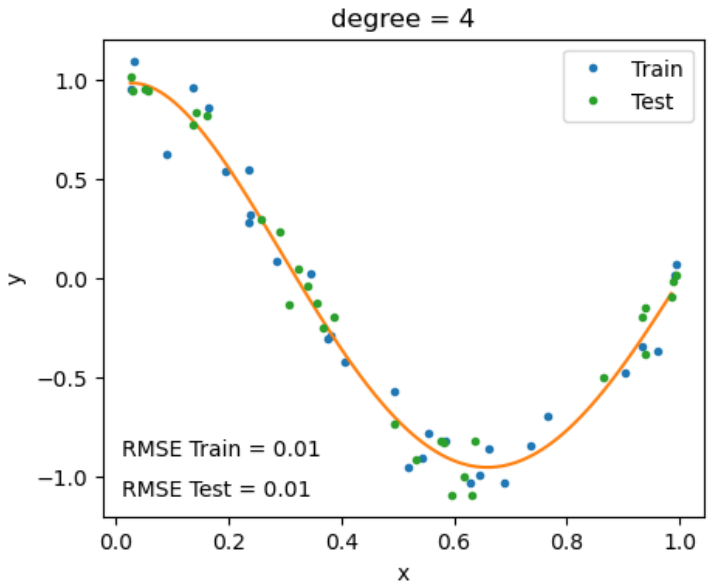
KEDGE BUSINESS SCHOOL

Recap of Supervised Learning

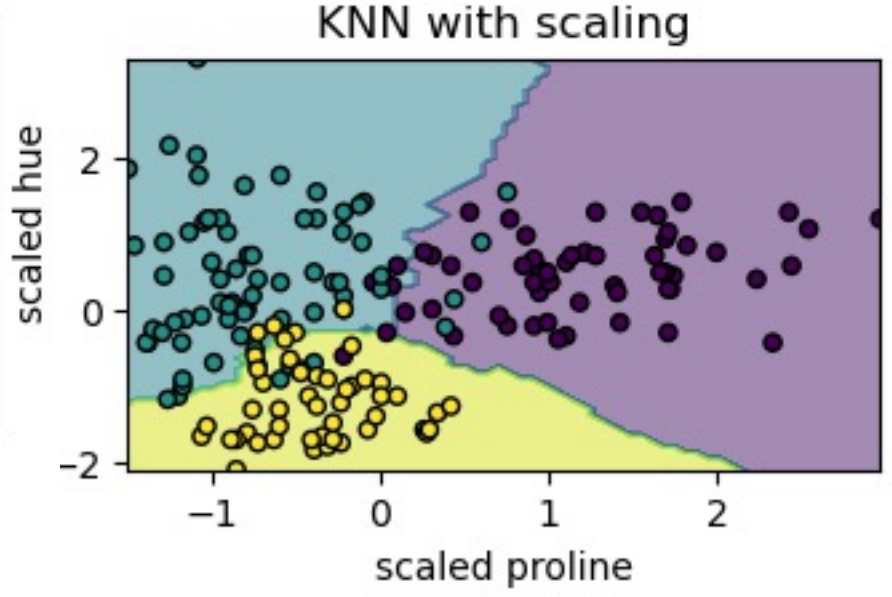
- Supervised learning
  - The algorithm is presented with labeled examples
  - The goal is to learn the hidden pattern between input features and output labels
  - After seeing lots of labeled examples, the model can make predictions on unseen unlabeled data
  - Two prediction tasks
    - Regression
      - Predict a continuous numeric value
    - Classification
      - Predict discrete categories or
      - Predict probabilities of the input example belongs to each category

Recap of Supervised Learning

- Regression
  - Estimate the relationship between features and labels by fitting to the training data



- Classification
  - Learn the way to divide the feature space into different parts, each represent a category



Recap of Supervised Learning

- Regression
  - Estimate the relationship between features and labels by fitting to the training data
  - Algorithms
    - Simple linear regression
    - Multiple linear regression
    - Polynomial regression

Single input feature

- Classification
  - Learn the way to divide the feature space into different parts, each represent a category
  - Algorithms
    - Logistic regression
    - K-Nearest neighbors
    - Support vector machine
    - Decision tree

Recap of Supervised Learning

- Regression
  - Estimate the relationship between features and labels by fitting to the training data
  - Algorithms
    - Simple linear regression
    - Multiple linear regression
    - Polynomial regression

Multiple input feature

- Classification
  - Learn the way to divide the feature space into different parts, each represent a category
  - Algorithms
    - Logistic regression
    - K-Nearest neighbors
    - Support vector machine
    - Decision tree

Recap of Supervised Learning

- Regression
  - Estimate the relationship between features and labels by fitting to the training data
  - Algorithms
    - Simple linear regression
    - Multiple linear regression
    - Polynomial regression

Nonlinear relationship between features and target

- Classification
  - Learn the way to divide the feature space into different parts, each represent a category
  - Algorithms
    - Logistic regression
    - K-Nearest neighbors
    - Support vector machine
    - Decision tree

Recap of Supervised Learning

- Regression
  - Estimate the relationship between features and labels by fitting to the training data
  - Algorithms
    - Simple linear regression
    - Multiple linear regression
    - Polynomial regression

- Classification
  - Learn the way to divide the feature space into different parts, each represent a category
  - Algorithms
    - Logistic regression
    - K-Nearest neighbors
    - Support vector machine
    - Decision tree

Estimate probabilities ∈ [0,1]
Binary classification

Recap of Supervised Learning

- Regression
  - Estimate the relationship between features and labels by fitting to the training data
  - Algorithms
    - Simple linear regression
    - Multiple linear regression
    - Polynomial regression
- Classification
  - Learn the way to divide the feature space into different parts, each represent a category
  - Algorithms
    - Logistic regression
    - K-Nearest neighbors
    - Support vector machine
    - Decision tree

A voting system;
Classify new data based on the classes of its nearest neighbors;
Directly applicable for multi-class

Recap of Supervised Learning

- Regression
  - Estimate the relationship between features and labels by fitting to the training data
  - Algorithms
    - Simple linear regression
    - Multiple linear regression
    - Polynomial regression
- Classification
  - Learn the way to divide the feature space into different parts, each represent a category
  - Algorithms
    - Logistic regression
    - K-Nearest neighbors
    - Support vector machine
    - Decision tree

Find a hyperplane with the maximum margin;
Use kernel trick to solve nonlinearly separable problem;
Binary classification

Recap of Supervised Learning

- Regression
  - Estimate the relationship between features and labels by fitting to the training data
  - Algorithms
    - Simple linear regression
    - Multiple linear regression
    - Polynomial regression
- Classification
  - Learn the way to divide the feature space into different parts, each represent a category
  - Algorithms
    - Logistic regression
    - K-Nearest neighbors
    - Support vector machine
    - Decision tree

White box model, a collection of conditions organized as a tree;
Prone to over-fitting, need to perform pruning
Directly applicable for multi-class

KEDGE BUSINESS SCHOOL

Recap of Supervised Learning

- Regression metric
  - Measure the error of predictions
    - Residual
    - MSE
    - RMSE
    - MAE
    - R^2
  - Affected by outliers

- Classification metric
  - Measure the correctness of predictions
    - Accuracy
    - Confusion matrix
      - Precision
      - Recall
      - F1-score
    - ROC curve
    - Area under ROC curve

Class-wise metric
- *Macro average*
- *Weighted average*

Recap of Supervised Learning

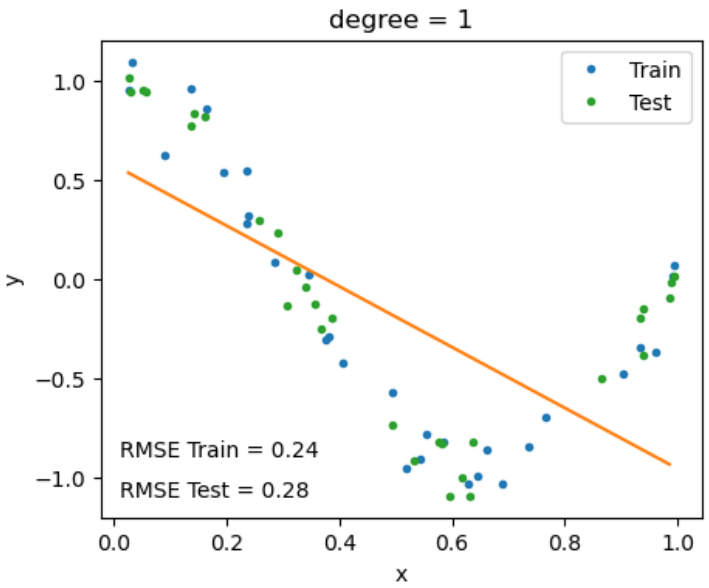
- Regression metric
  - Measure the error of predictions
    - Residual
    - MSE
    - RMSE
    - MAE
    - R^2
  - Affected by outliers

- Classification metric
  - Measure the correctness of predictions
    - Accuracy
    - Confusion matrix
      - Precision
      - Recall
      - F1-score
    - ROC curve
    - Area under ROC curve

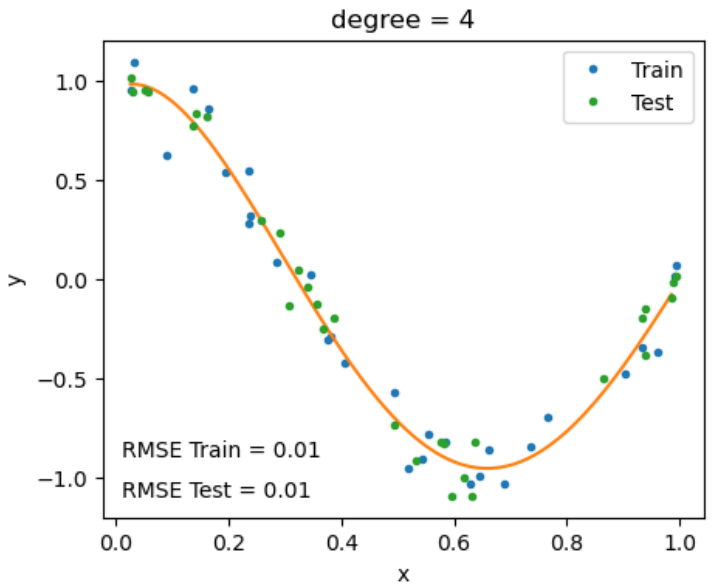
- *Class-wise*
- *Regardless threshold*

Recap of Supervised Learning

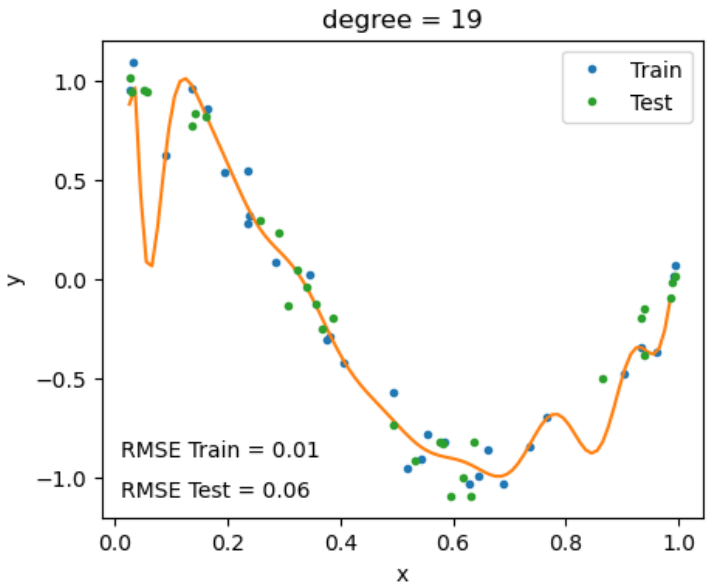
- Under-fitting & Over-fitting
  - Regression



Under-fitting



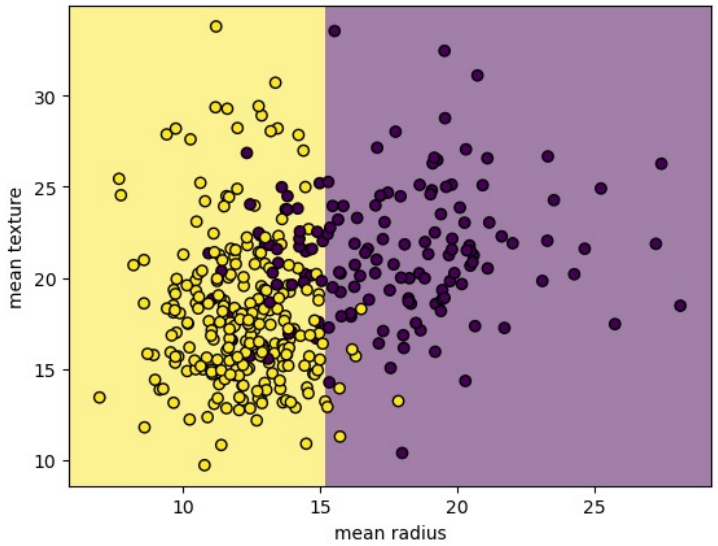
Well-fitting



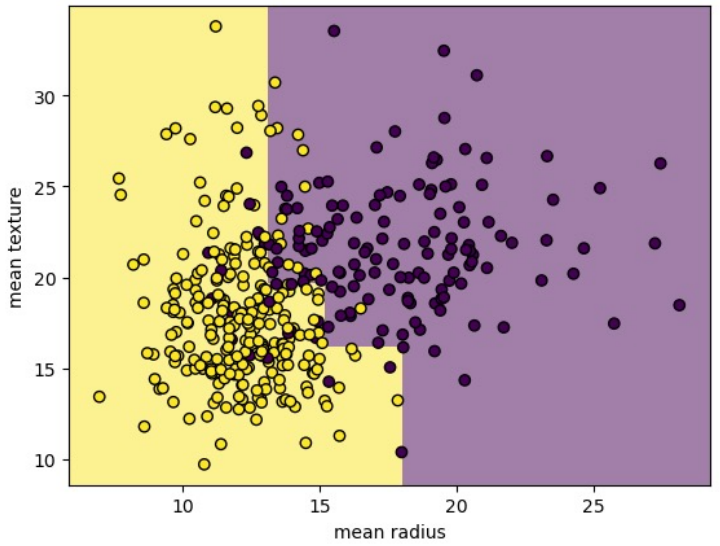
Over-fitting

Recap of Supervised Learning

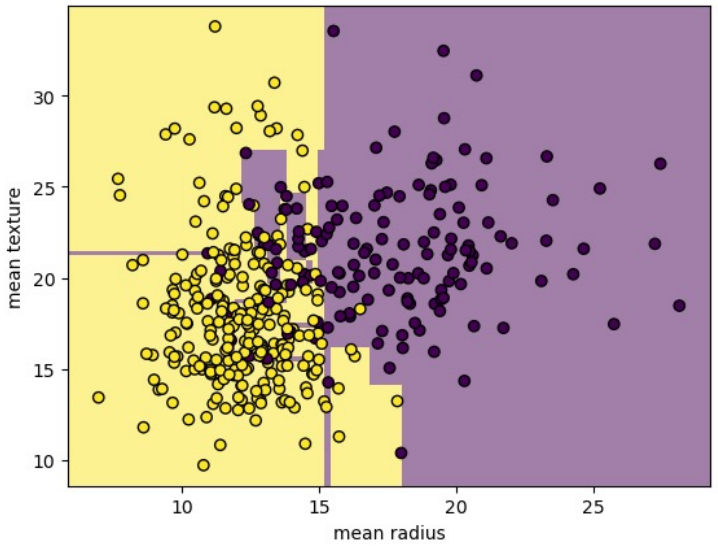
- Under-fitting & Over-fitting
  - Classification



Under-fitting



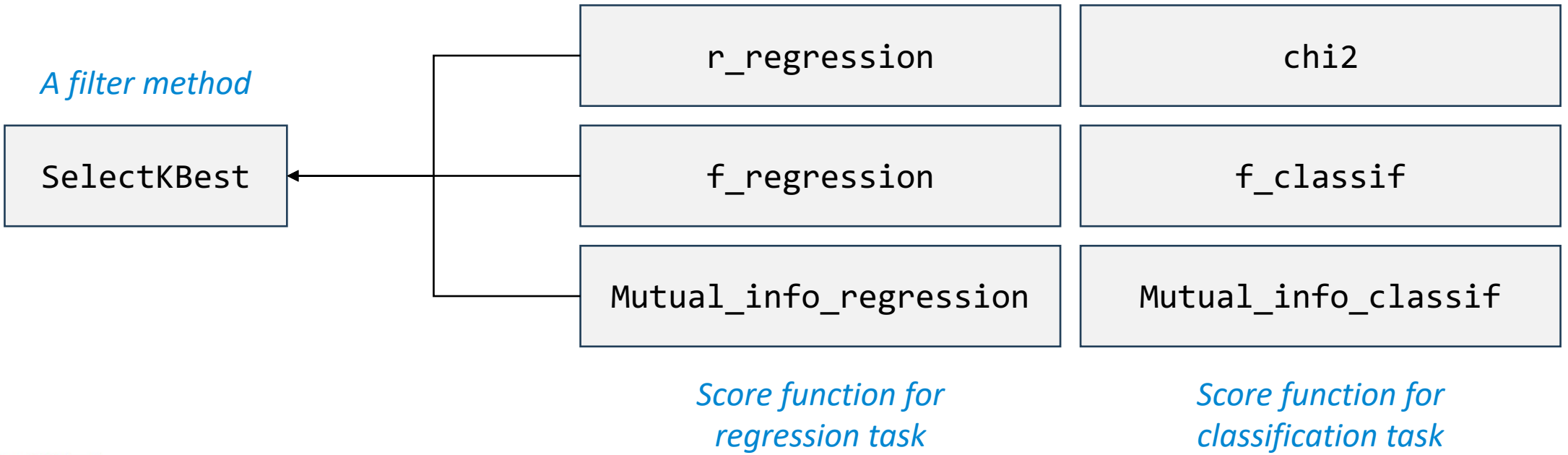
Well-fitting



Over-fitting

Recap of Supervised Learning

- Feature selection
  - Select a subset of relevant features for model construction
  - Improve model performance, reduce model complexity, reduce training time

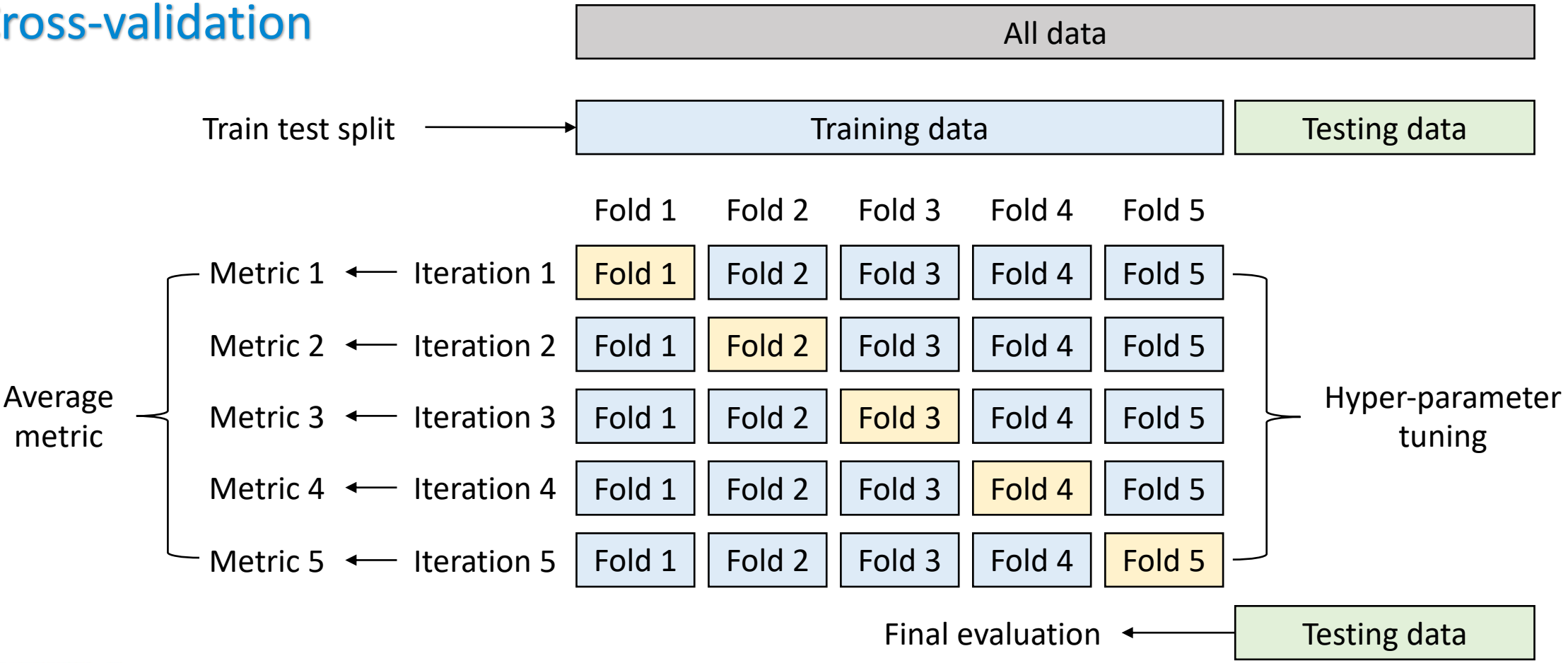


Recap of Supervised Learning

- Hyper-parameter
  - Hyper-parameter controls model's training process
  - The value of hyper-parameter is defined before training
  - The value of hyper-parameter remains unchanged during training
- Hyper-parameter tuning
  - Search the best combination of values of hyper-parameters
  - Grid search
  - Random search
- Select the best combination based on the performance on validation set

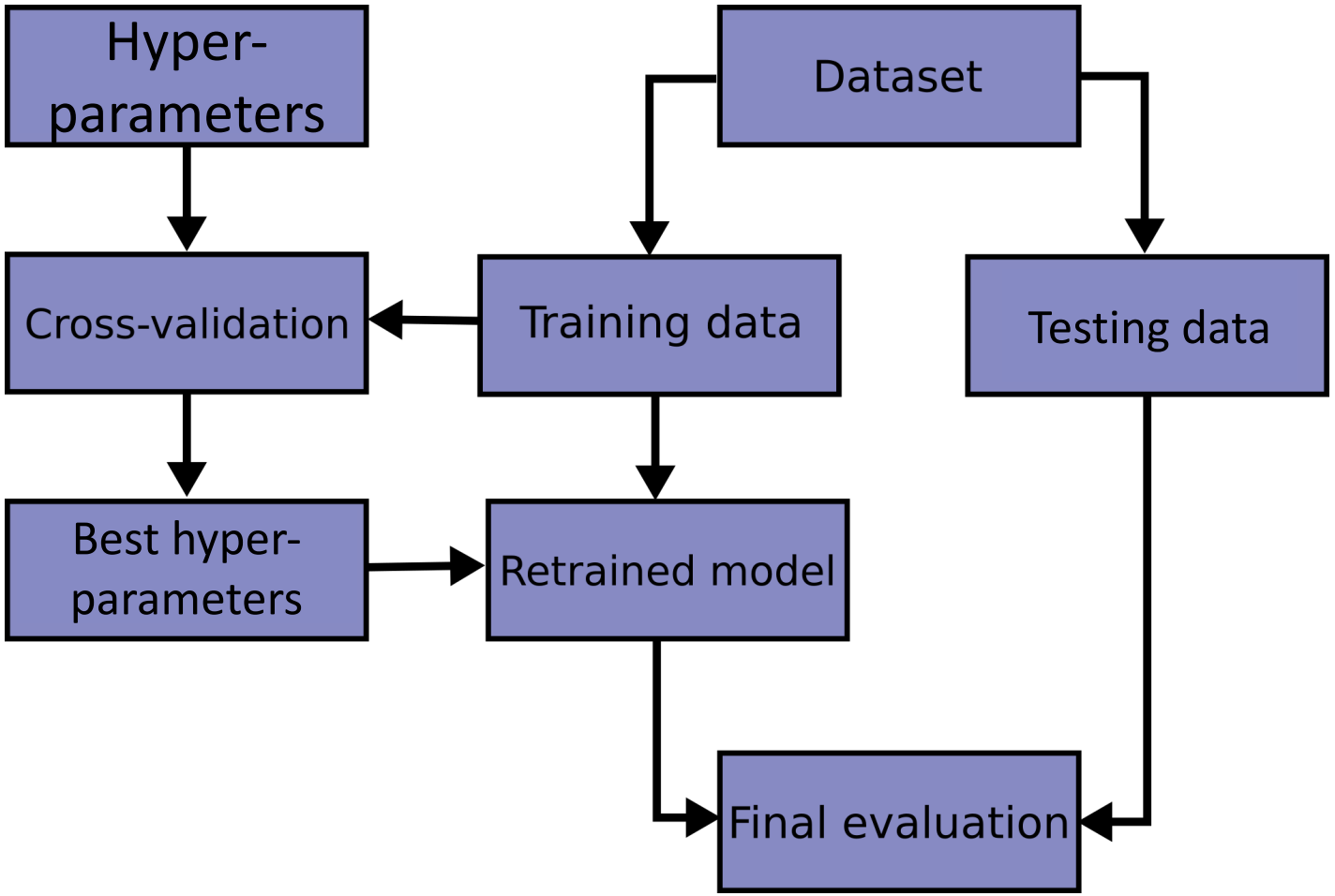
Recap of Supervised Learning

- Cross-validation



Recap of Supervised Learning

- Best practice



Outline

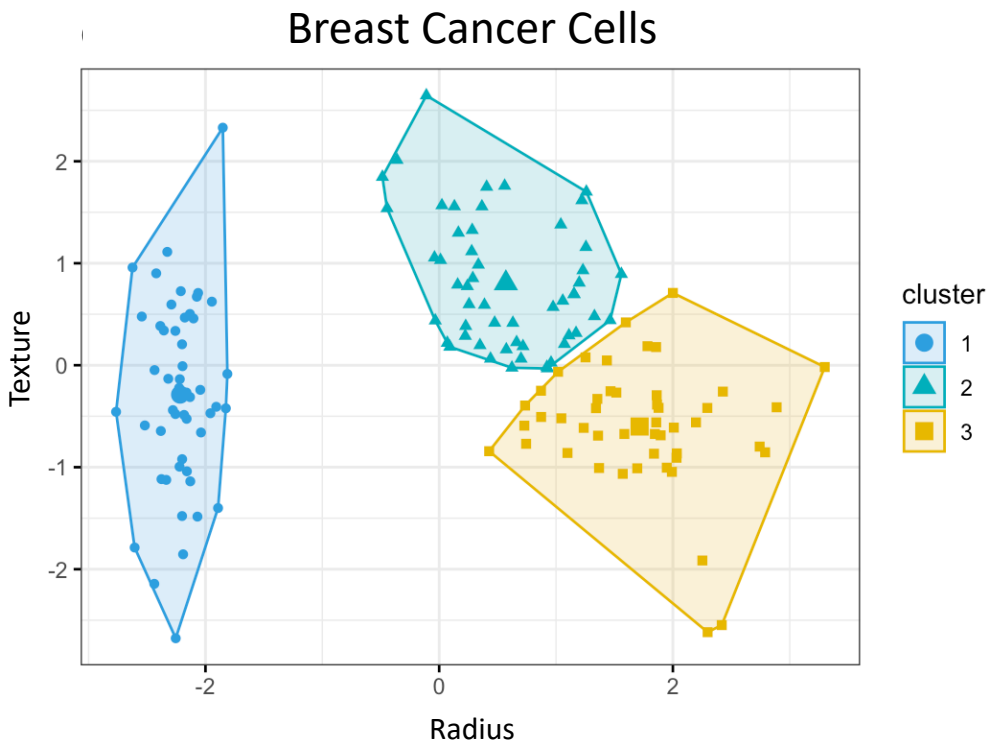
- **Unsupervised Learning**
- Clustering
- K-Means clustering

Unsupervised Learning

- Definition
  - Machine learning algorithms that learn hidden patterns or structures exclusively from unlabeled data
- Advantage
  - No need for labels / correct answers
  - Unlabeled data is the norm in the real world

Unsupervised Learning

- What does the hidden pattern or structure mean?
  - Natural groups → Clusters



Unsupervised Learning

- What does the hidden pattern or structure mean?
  - Natural groups → Clusters
  - Frequent co-occurrence → Association rules

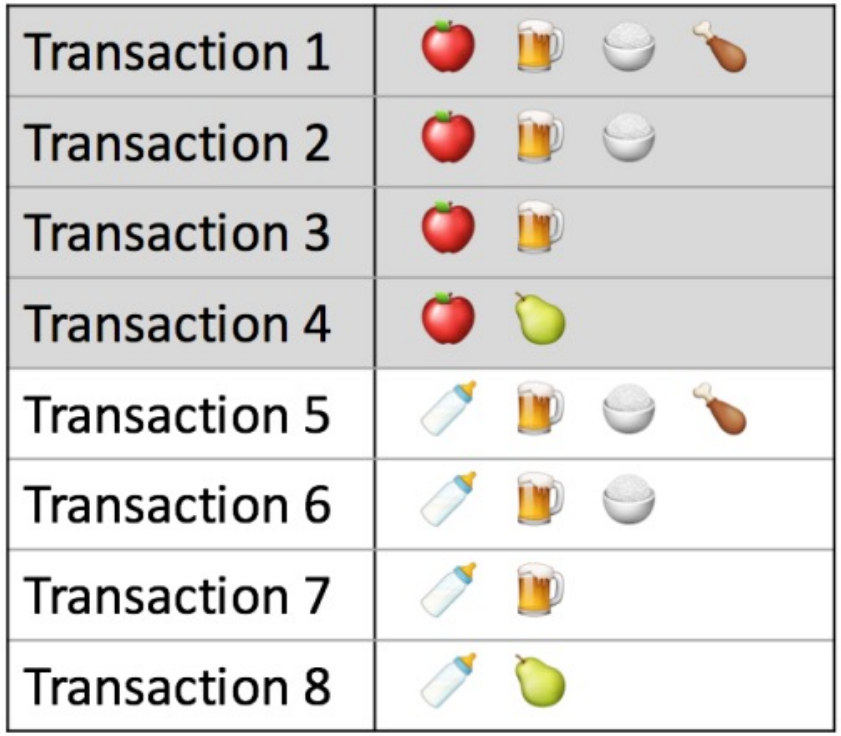


Image source: https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html

Unsupervised Learning

- What does the hidden pattern or structure mean?
  - Natural groups → Clusters
  - Frequency co-occurrence → Association rules
  - Simplifications → Compressed data

2 colors

3 colors

10 colors

256 colors
original

Unsupervised Learning

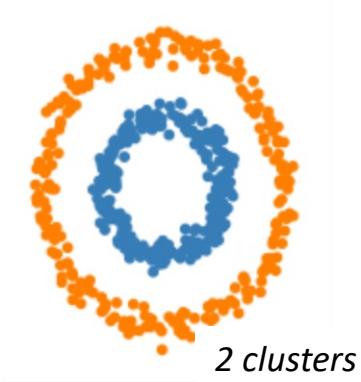
- Three types of common unsupervised learning algorithms
  - Clustering:
    - Natural groups → Clusters
  - Association rule mining:
    - Frequency co-occurrence → Association rules
  - Dimensionality reduction:
    - Simplifications → Compressed data

Outline

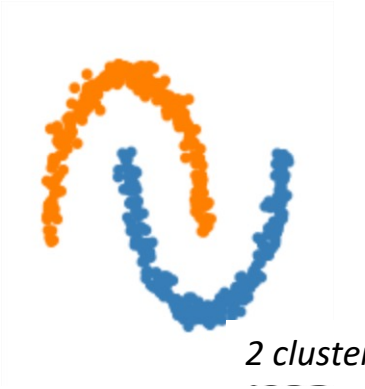
- Unsupervised Learning
- **Clustering**
- K-Means clustering

Clustering

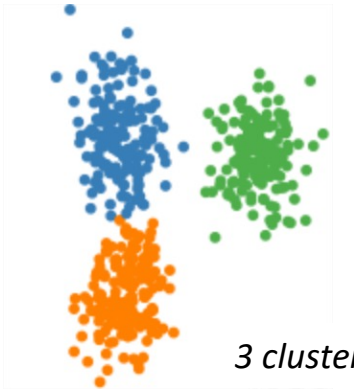
- Definition
  - Unsupervised learning algorithms that group unlabeled data into different natural clusters



2 clusters



2 clusters



3 clusters

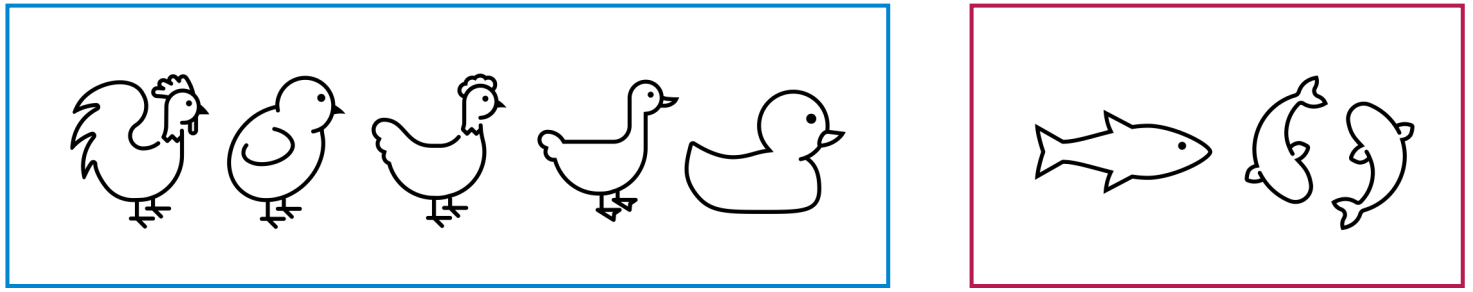
3 clusters

  - Unlike classification, there is no pre-defined correct answer for clustering

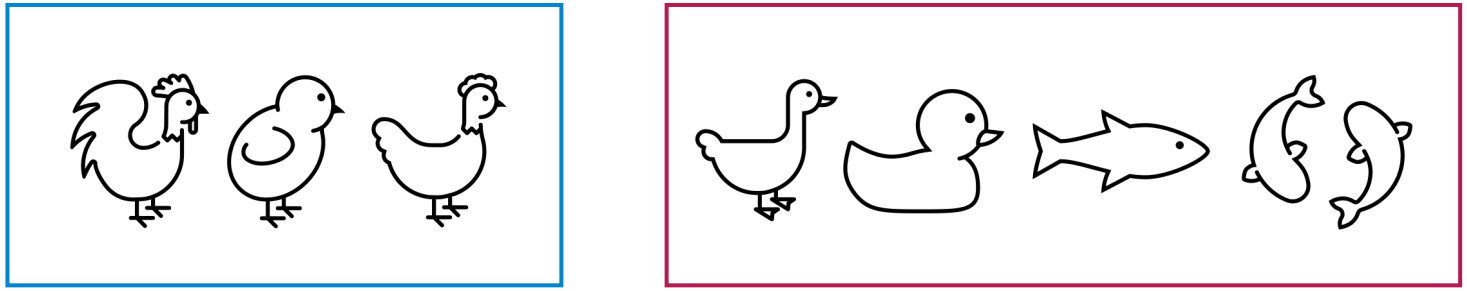
Clustering

- Key characteristics
  - The clusters are not pre-defined by humans before clustering
  - We need to interpret the clustering results after clustering

Potential 1:
Bird vs. Fish



Potential 2:
Can't swim vs Can swim



Clustering

- Typical application
  - Customer segmentation
    - Group historical customer into several clusters
    - Assign the new customer into one of the clusters

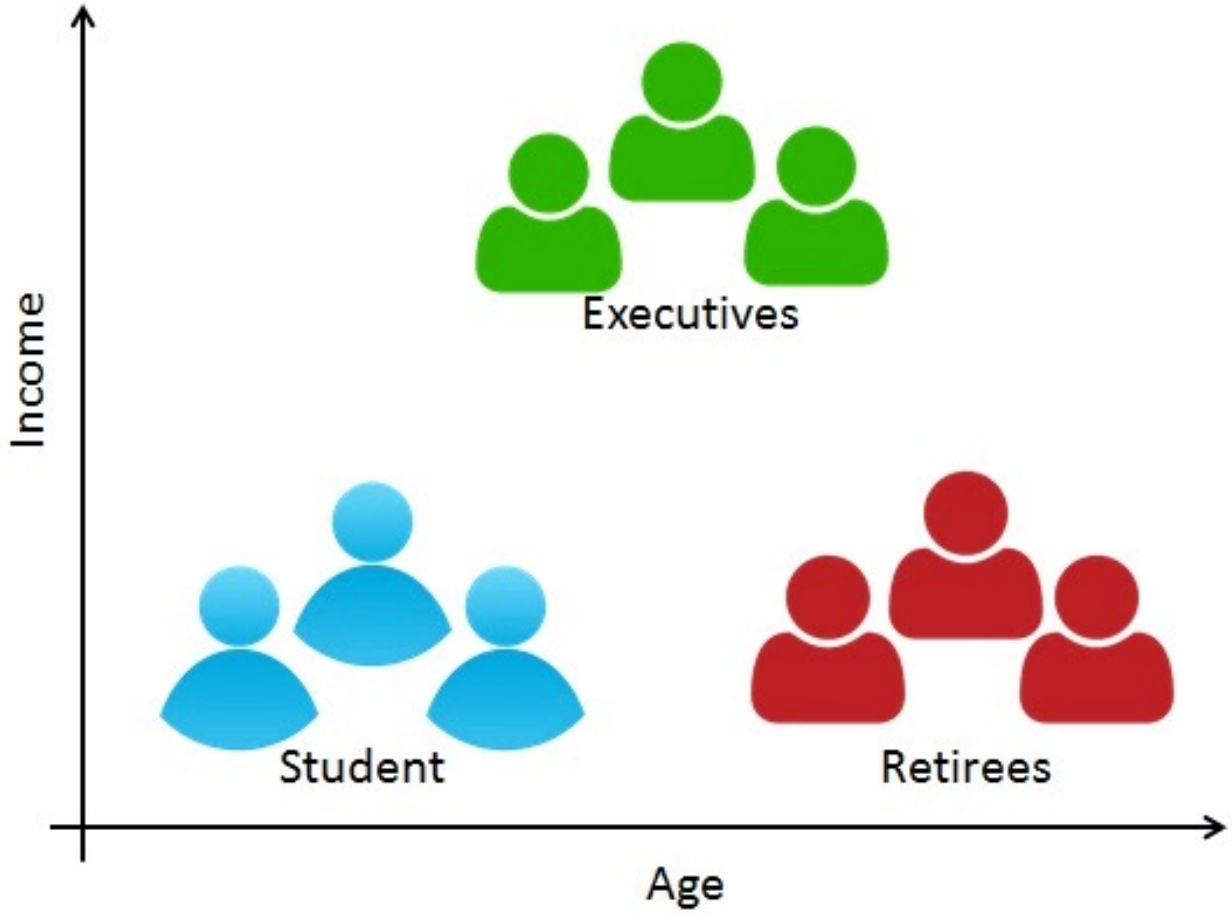


Image source: https://www.datacamp.com/tutorial/introduction-customer-segmentation-python

Clustering

- Typical application
  - Customer segmentation
  - Fraud detection
    - Group normal transaction records into different clusters
    - For a new transaction, detect whether it is anomaly by detecting whether it fall into any clusters

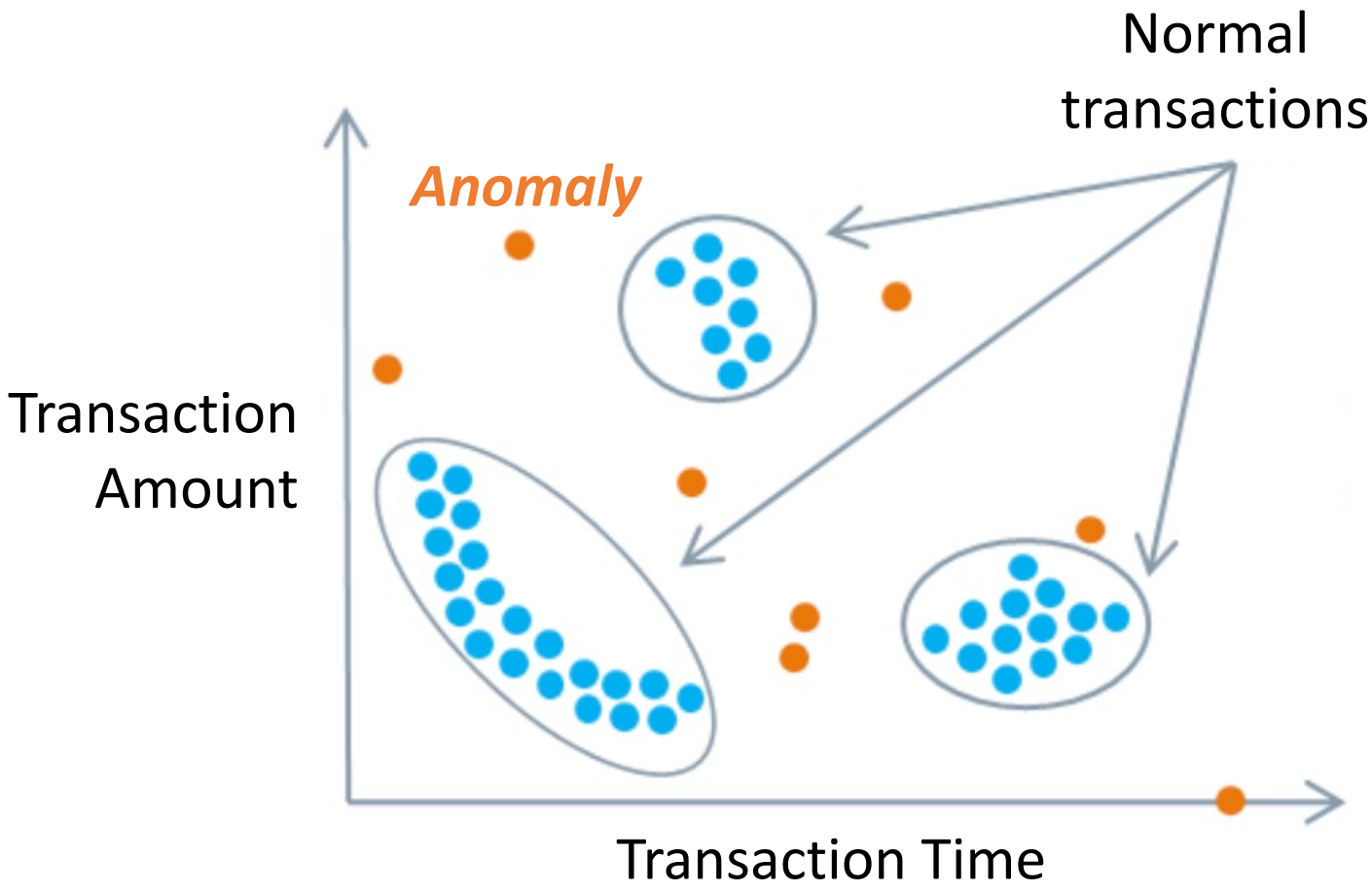


Image source: https://medium.com/adventures-with-deep-learning/anomaly-detection-615d5860755d

Clustering

- Typical application
  - Customer segmentation
  - Fraud detection
  - Social network analysis
    - Group users into different communities based on the interaction between them
      - Follow
      - Like
      - Forward
      - Comment
      - …

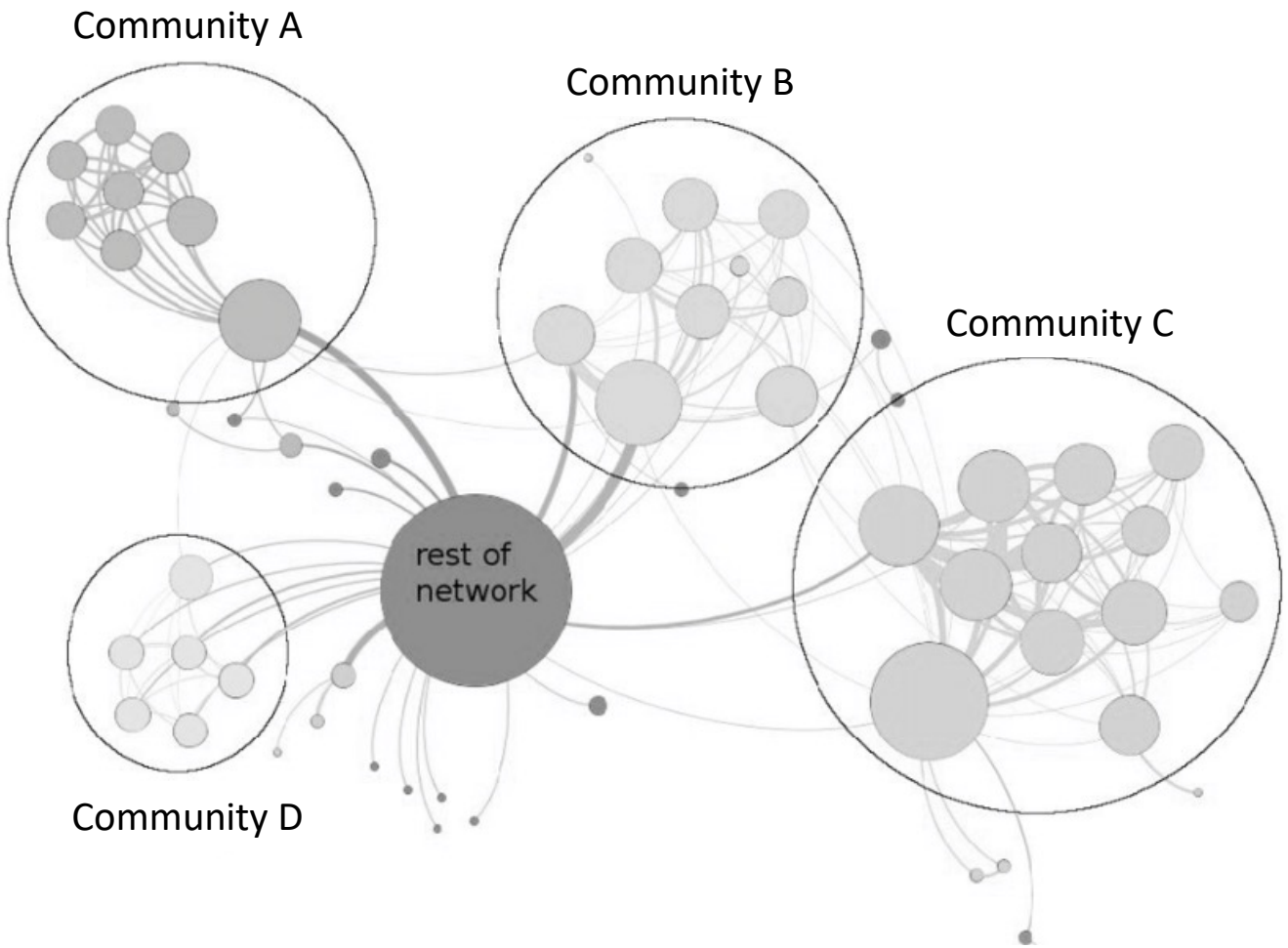
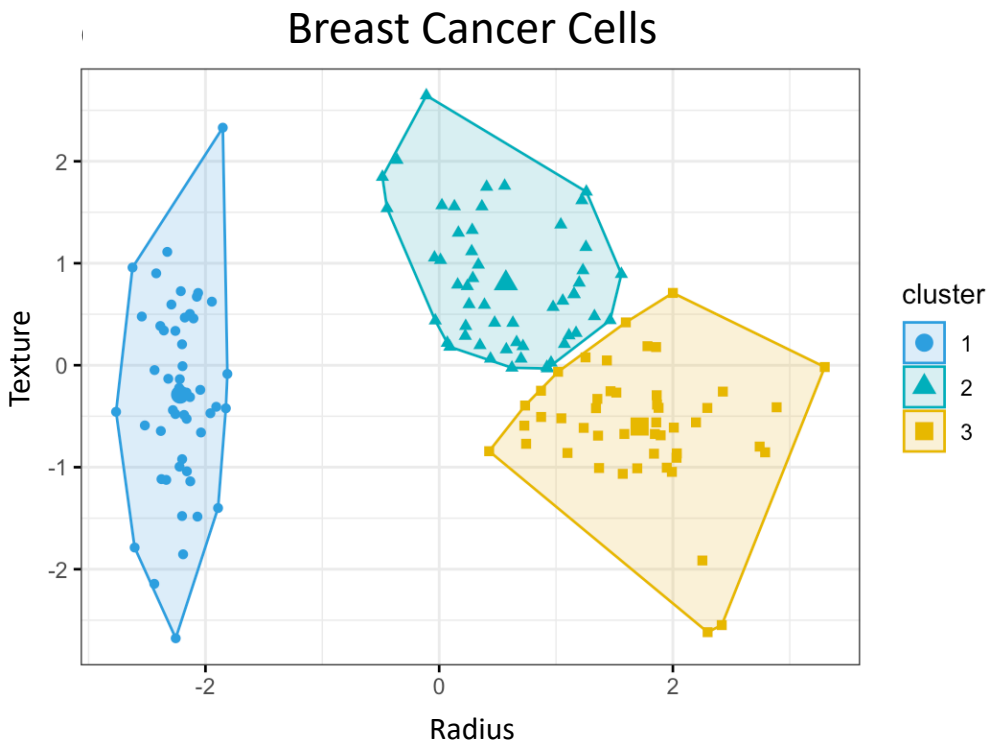


Image source: https://www.researchgate.net/publication/332207478_The_Distance_Between_Us

Clustering

- What is a cluster?
  - A natural group of data points
  - Data points within the same cluster are similar.
    - Close to each other in the feature space.
  - Data points in different clusters are different.
    - Far away from each other in the feature space.

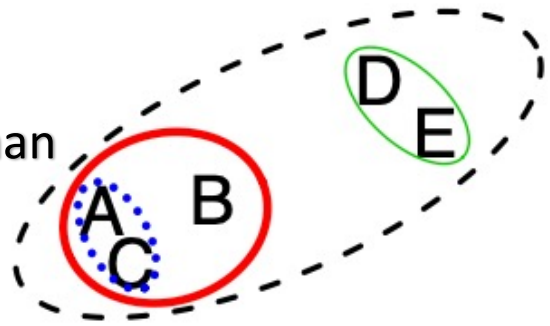


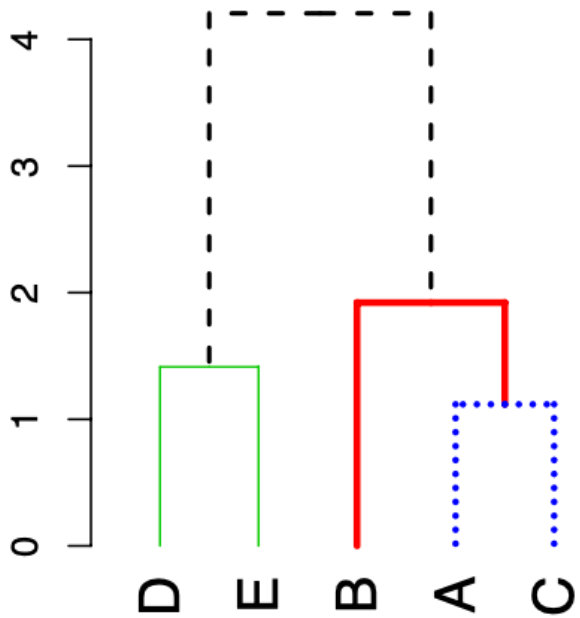
Clustering

- Different types of clustering algorithms
  - Connectivity-based clustering
  - Centroid-based clustering
  - Distribution-based clustering
  - Density-based clustering

Clustering

- Different types of clustering algorithms
  - Connectivity-based clustering (hierarchical clustering)
    - Cluster is defined as sets of points that are closely connected to each other
    - Points are more related to nearby points than to points far away
    - Connect close points to form clusters hierarchically
    - Result in a tree structure





Clustering

- Different types of clustering algorithms
  - Centroid-based clustering (partitioning method)
    - Cluster is defined by the proximity of data points to the centroid.
    - The centroid is the average of all points in a cluster.
    - Each cluster is represented by a centroid
    - Partition the feature space into different parts based on the distance to the centroid of a cluster

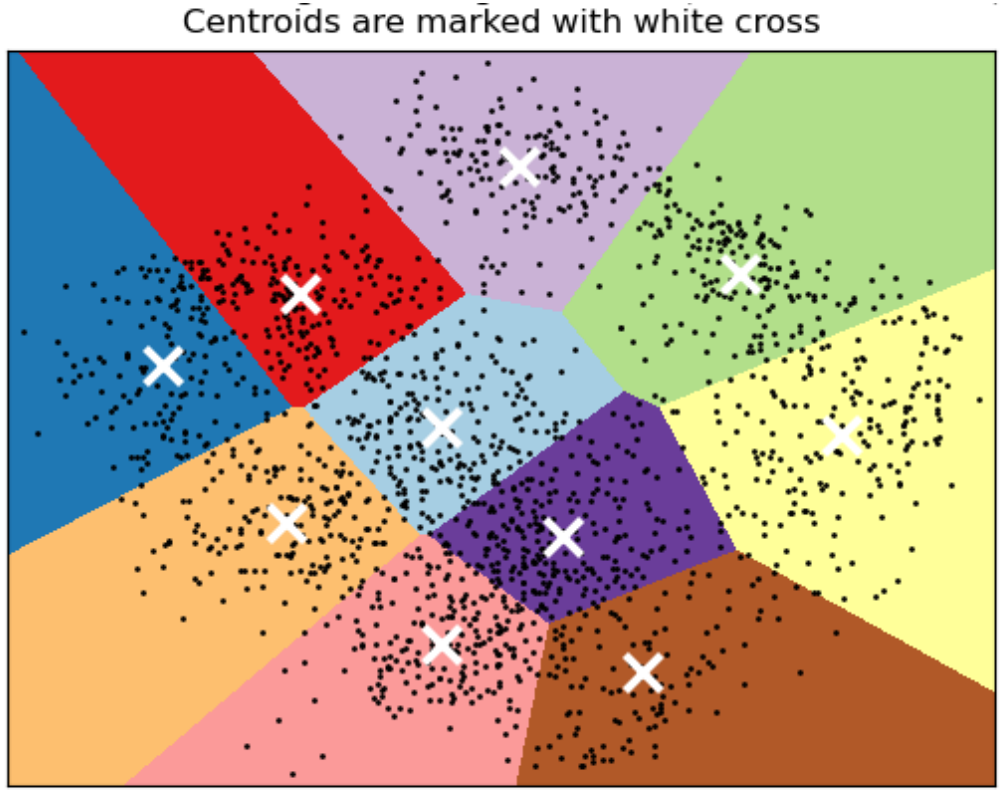
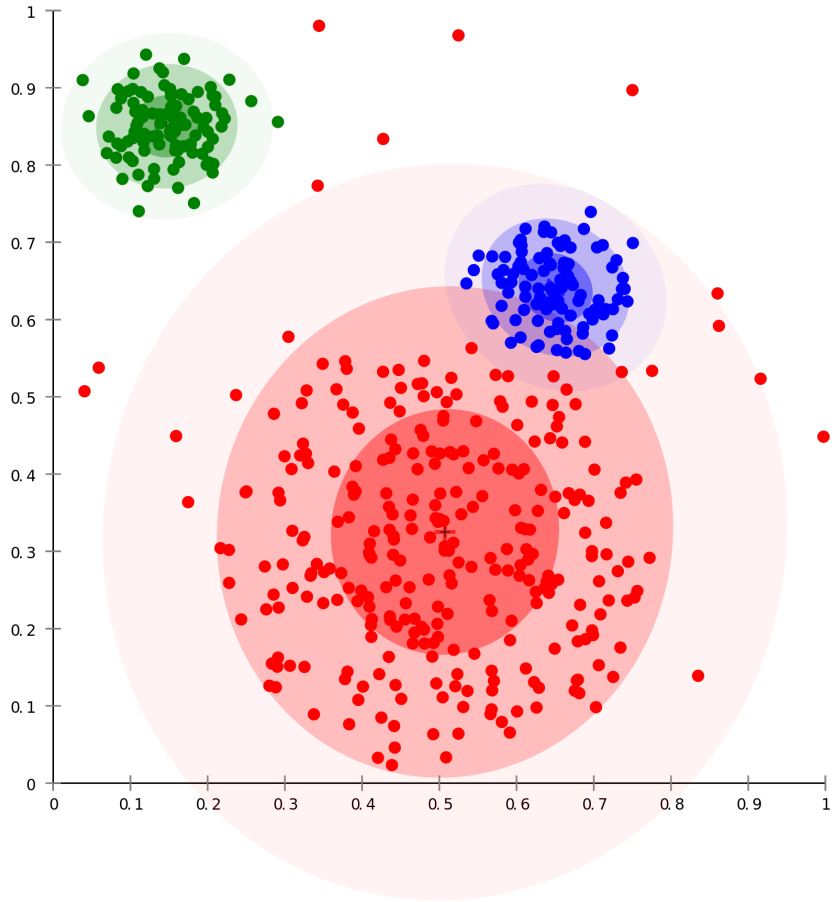


Image source: https://towardsdatascience.com/6-types-of-clustering-methods-an-overview-7522dba026ca

Clustering

- Different types of clustering algorithms
  - Distribution-based clustering
    - A cluster can be defined by a statistical distribution (e.g., Gaussian distributions).
    - A distribution describes how points in a cluster are spread out.
    - Clusters are formed based on the likelihood of belonging to the same distribution.
    - Points are more likely to be in a cluster if they belong to the same distribution.



Clustering

- Different types of clustering algorithms
  - Density-based clustering
    - Clusters are defined as areas of the data space where data points are densely packed together.
    - Points in sparse regions are usually considered noise or border points.

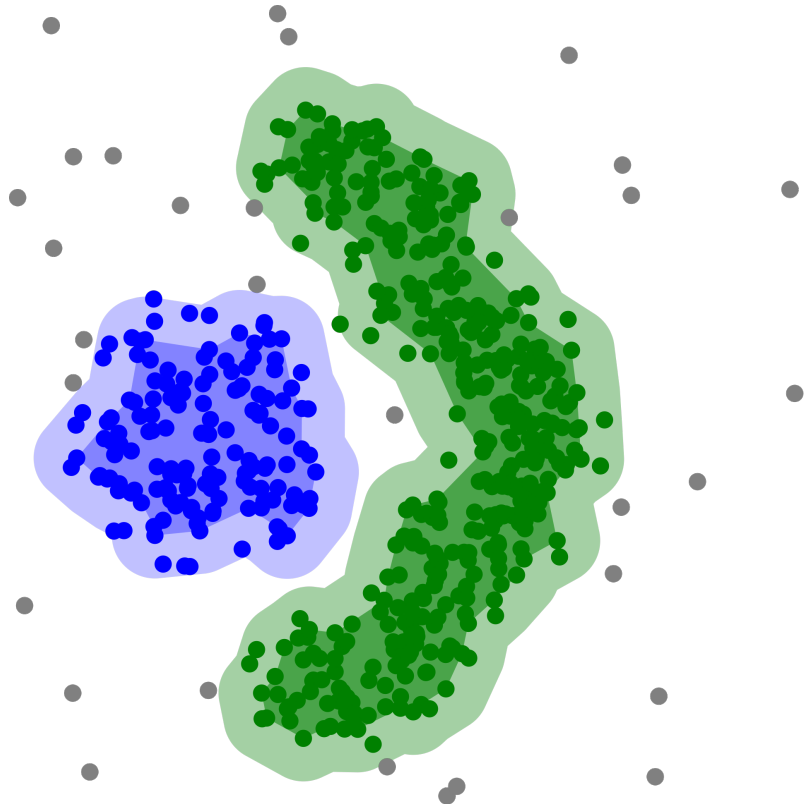


Image source: https://en.wikipedia.org/wiki/Cluster_analysis

Outline

- Unsupervised Learning
- Clustering
- **K-Means clustering**

K-Means clustering

- An iterative centroid-based clustering algorithm
- A partitioning method that divides the feature space into K distinct, non-overlapping subsets, so-called clusters
- Each cluster is represented by a centroid
  - Which cluster a point belongs to depends on which centroid it is closest to

K-Means clustering

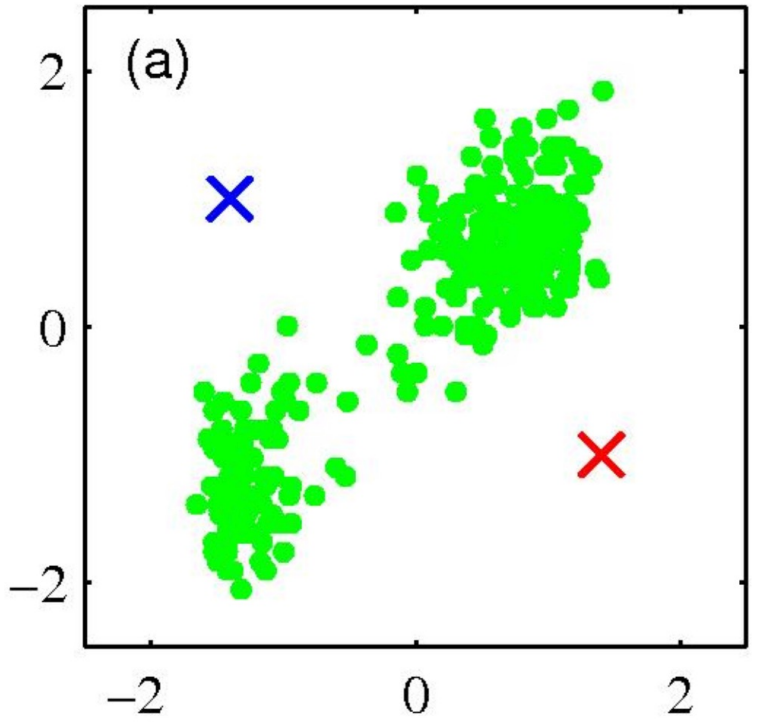
- Steps
  1. Initialize K centroids randomly
  2. Assign each data point to the nearest centroid, forming K clusters
  3. Recalculate the centroids as the mean of all points in each cluster
  4. Repeat steps 2 and 3 until convergence
     - i.e., centroids do not change significantly

K-Means clustering

- Steps
  1. Initialize K centroids randomly
  2. Assign each data point to the nearest centroid, forming K clusters
  3. Recalculate the centroids as the mean of all points in each cluster
  4. Repeat steps 2 and 3 until convergence
     - i.e., centroids do not change significantly
     - Or we can set a hard limit on the number of iterations

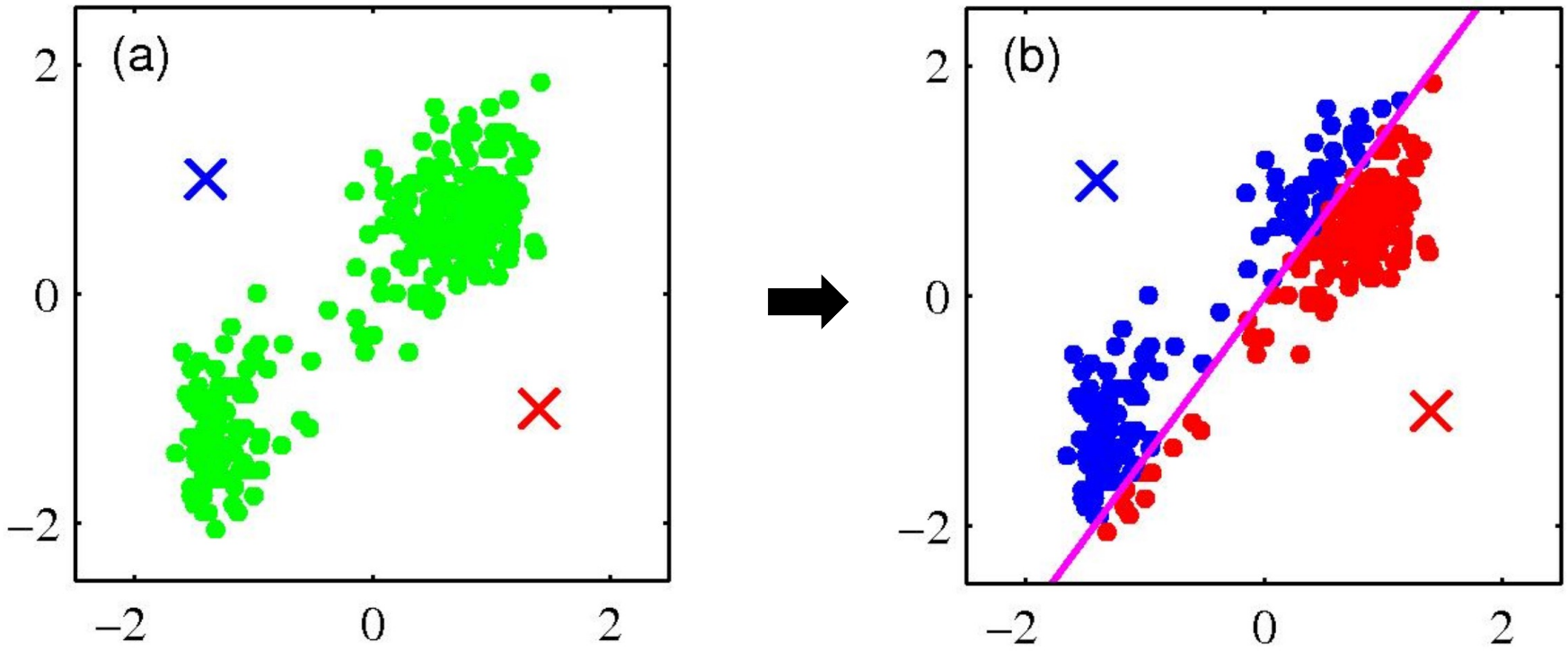
K-Means clustering

- Example
  1. Initialize K centroids randomly
     - The value of K is determined by us
     - Assume K = 2 for this example
     - Randomly initialize K centroids
       - Randomly choose K examples in the dataset
       - Or Randomly select K points in the feature space
         - No need to be existing examples in the dataset



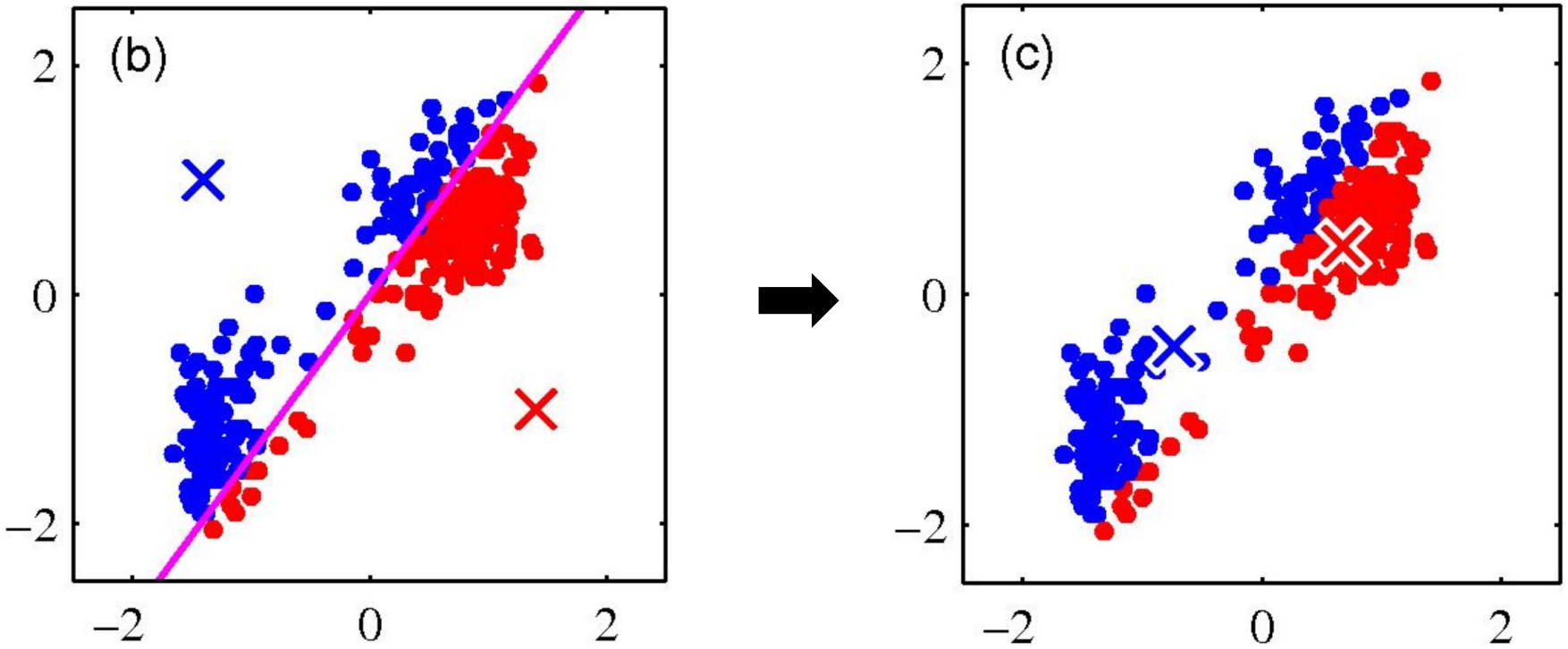
K-Means clustering

- Example
  2. Assign each data point to the nearest centroid, forming K clusters



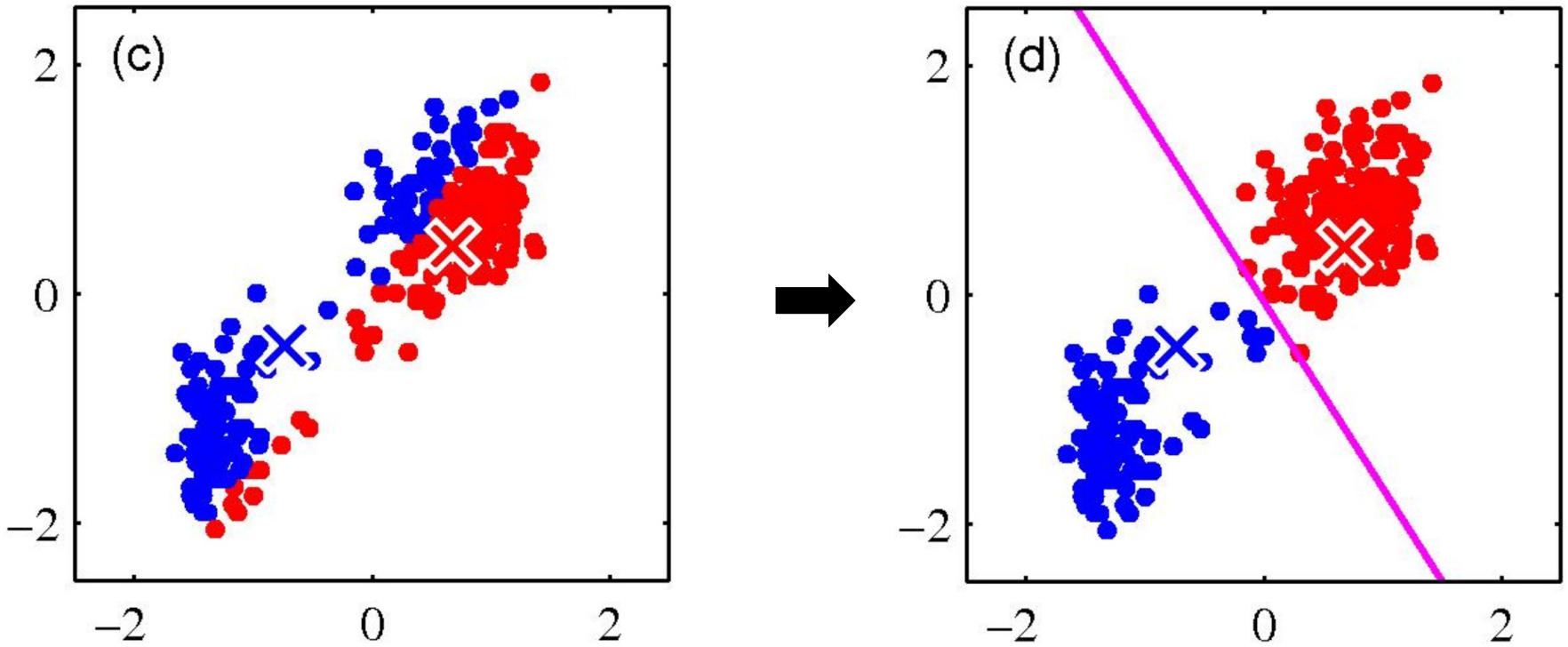
K-Means clustering

- Example
  3. Recalculate the centroids as the mean of all points in each cluster



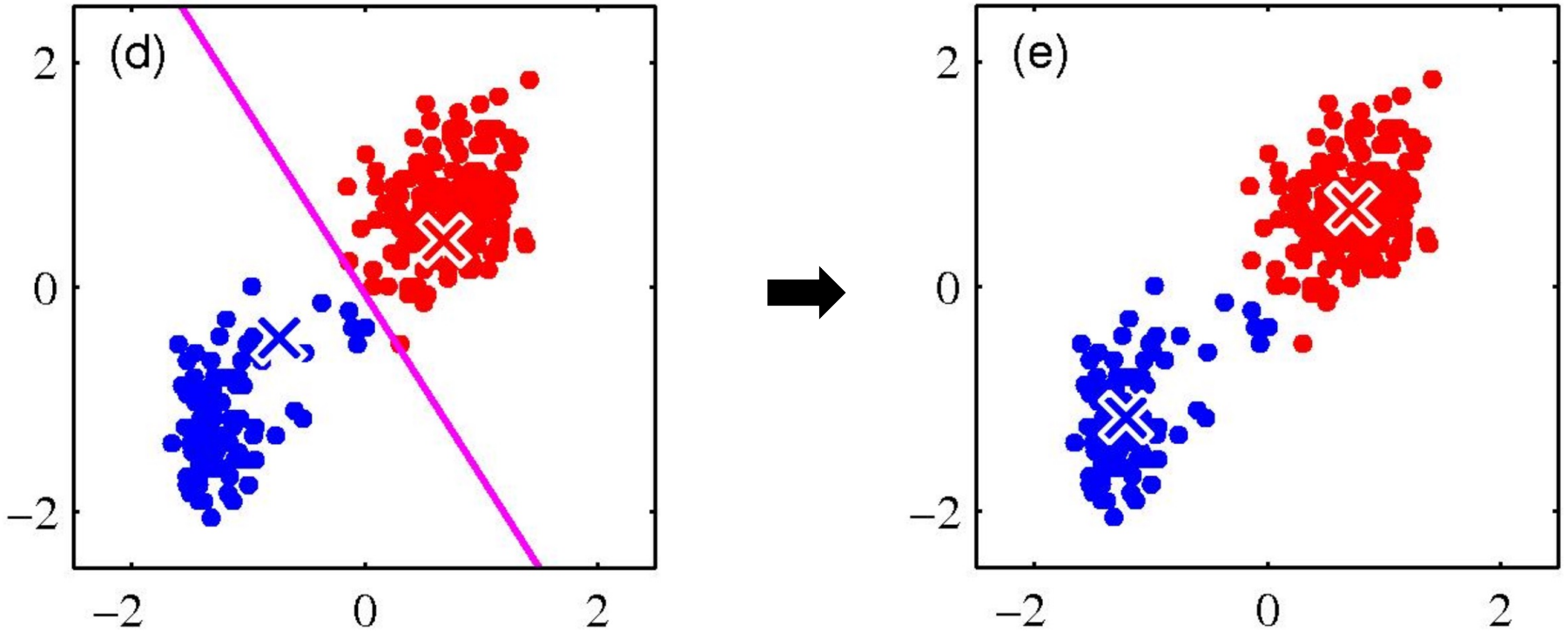
K-Means clustering

- Example
  - Repeat step 2, assign each data point to the nearest centroid, forming K clusters



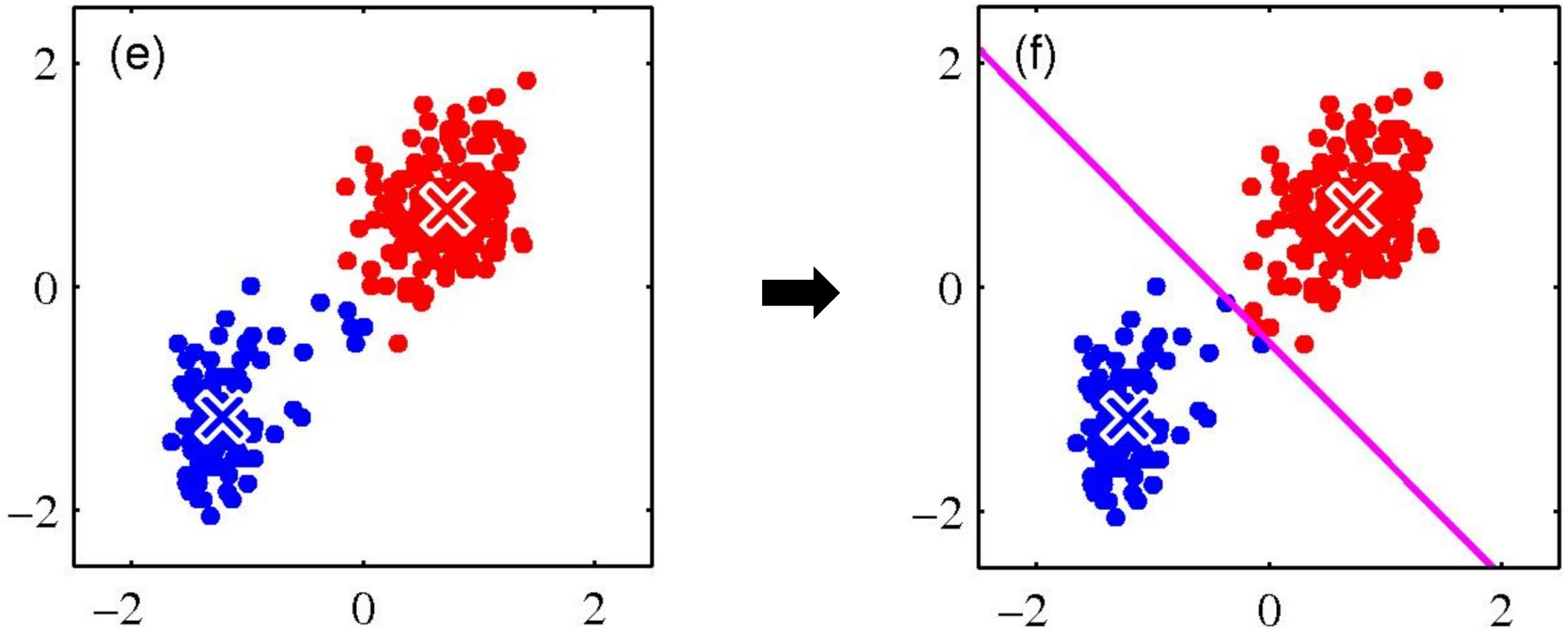
K-Means clustering

- Example
  - Repeat step 3, recalculate the centroids as the mean of all points in each cluster



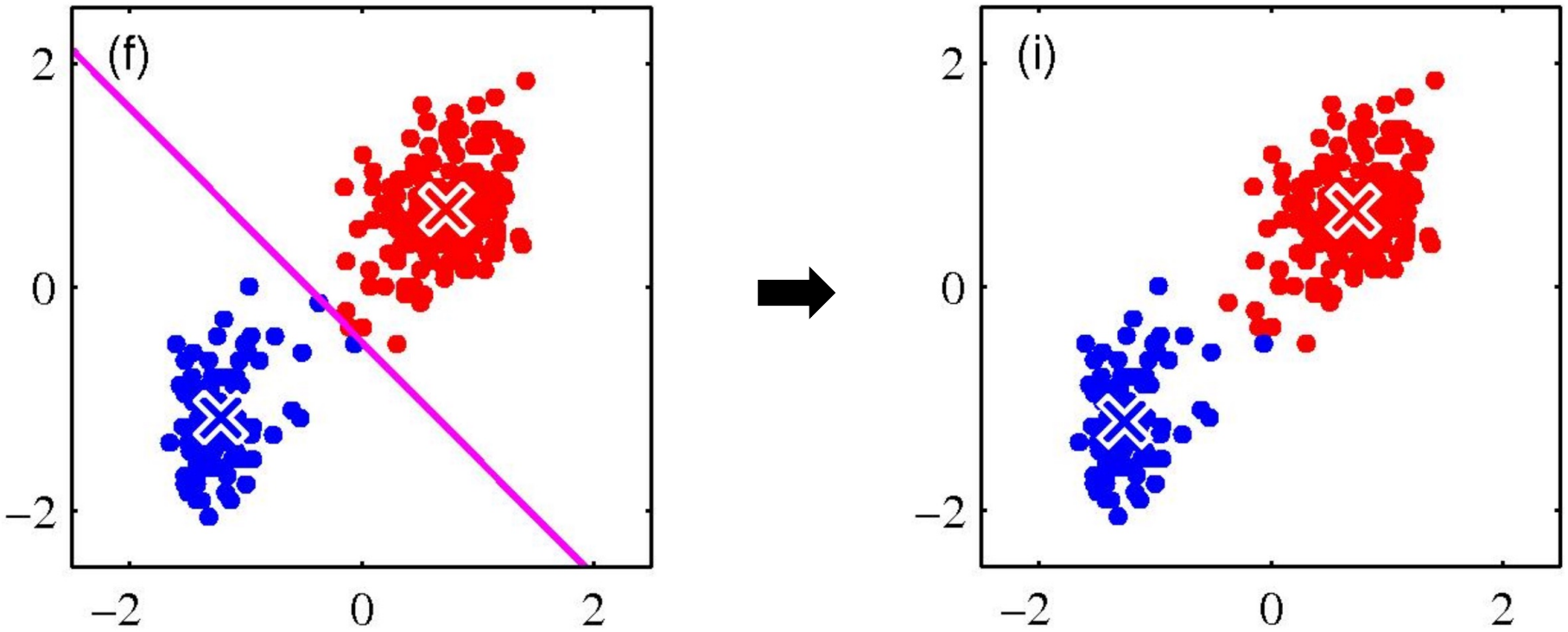
K-Means clustering

- Example
  - Repeat step 2, assign each data point to the nearest centroid, forming K clusters



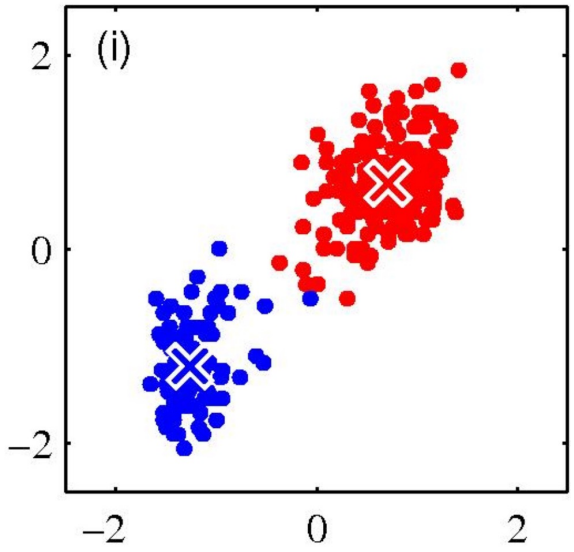
K-Means clustering

- Example
  - Repeat step 3, recalculate the centroids, no significant changes, convergence



K-Means clustering

- Clustering results
  - A set of centroids that minimize the inertia
  - Or WCSS (Within-Cluster Sum of Squares)
  - n samples $(x_1, \dots, x_n)$
  - divided into k clusters $(C_1 \dots, C_k)$
  - represented by k centroids $(\mu_1 \dots, \mu_k)$



$$inertia = \sum_{i=0}^{n} \min_{\mu_j \in C} (||x_i - \mu_j||^2)$$

K-Means clustering

- Assumptions of K-Means clustering
  - The clusters need to be convex and isotropic.
  - Perform poorly to elongated clusters, or manifolds with irregular shapes.

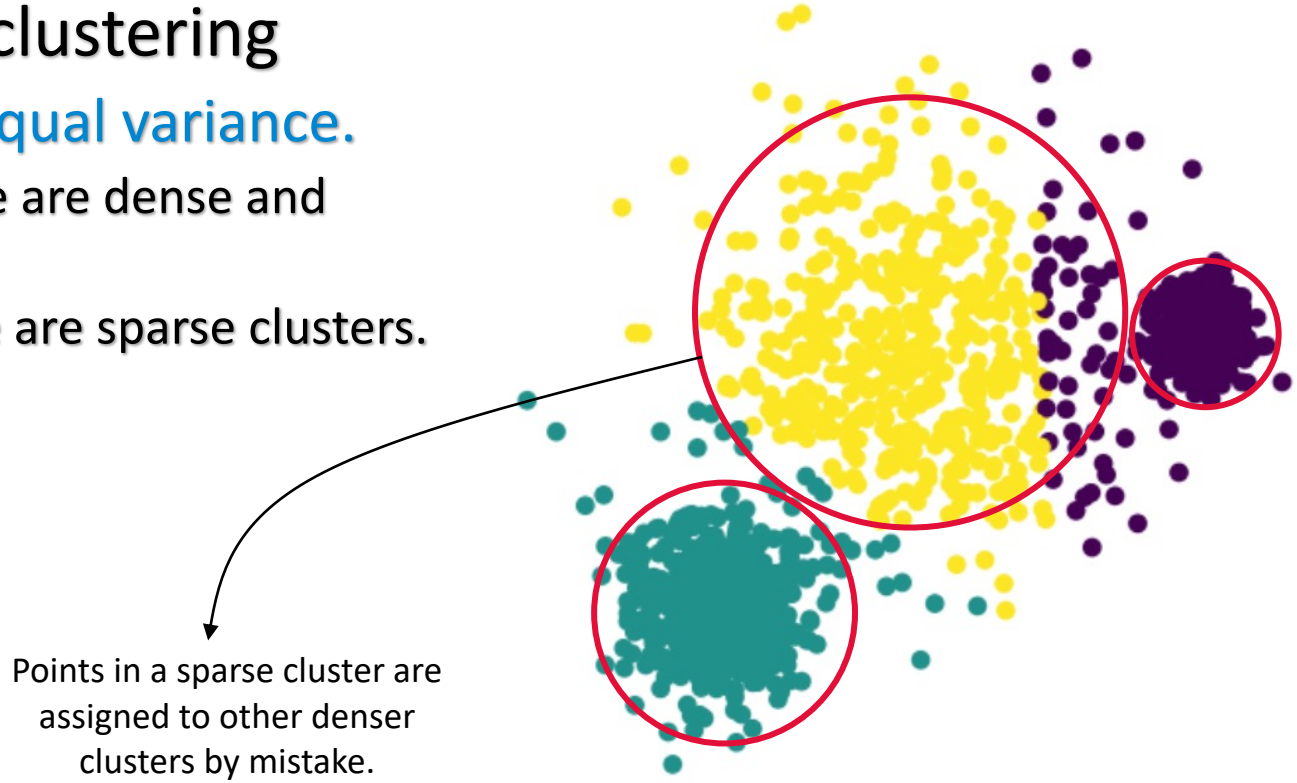
convex & isotropic
good clustering

elongated clusters
poor clustering

manifolds with irregular
shapes, poor clustering

K-Means clustering

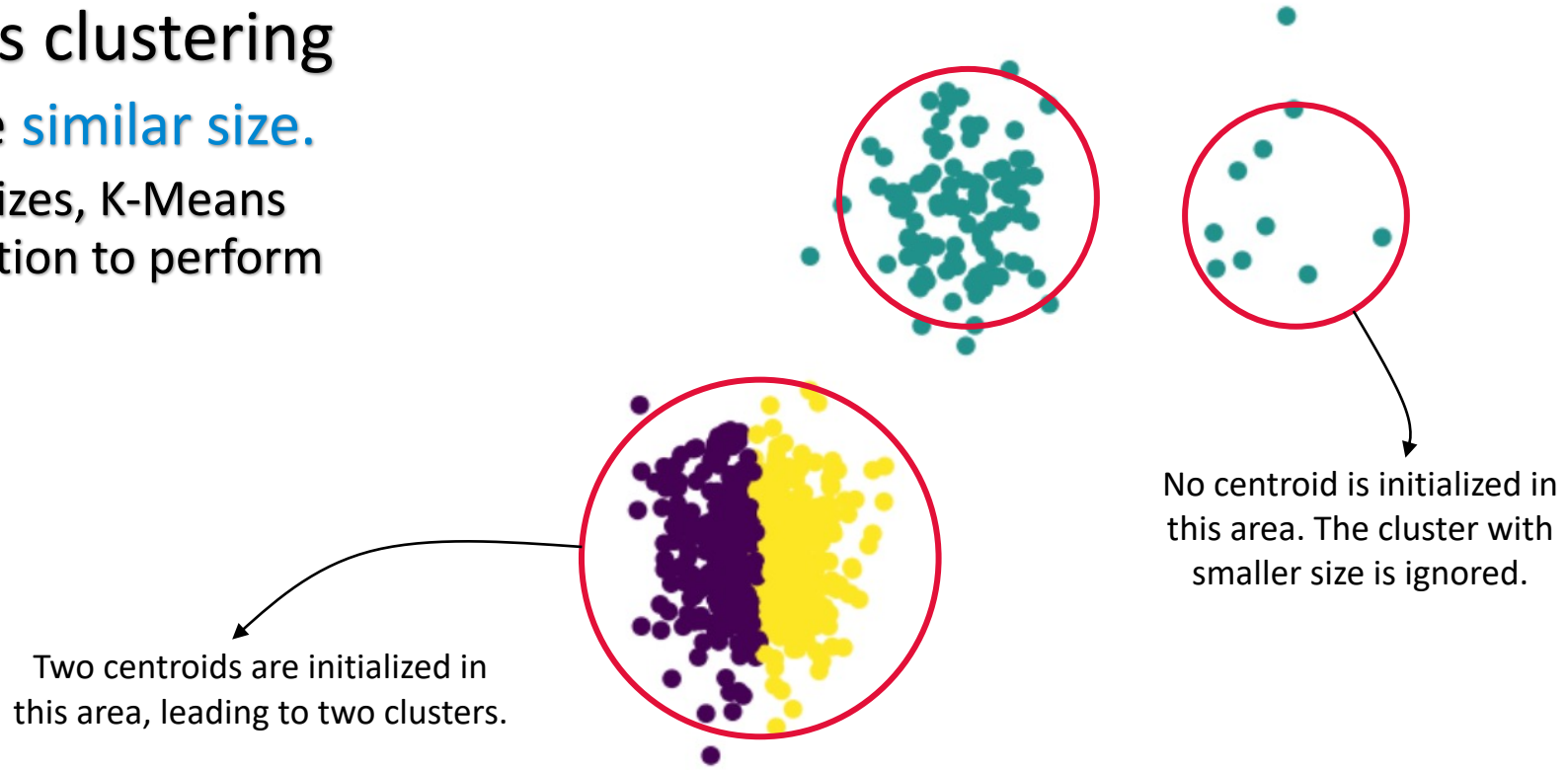
- Assumptions of K-Means clustering
  - The clusters should have equal variance.
    - Clusters with small variance are dense and compact clusters.
    - Clusters with large variance are sparse clusters.



Points in a sparse cluster are assigned to other denser clusters by mistake.

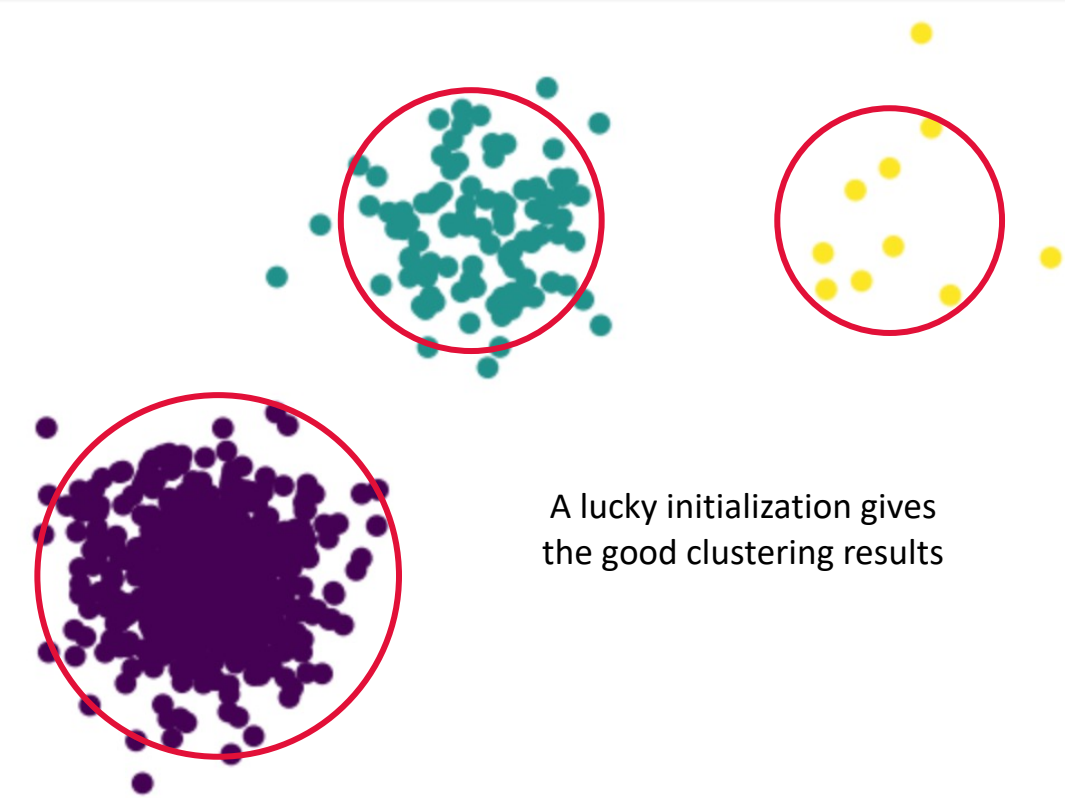
K-Means clustering

- Assumptions of K-Means clustering
  - The clusters should have similar size.
    - If clusters have uneven sizes, K-Means requires a good initialization to perform good.



K-Means clustering

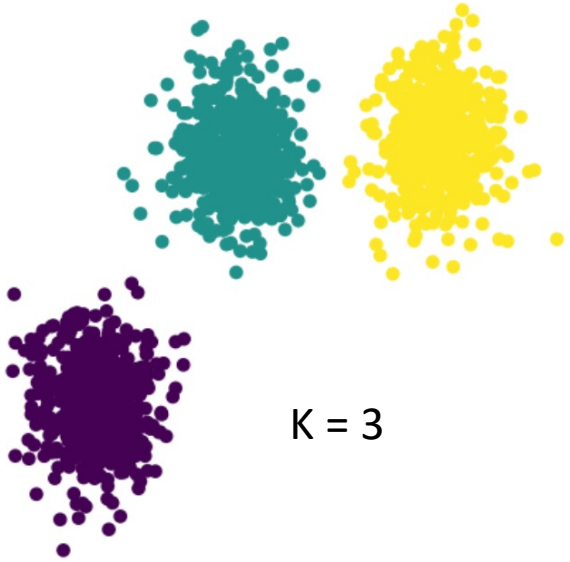
- Assumptions of K-Means clustering
  - The clusters should have similar size.
    - If clusters have uneven sizes, K-Means requires a good initialization to perform good.



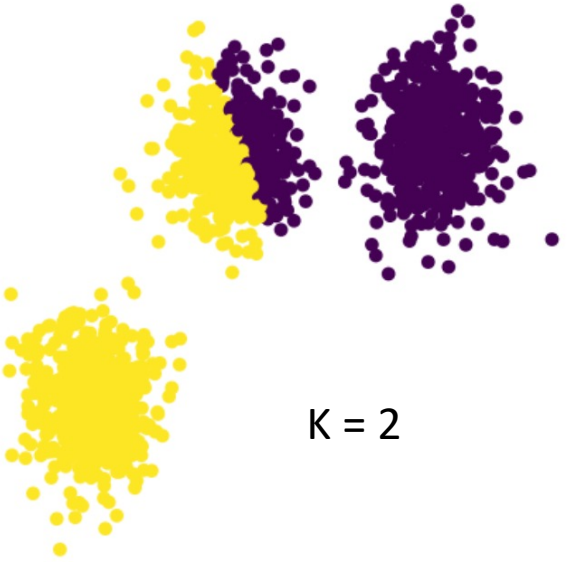
A lucky initialization gives the good clustering results

K-Means clustering

- Assumptions of K-Means clustering
  - We need to specify a good value of K.



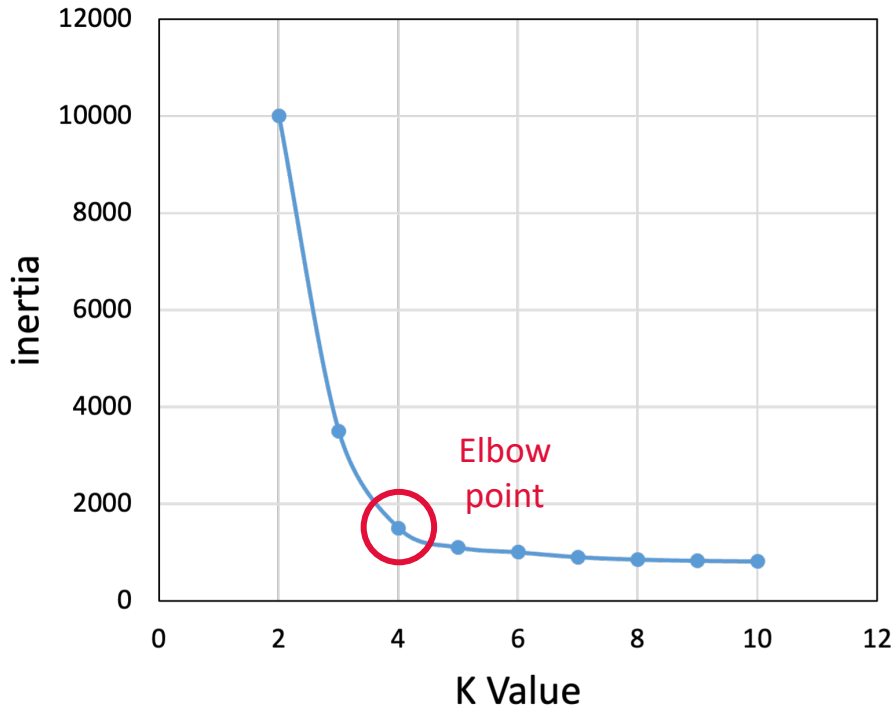
K = 3



K = 2

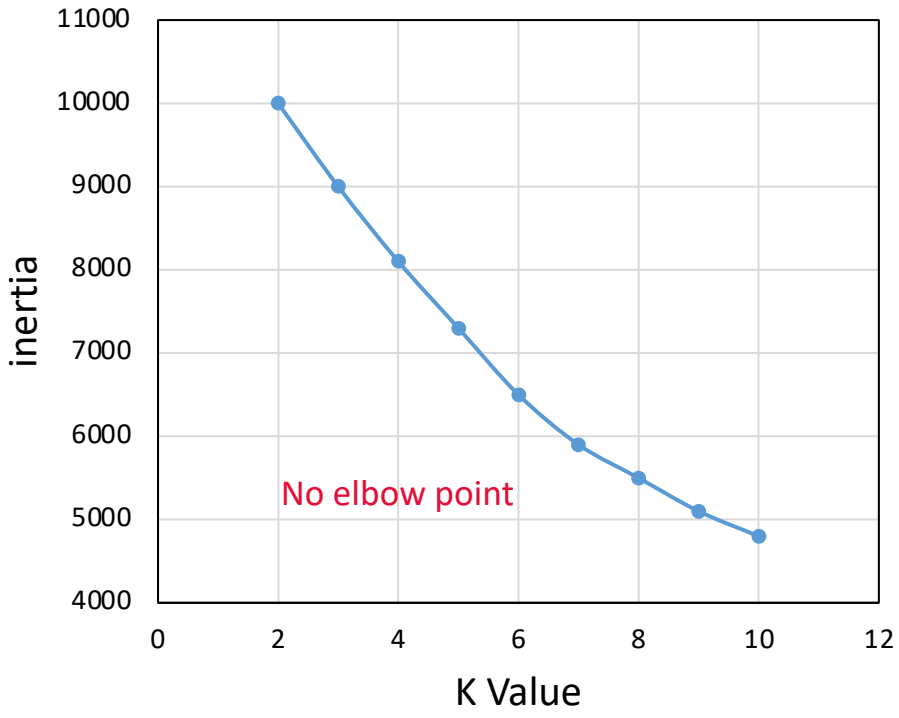
K-Means clustering

- The elbow method for selecting optimal K
  - Increase K will decrease inertia
    - In extreme case, $K = n$, each cluster only have one point, then inertia will be zero
    - We expect that the K vs. inertia curve has an elbow shape
      - As K increases, the inertia decreases first rapidly and then slowly.
      - Elbow point is a subjective choice.
      - Beyond the elbow point, increasing K does not lead to a significant reduction in inertia.



K-Means clustering

- The elbow method for selecting optimal K
  - Elbow point is a subjective choice.
  - Elbow point doesn't guarantee the optimal clustering quality.
    - Inertia only considers how compact each cluster is.
    - Inertia doesn't consider how dissimilar different clusters are.
  - Sometimes there is no clear elbow point.
- We need to combine other clustering performance evaluation metrics.



K-Means clustering

- Clustering performance evaluation metrics
  - Metrics used to evaluate the clustering quality
  - What is a good clustering?
    - Points within the same cluster are similar to each other
      - The clusters are dense and compact.
      - Low within-cluster variance
      - High intra-cluster similarity
    - Points in different clusters are different from each other
      - Different clusters are far away from each other.
      - High inter-cluster differences

Inertia:

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (||x_i - \mu_j||^2)$$

K-Means clustering

- Clustering performance evaluation metrics
  - Metrics considering both aspects
    - Silhouette Coefficient

      - **a**: The mean distance between a sample and all other points in the same class.
      - **b**: The mean distance between a sample and all other points in the *next nearest cluster*.

      The Silhouette Coefficient *s* for a single sample is then given as:

      $$s = \frac{b - a}{max(a, b)}$$

      - The score ranges from -1 to 1.
      - A high value indicates that the point is well matched to its own cluster and poorly matched to neighboring clusters.

K-Means clustering

- Clustering performance evaluation metrics
  - Metrics considering both aspects
    - Calinski-Harabasz Index

$$CH(k) = \frac{B(k)/(k-1)}{W(k)/(n-k)}$$

      - Where $B(k)$ is the between group dispersion matrix
      - $W(k)$ is the within-cluster dispersion matrix
      - n is the number of points, and k is the number of clusters.

      - The score is higher when clusters are dense and well separated.

K-Means clustering

- Clustering performance evaluation metrics
  - Metrics considering both aspects
    - Davies-Bouldin Index

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right)$$

      - Where σ is the average distance of all points in the cluster to the cluster centroid
      - c_i and c_j are the centroids of clusters i and j
      - d is the distance between centroids

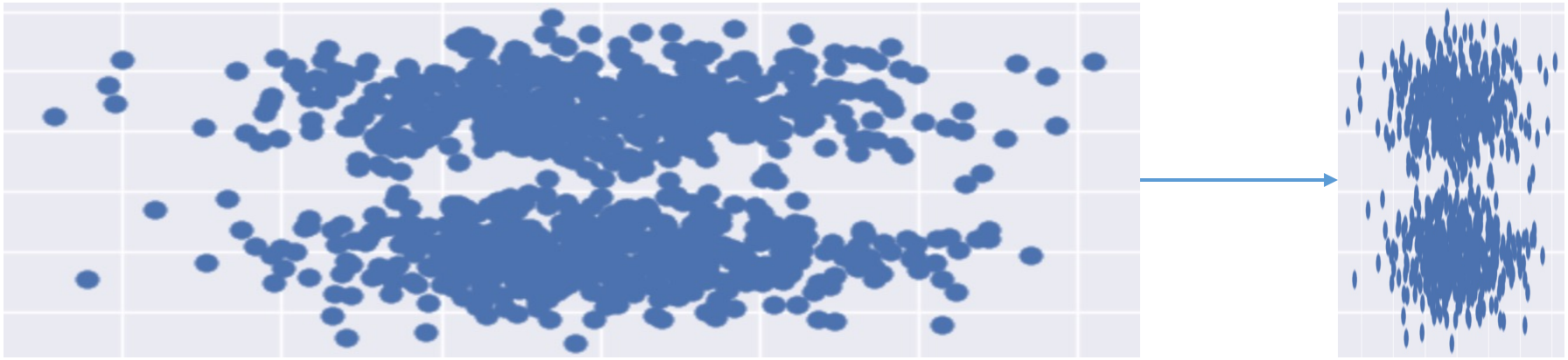
      - Lower values indicate better clustering as they imply lower intra-cluster distances and higher inter-cluster distances.

K-Means clustering

- Matters need attention:
  - K-Means groups data points based on distance
  - We need to perform feature scaling before using K-Means algorithm
  - Be careful:
    - `StandardScaler` might change the original shape of the natural clusters
    - `MaxMinScaler` will keep the original shape of the natural clusters
    - It's better to use `MaxMinScaler` if the origial clusters are convex and isotropic

K-Means clustering

- Matters need attention:
  - If the original clusters are not convex or isotropic
    - Transform them to convex and isotropic ones through feature scaling



    - Or it's better to use other clustering algorithms

K-Means clustering

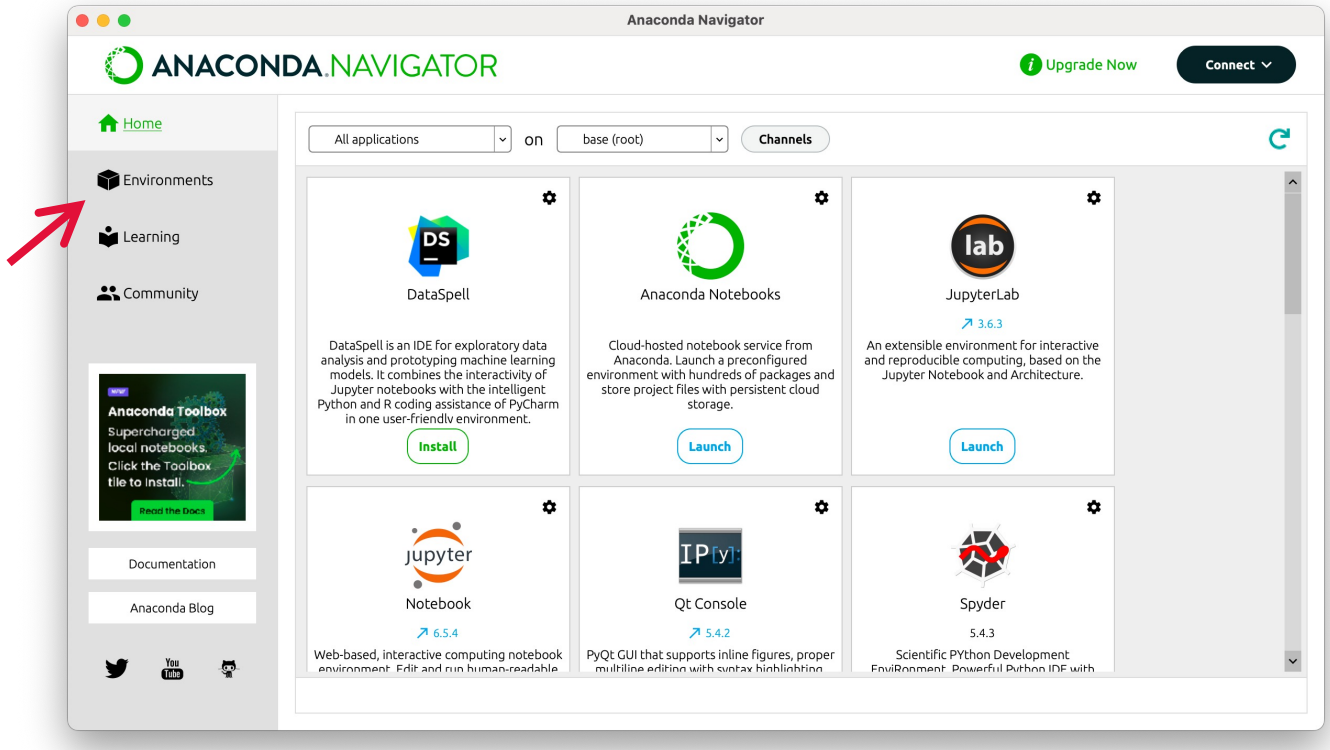
- Matters need attention:
  - Getting clustering results is not the final goal.
  - The final goal is to interpret the clustering results and get insights.
  - Interpretation requires to understand the dataset
  - Perform data exploration before clustering to gain a better understanding
    - Also helpful to determine whether:
      - Keep or delete a specific feature
      - Perform feature sclaing
      - Delete outliers
      - …

Hands-on Exercise

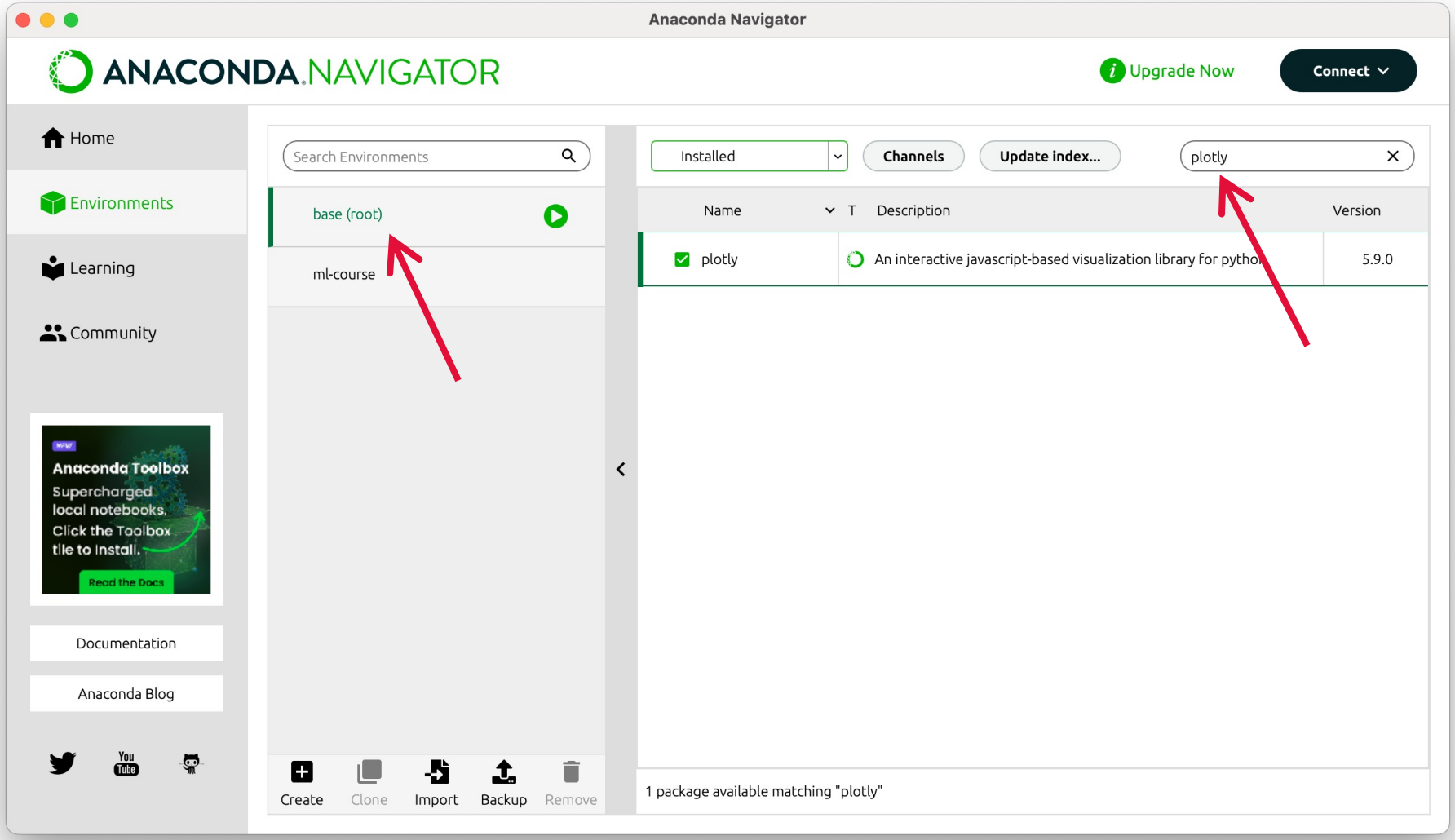
- Exercise 06 Clustering I
  - First install a new library ‘plotly’

Install ‘plotly’ library

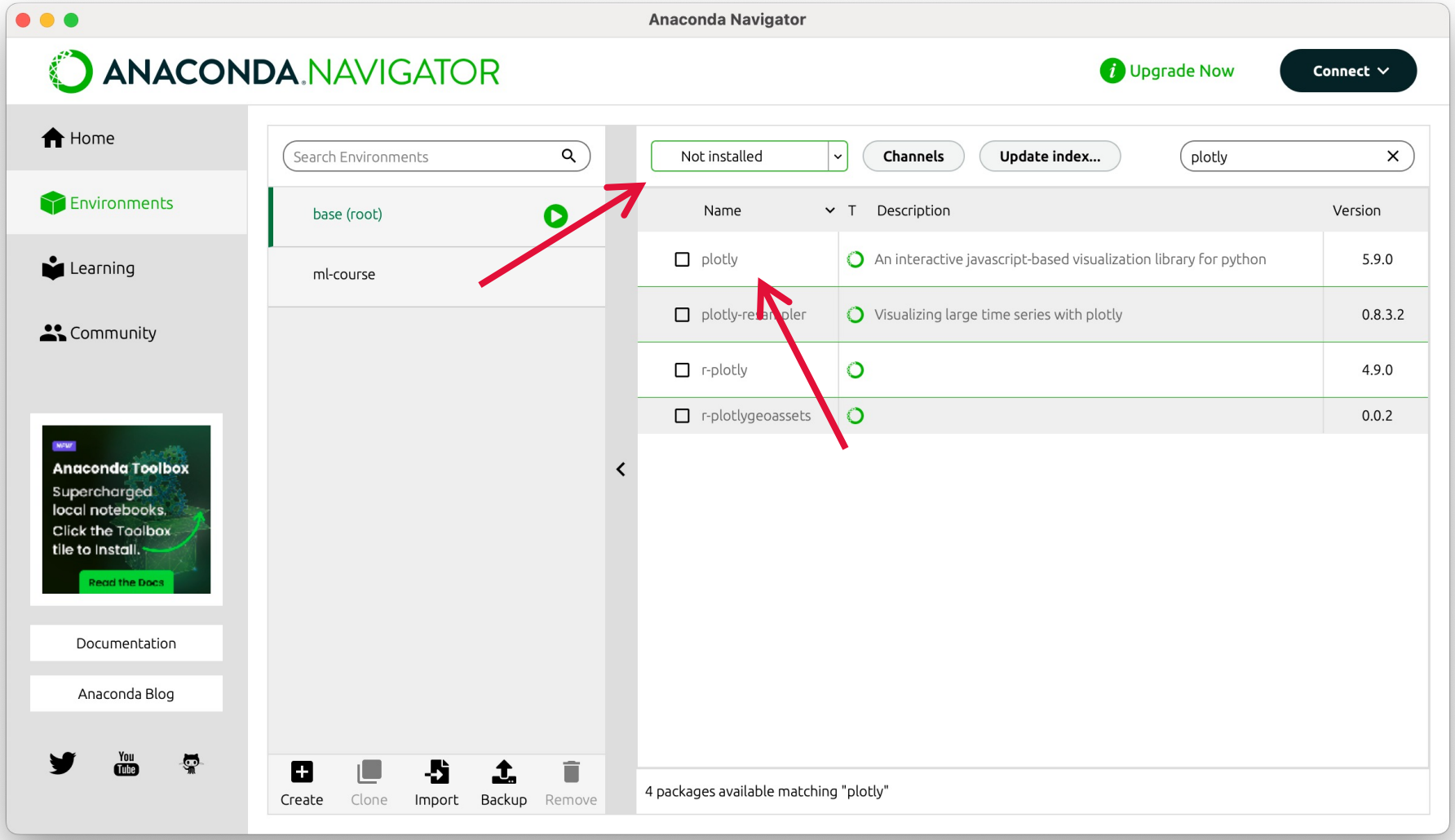
- Open Anaconda Navigator



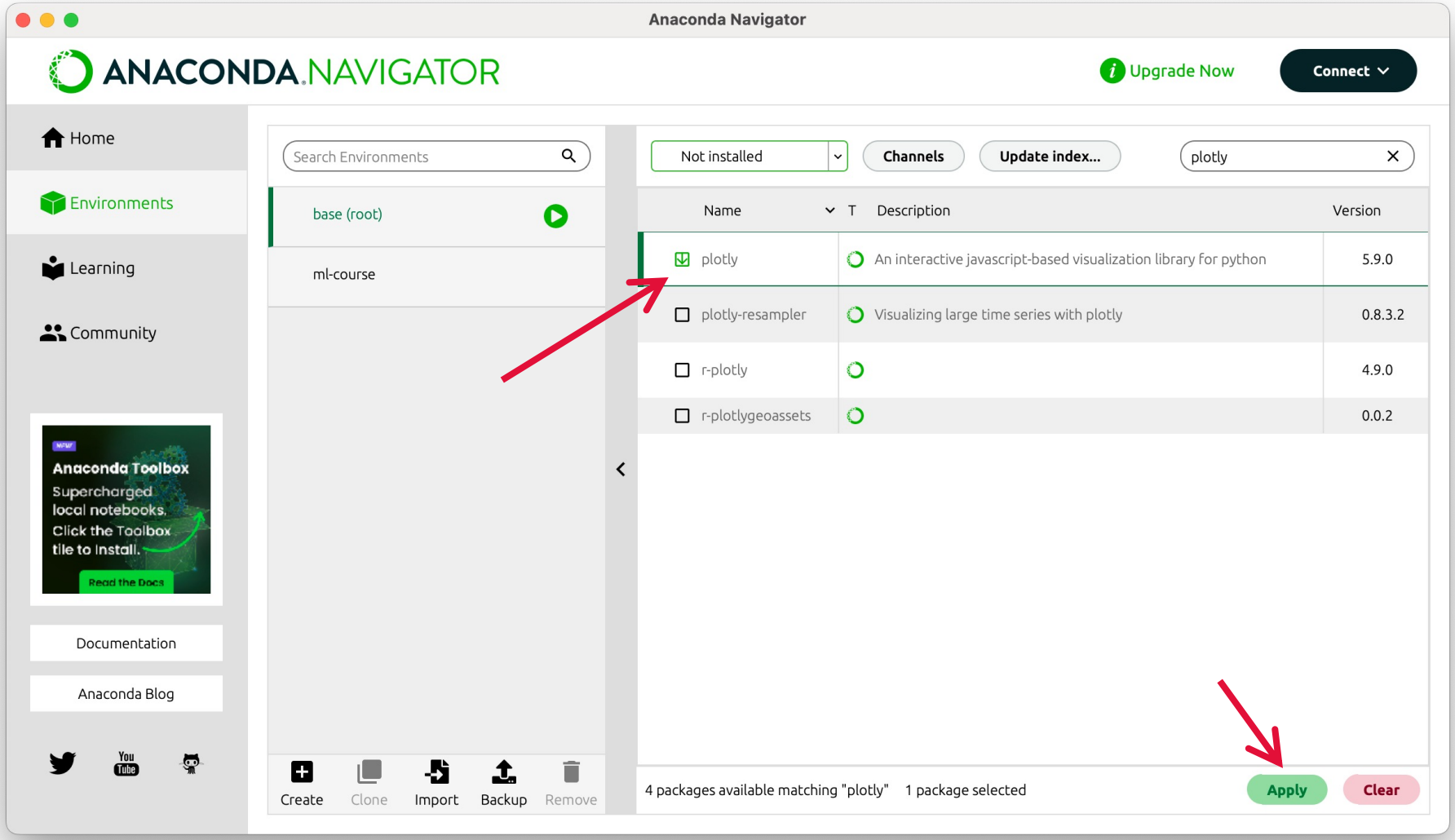
Install ‘plotly’ library



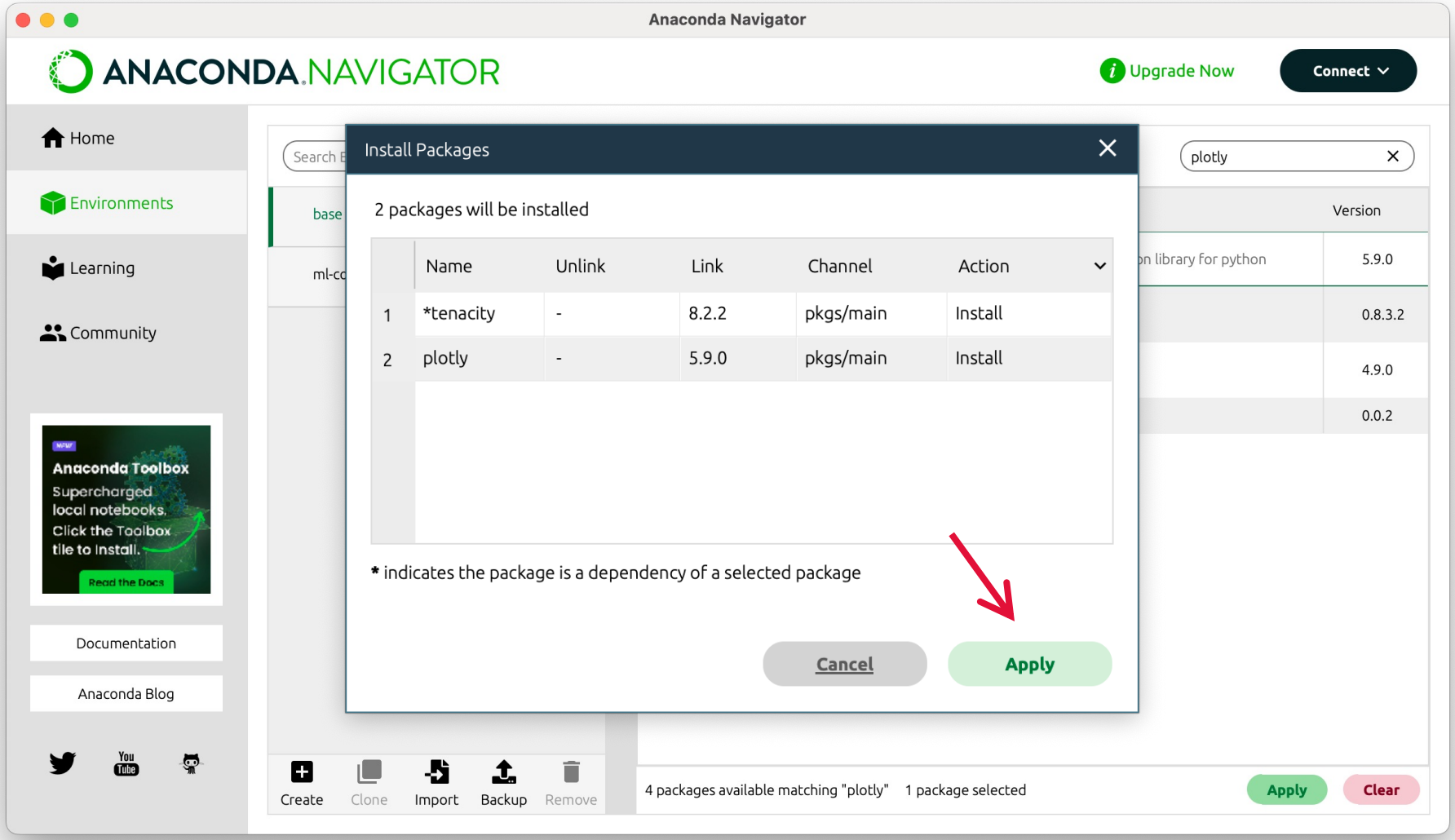
Install ‘plotly’ library



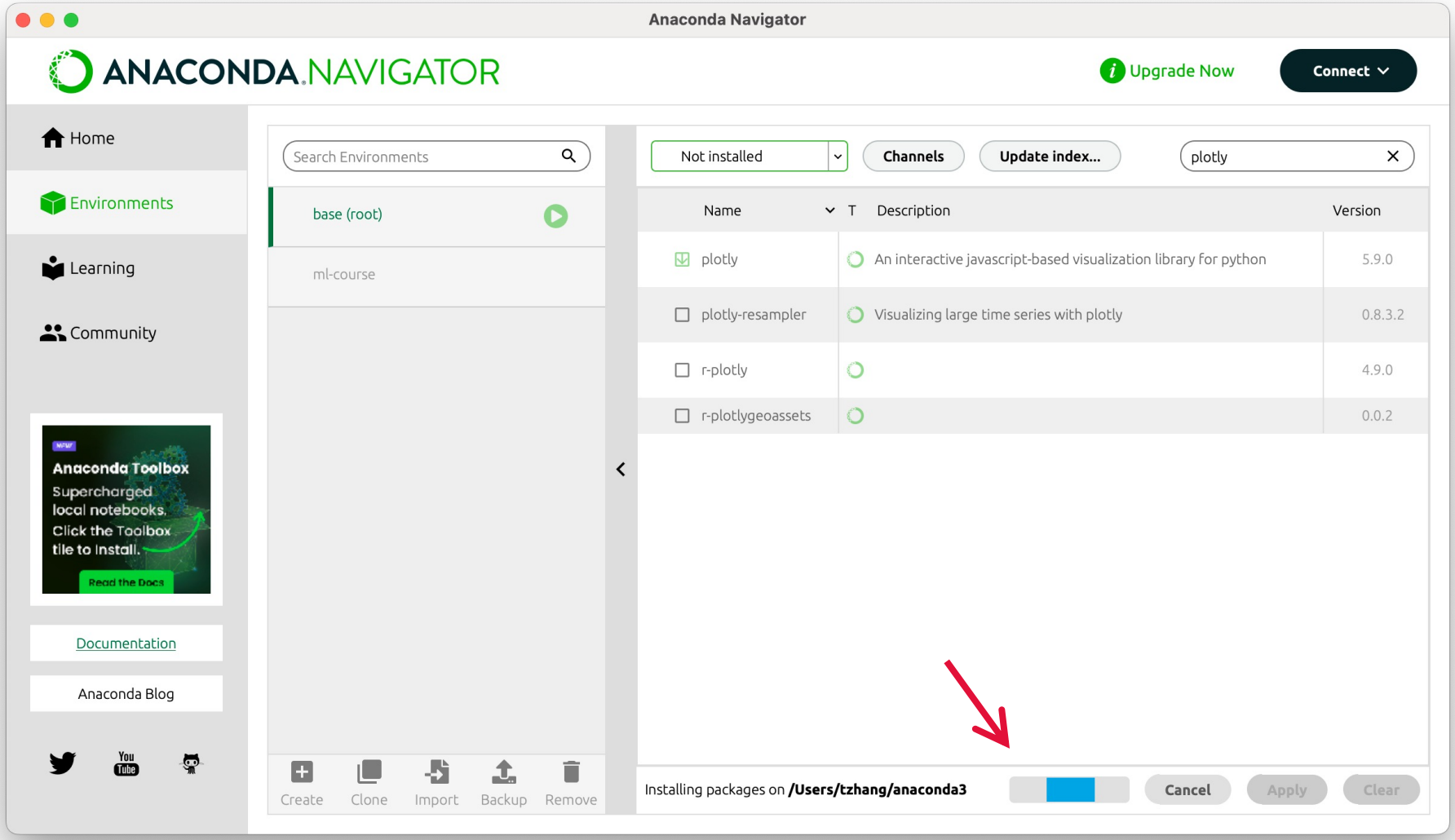
Install ‘plotly’ library



Install ‘plotly’ library



Install ‘plotly’ library



Install ‘plotly’ library