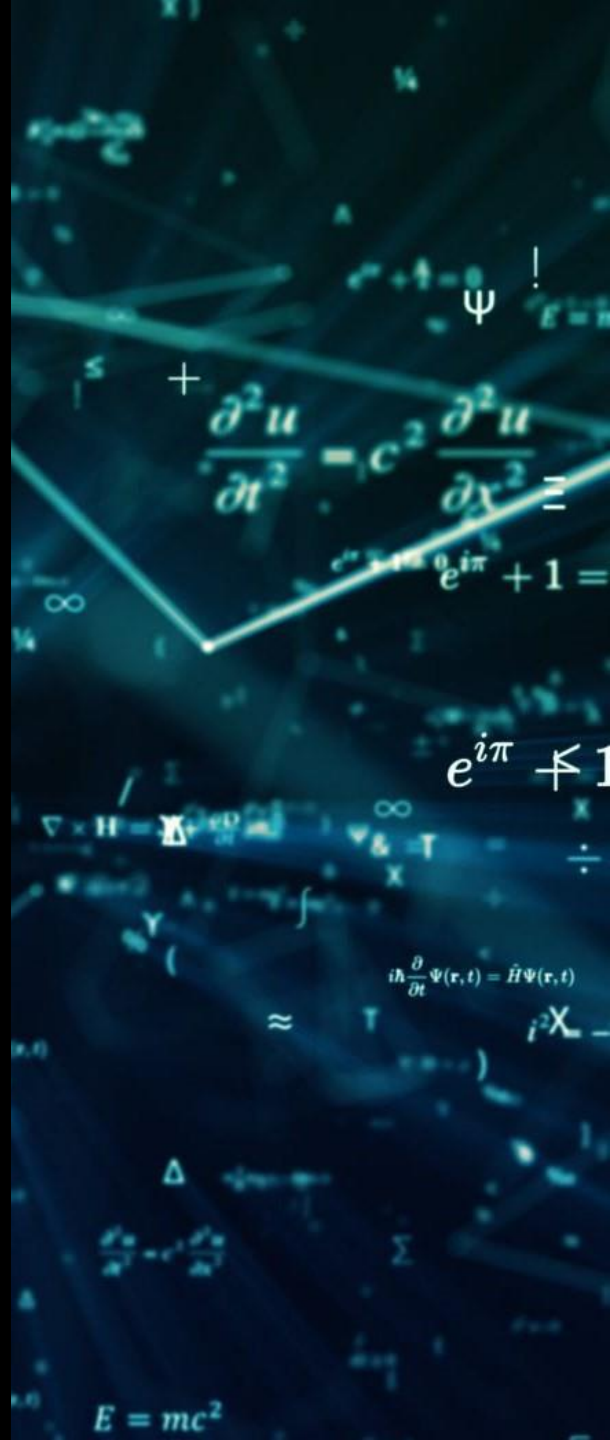
Artificial Intelligence Algorithms and Mathematics

CSCN 8000

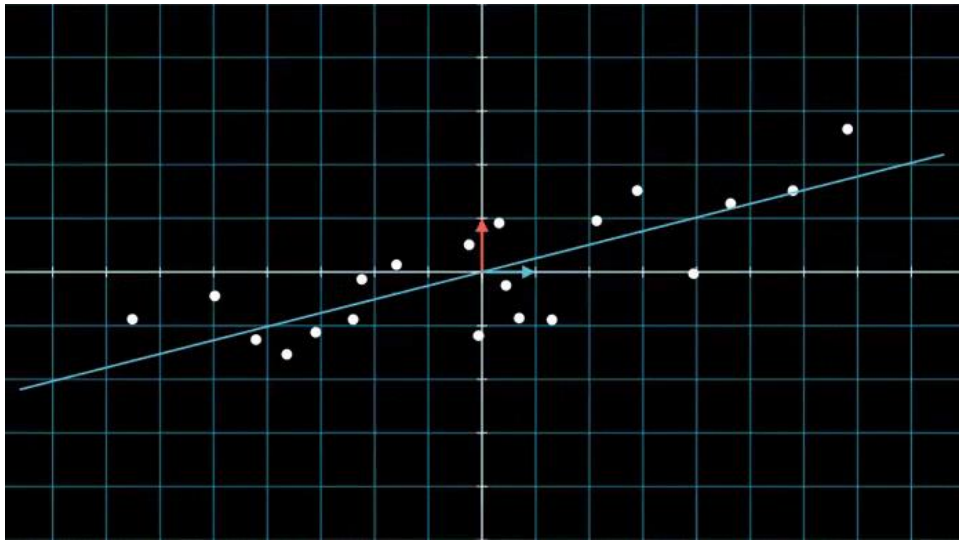


Unsupervised

- FDA/LDA
- K-Means
- Hierarchical Clustering

Recall: Dimensionality Reduction

- The efficiency of ML methods depends crucially on the choice of features that are used to characterize data points.
- Target → have a small number of highly relevant features to characterize data points.
- Dimensionality Reduction techniques reduce the number of input variables or features in a dataset *while retaining its essential characteristics*
- Benefits of dimensionality reduction:
  - Reduce excessive resource requirements
  - Reduce the probability of overfitting
  - Make data visualizations easier



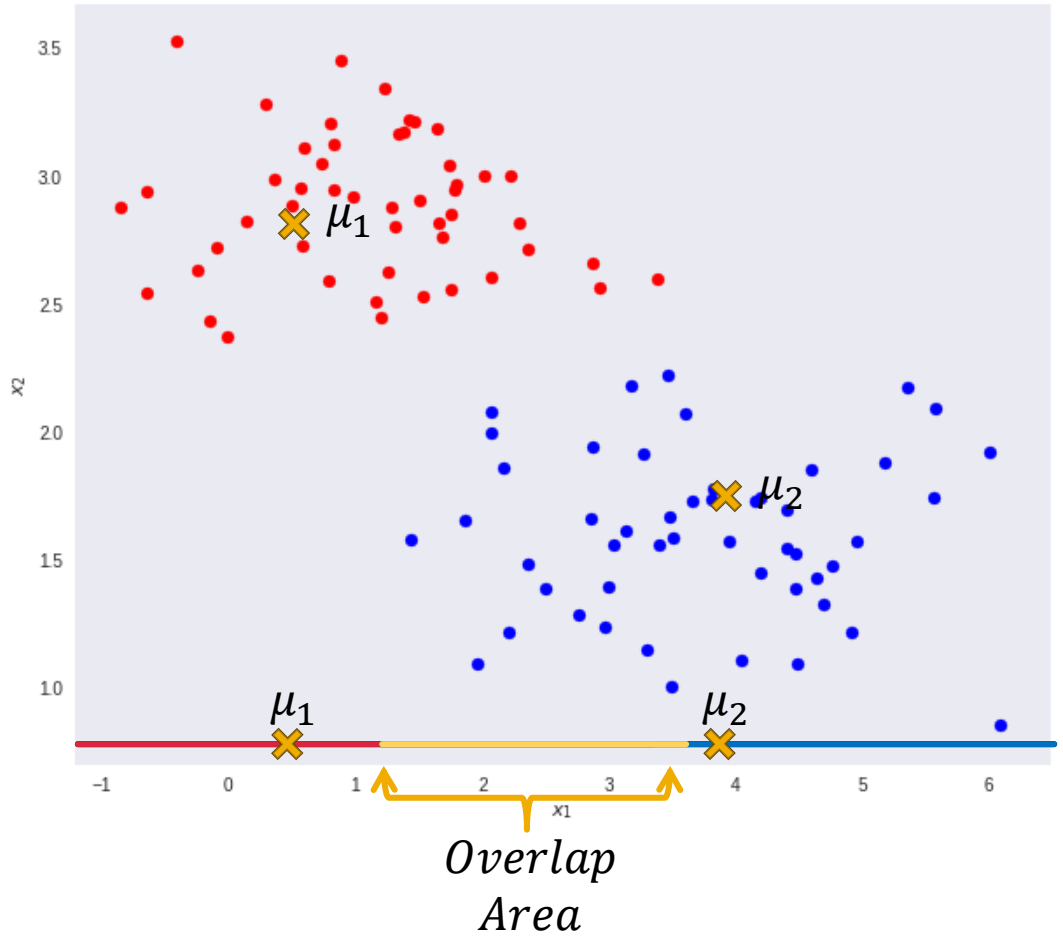
Principal Component Analysis (PCA)

$$\boldsymbol{Su = \lambda u}$$

- To project original data $X \in R^{D*N}$ to P features, where $P \leq D$:
  - Calculate the Covariance Matrix $S = \frac{1}{N}(X - \bar{X})(X - \bar{X})^T$
  - Get all the possible eigenvalues and eigenvectors of S.
  - Sort the eigenvalues in descending order:
    - The Largest eigenvalue corresponds to the (eigenvector) axis with highest variance of data projected on that axis.
    - Lower eigenvalues correspond to axes that are worse in preserving the characteristics of the data.
  - Get the highest P eigenvalues and their eigenvectors.
  - Construct full matrix $U \in R^{P*D}$ by stacking all chosen eigenvectors vertically (row-wise)
  - Get the final full projected dataset $Z \in R^{P*N} \rightarrow Z = UX^T$

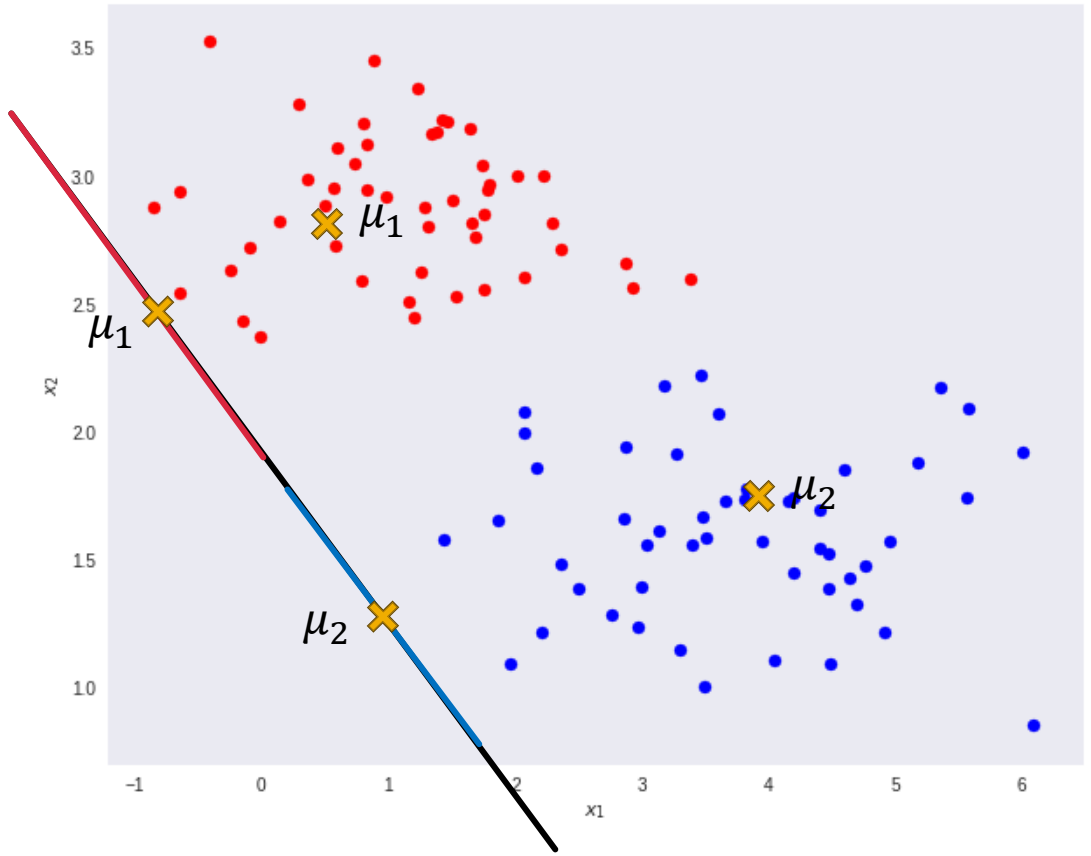
Fisher Discriminant Analysis (FDA)

- Assume we want to project our features to fewer dimensions *while maintaining the separability of our classes.*
- A possible approach would be to ***maximize*** the distance between the centers (means) of the projected classes.
- In the following example, does the proposed axis maintain the best separability between the classes?

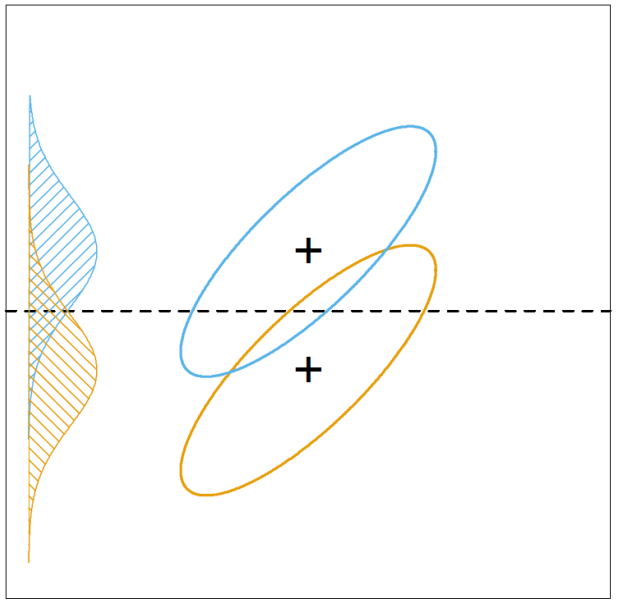


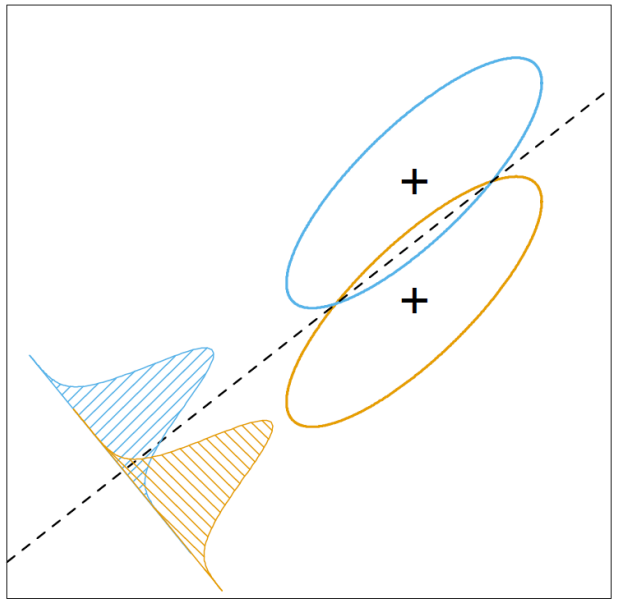
Fisher Discriminant Analysis (FDA)

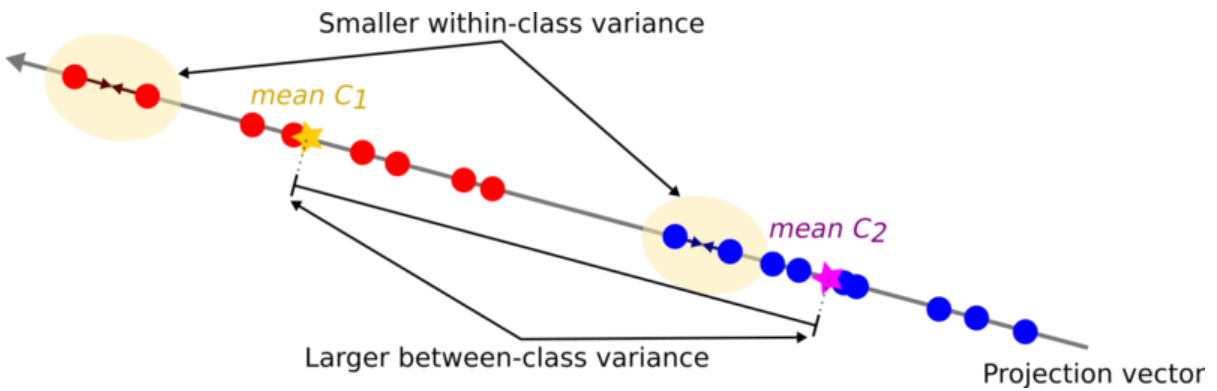
- It looks like maximizing the distance between means of projected classes is not enough if the classes are wide-spread with high variance.
- An additional constraint could be imposed by *minimizing the within-class variance of the projected data.*
- Combining the two constraints leads to a new axis (dimensionality) that maintains the separability of the original data with minimum or no overlap.



Fisher Discriminant Analysis (FDA)







Fisher Discriminant Analysis (FDA)

- Fisher's Linear Discriminant Analysis (FDA) is a linear dimensionality reduction technique that aims to project high-dimensional data into a lower-dimensional space while maximizing the separation between classes.
- This is achieved through two targets:
  - **T1: Maximizing Inter-Class Variance (Distance between class means)**
  - **T2: Minimizing the Intra-Class Variance (Within-class variance).**

Fisher Discriminant Analysis (FDA)

- Assume that we have two classes to be projected.
- We will apply a linear transformation $u \in R^{D*1}$ on each original data point $x_{\{0,1\}} \in R^{D*1}$ belonging to classes $0\ and\ 1$, such that the value z of the projected point at the new axis is formulated as,

$$z = u^T X$$

- The means of the points in each class are formulated as,
  - $\mu_0 = \frac{1}{N_0} \sum_{i=0}^{N_0} x_0^i$, $\mu_1 = \frac{1}{N_1} \sum_{i=0}^{N_1} x_1^i$

Fisher Discriminant Analysis (FDA)

- $\mu_0 = \frac{1}{N_0}\sum_{i=0}^{N_0} x_0^i$, $\mu_1 = \frac{1}{N_1}\sum_{i=0}^{N_1} x_1^i$
- **For Target 1:** Distance between the projected means is formulated as:
  - $\left(u^T\mu_0 - u^T\mu_1\right)^2 = \left(u^T\mu_0 - u^T\mu_1\right)^T \left(u^T\mu_0 - u^T\mu_1\right)$
  - $\left(u^T\mu_0 - u^T\mu_1\right)^2 = (\mu_0 - \mu_1)^T uu^T(\mu_0 - \mu_1)$
  - $\left(u^T\mu_0 - u^T\mu_1\right)^2 = u^T(\mu_0 - \mu_1)(\mu_0 - \mu_1)^T u$
  - $\left(u^T\mu_0 - u^T\mu_1\right)^2 = \boldsymbol{u^T S_B u}$, $S_B = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T$
  - Where S_B represents the distance between class means before projection.

Fisher Discriminant Analysis (FDA)

- Recall from PCA → $Cov(Z) = u^T Su, \quad where\ S = Cov(X)$
- **For Target 2:** Within class-variance for the two classes can be formulated as:
  - $Cov(Z_0 + Z_1) = Cov(Z_0) + Cov(Z_1)$
  - $Cov(Z_0 + Z_1) = u^T S_0 u + u^T S_1 u, \quad where\ S_0 = Cov(X_0), S_1 = Cov(X_1)$
  - $Cov(Z_0 + Z_1) = u^T (S_0 + S_1) u = \boldsymbol{u^T S_W u}, \quad S_W = S_0 + S_1$
  - Where S_W represents the within-class variance of the two classes altogether.

Fisher Discriminant Analysis (FDA)

- To Achieve both **Target 1 and Target 2**, our target could be formulated as follows:

$$maximize\ \frac{u^T S_B u}{u^T S_W u}$$

- Recall that our new axis needs to be unit vector (from PCA), to enforce it we can formulate the target as follows:

$$maximize\ \ u^T S_B u, \qquad s.t.\ \ u^T S_W u = 1$$

- To formulate it as a loss function, we will borrow the concepts from Lagrangian Multipliers:

$$L(u, \lambda) = -\left(u^T S_B u - \lambda\left(u^T S_W u - 1\right)\right)$$

  - The negative sign is added to minimize rather than maximize

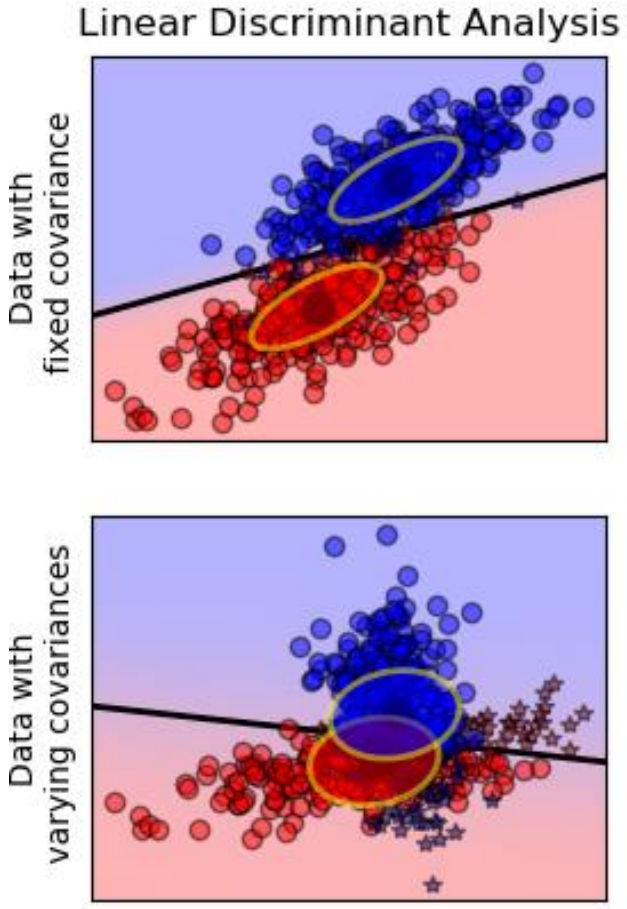
Fisher Discriminant Analysis (FDA)

$$L(u, \lambda) = -\left(u^T S_B u - \lambda\left(u^T S_W u - 1\right)\right)$$

- To minimize the loss with respect to u → Solve $\frac{dL}{du} = 0$
- $\frac{dL}{du} = 2S_B u - 2\lambda S_W u = 0$
- $S_B u = \lambda S_W u \rightarrow \left[\boldsymbol{S_W^{-1} S_B}\right]\boldsymbol{u} = \boldsymbol{\lambda u}$
- We reach a formulation exactly similar to the one of *eigenvalues and eigenvectors*.
- In other words, u is considered an eigenvector of the matrix $\boldsymbol{S_W^{-1} S_B}$ calculated from the original data and λ is the associated eigenvalue.
- To transform the full dataset, follow the same steps as PCA, but with calculating $\boldsymbol{S_W^{-1} S_B}$ in the first step instead.

LDA vs FDA

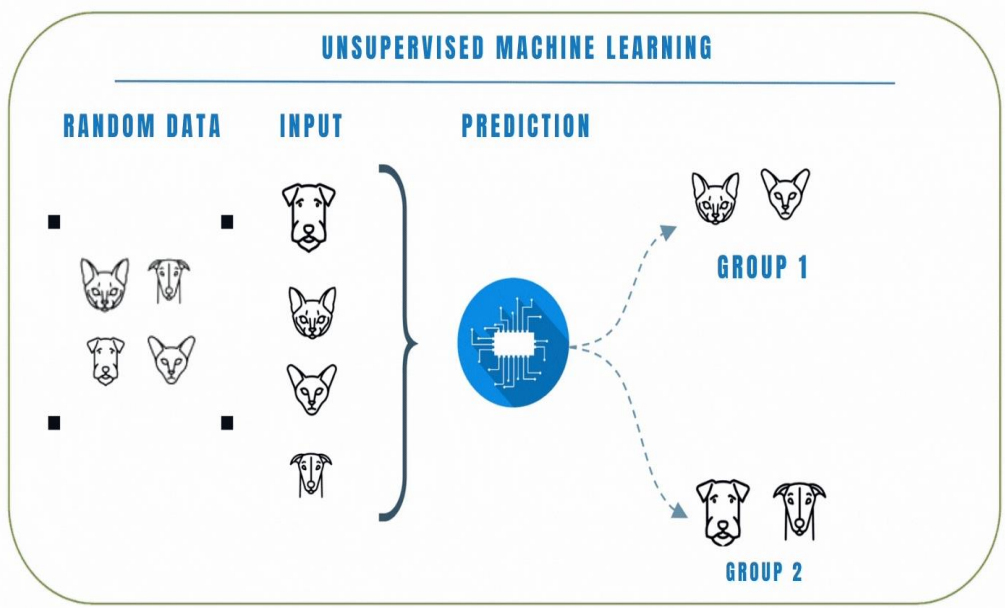
- Both Linear Discriminant Analysis (LDA) and FDA refer to the same technique which aims to project the data to lower dimensions while maximizing the class separability.
- LDA is the direct extension of FDA to work with two **or more** classes.
- LDA is not only doing dimensionality reduction, but also computes the *linear decision boundary* between the classes in the projected space.
- LDA makes important assumptions about the shape of the data:
  - All classes follow a gaussian (normal) distribution
  - All classes have equal (identical) covariance matrices.
- If any of those assumptions doesn't hold, LDA won't perform well in classification or dimensionality reduction.



Unsupervised Learning

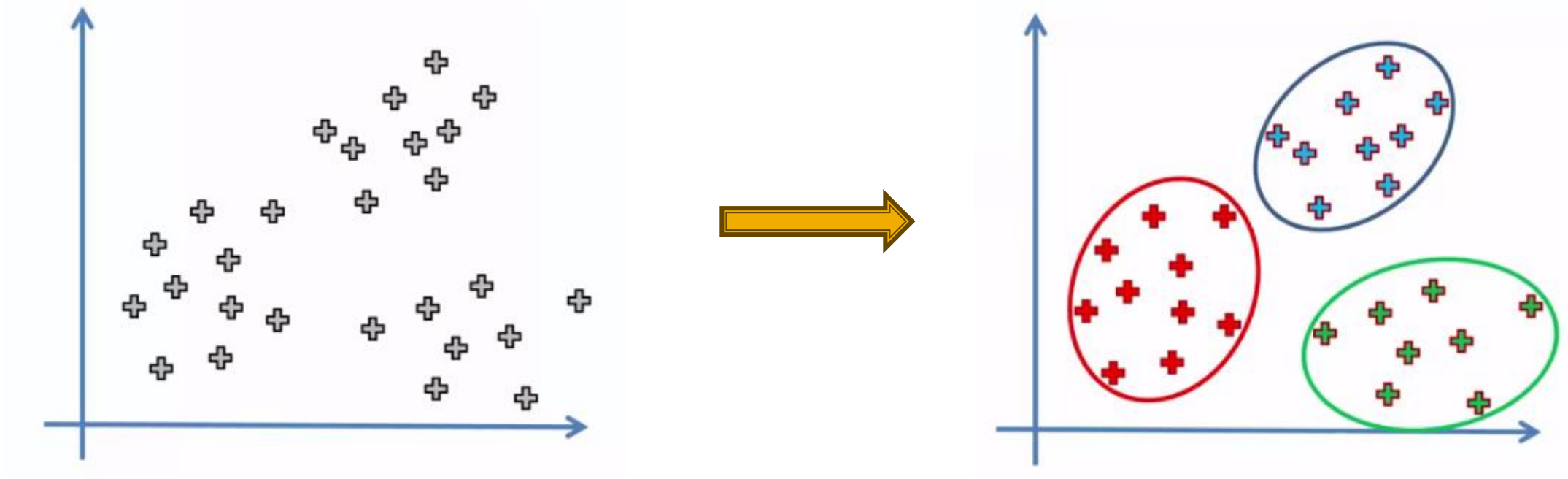
Unsupervised Learning

- Unsupervised learning is a type of machine learning where the algorithm is given data without explicit instructions on what to do with it.
- The system tries to learn the patterns and the structure of the data without any labeled responses to guide the learning process.
- Benefits:
  - Uncover hidden patterns and structures
  - Adaptable to various types of data without the need for labeled examples
  - Valuable tool for exploratory data analysis



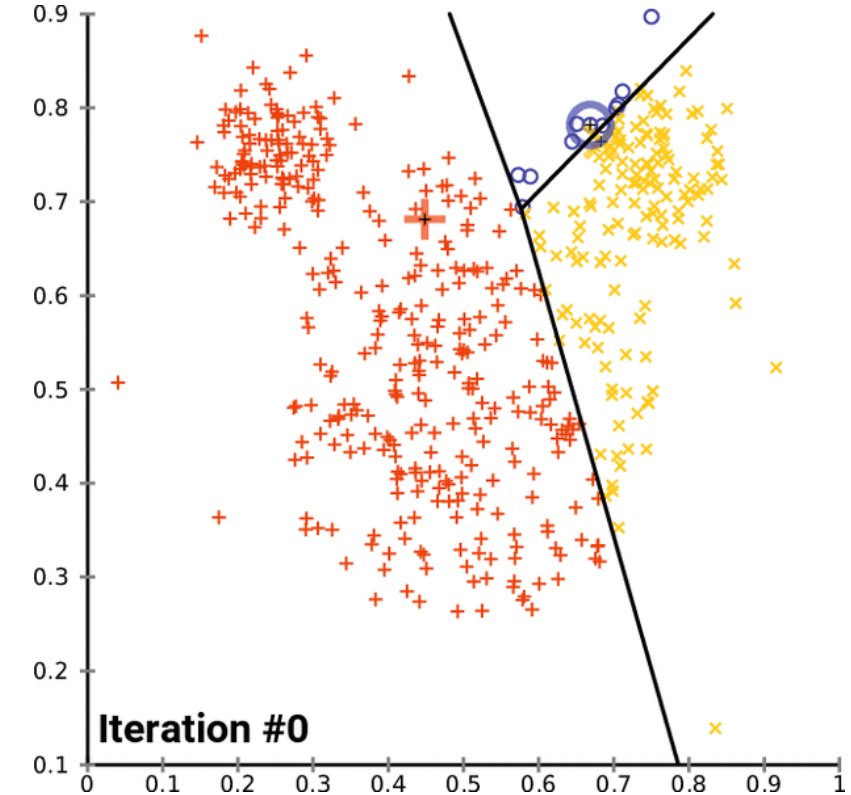
K-Means Clustering

- Given the following dataset, can we group the points into 3 meaningful clusters that are sufficiently far from each other?



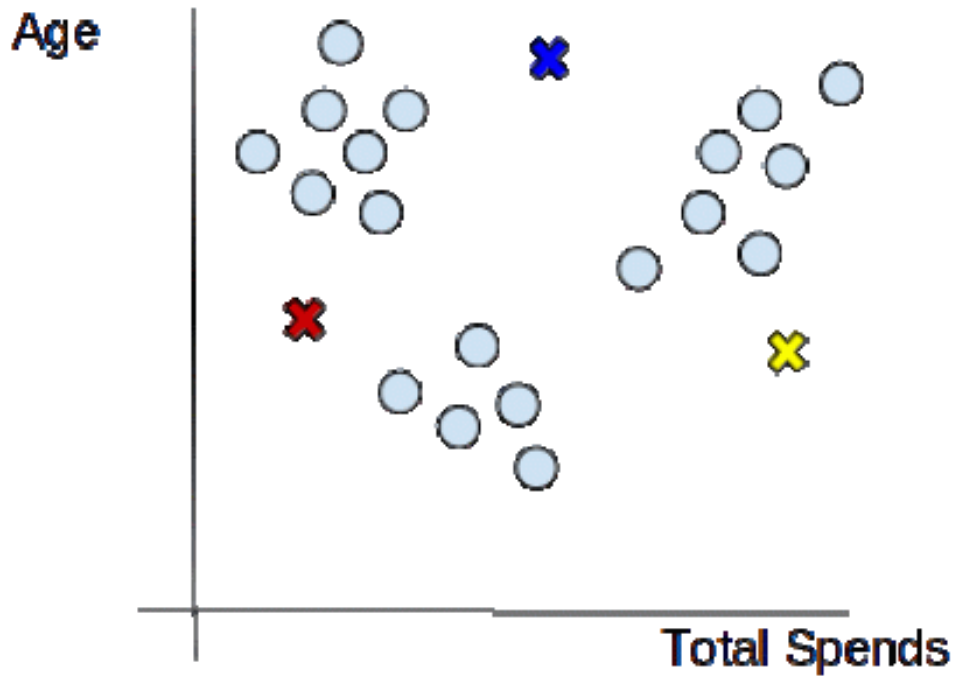
K-Means Clustering

- K-Means is a popular unsupervised machine learning algorithm used for clustering data into groups or clusters based on similarity.
- **The primary goal** of K-Means is to partition data points into K clusters, where each point belongs to the cluster with the *nearest mean*.
- K is a hyperparameter manually set to determine the number of clusters.



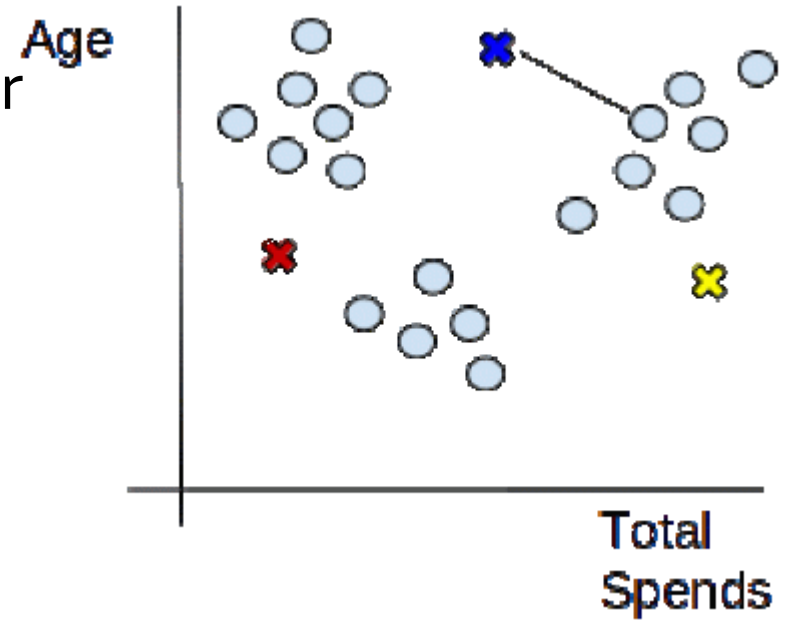
K-Means Clustering

- Assume we want to cluster the following dataset into 3 clusters where $K = 3$.
- *Step 1: Initialize Clusters:*
  - Choose a strategy to initialize the means (centers) of the 3 clusters.
    - A popular strategy is just to choose 3 random points to define the cluster means.
    - Other methods include: Naiive Shardin and K-Means++



K-Means Clustering

- *Step 2: Assign Points to Clusters:*
  - For each point in the dataset, calculate the distance between the points and the K-Cluster means (centers).
    - $d(x_i, \mu_k) = \sqrt{\sum_{d=1}^{D} \left(x_i^d - \mu_k^d\right)^2}$
  - Assign the point to the cluster closest to it (smallest distance).
  - Repeat this step until all points in the dataset are assigned successfully to a cluster.

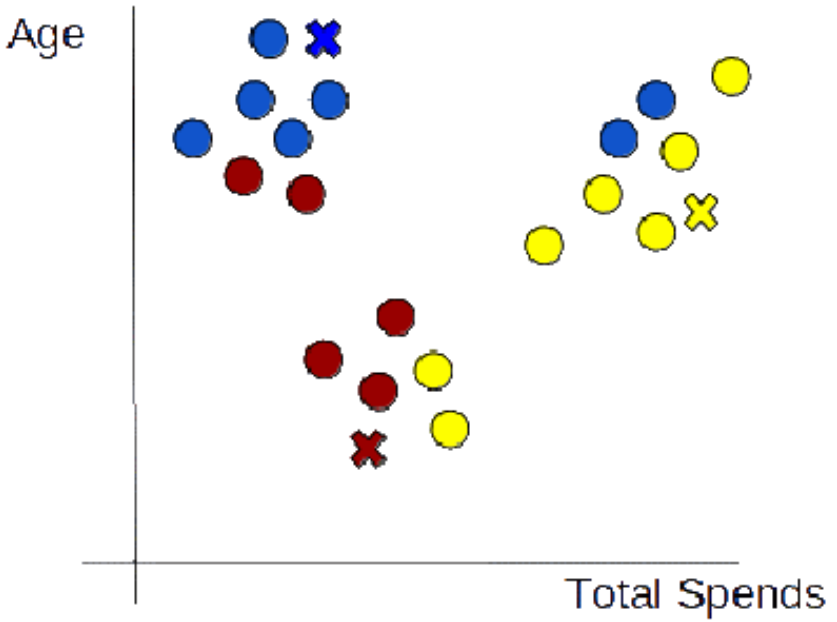


K-Means Clustering

- *Step 3: Re-calculate Cluster Means:*
  - Now since different points belong to each cluster, we need to recalculate the cluster means (centers), such that:

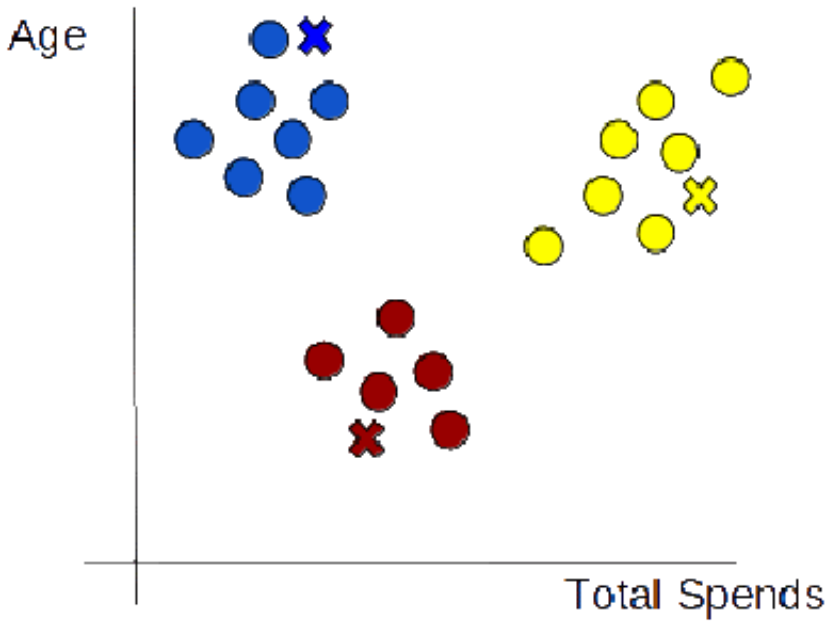
$$\mu_k = \frac{1}{n_k} \sum_{i=0}^{n_k} x_i$$

  - Where μ_k represents the mean of cluster k as the mean of the points belonging to the cluster.
  - n_k represents the number of points belonging to cluster k.



K-Means Clustering

- *Step 4: Repeat steps 2 and 3:*
  - Repeat steps 2 and 3 until the assignment of points to clusters is not changing anymore (saturation).
  - The final assignment of points to clusters will define the optimal K for the current dataset.

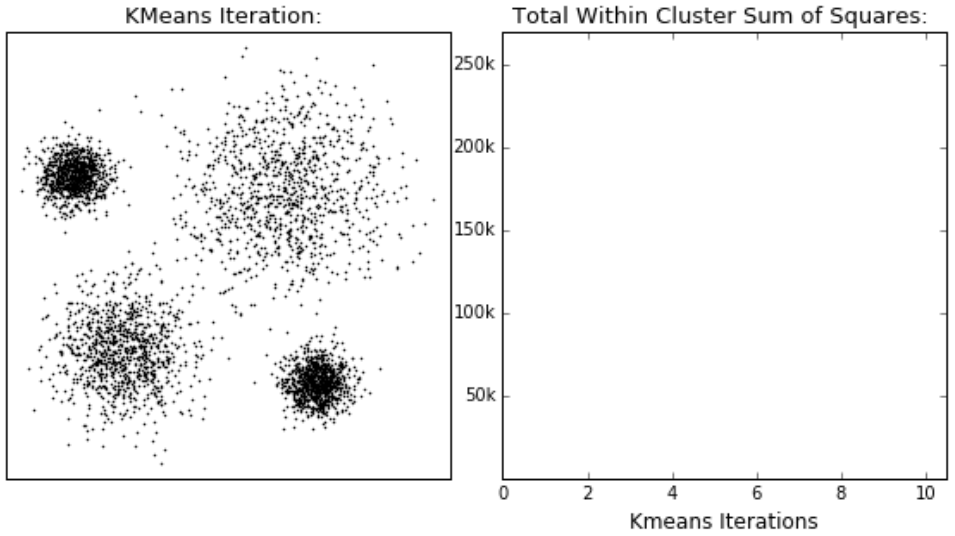


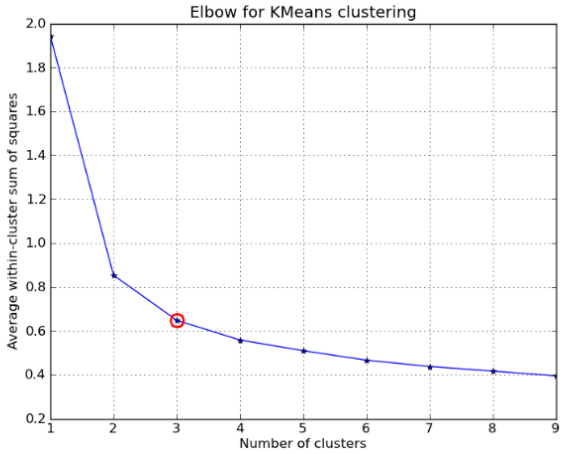
K-Means Clustering

- How to choose optimal K?
  - One method is called the ***Elbow Plot:***
    - Calculate the Within-Cluster-Sum-of-Squares (WCSS) for different values of K:

$$WCSS_K = \sum_{i=1}^{K} \sum_{j=1}^{n_i} \left(x_j - \mu_i\right)^2$$

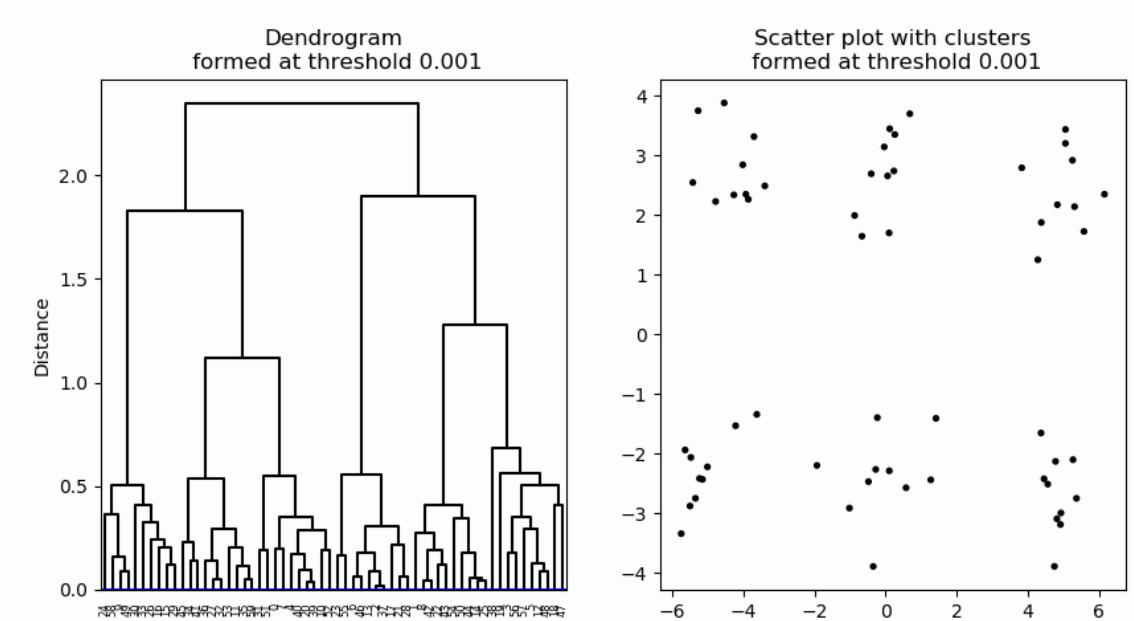
    - Where n_i represents the number of points in cluster i.
    - The WCSS tells us how spread the points in each cluster are. Lower WCSS means better clusters (more compact).
    - Plot the WCSS for different K values and choose the K value where an inflection point (elbow) occurs.





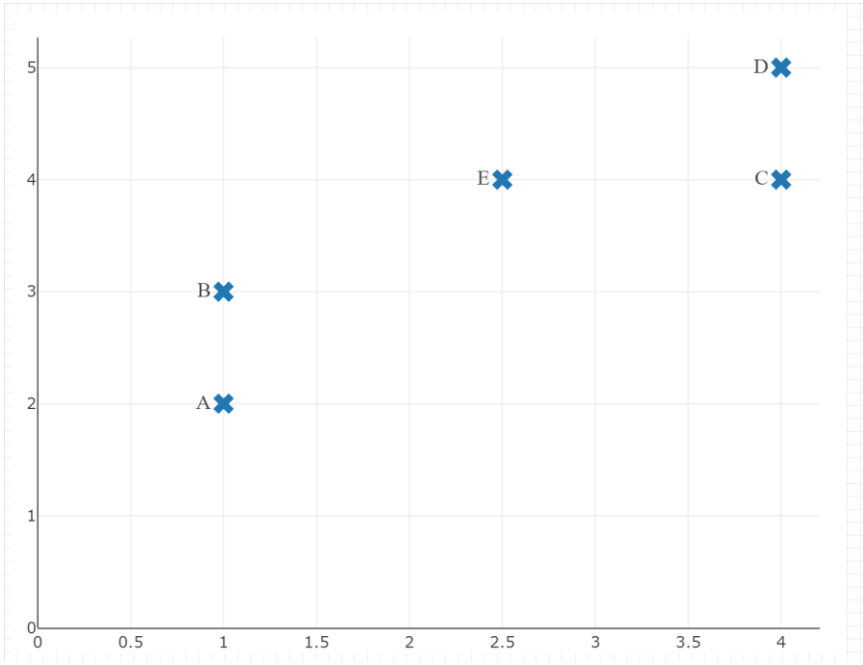
Hierarchical Clustering

- Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters.
- There are two main types of hierarchical clustering: **Agglomerative** and **Divisive.**



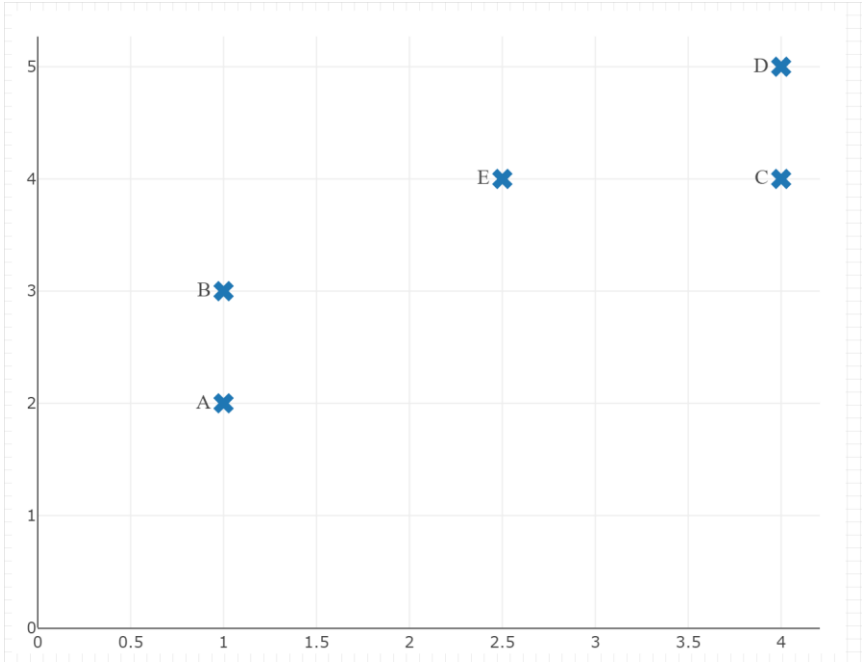
Agglomerative Hierarchical Clustering

- This method is "bottom-up," meaning it starts with each data point as a separate cluster and iteratively merges them into larger clusters.
- Assume that we have the example dataset with 5 points, we will use Agglomerative clustering to combine them into clusters.



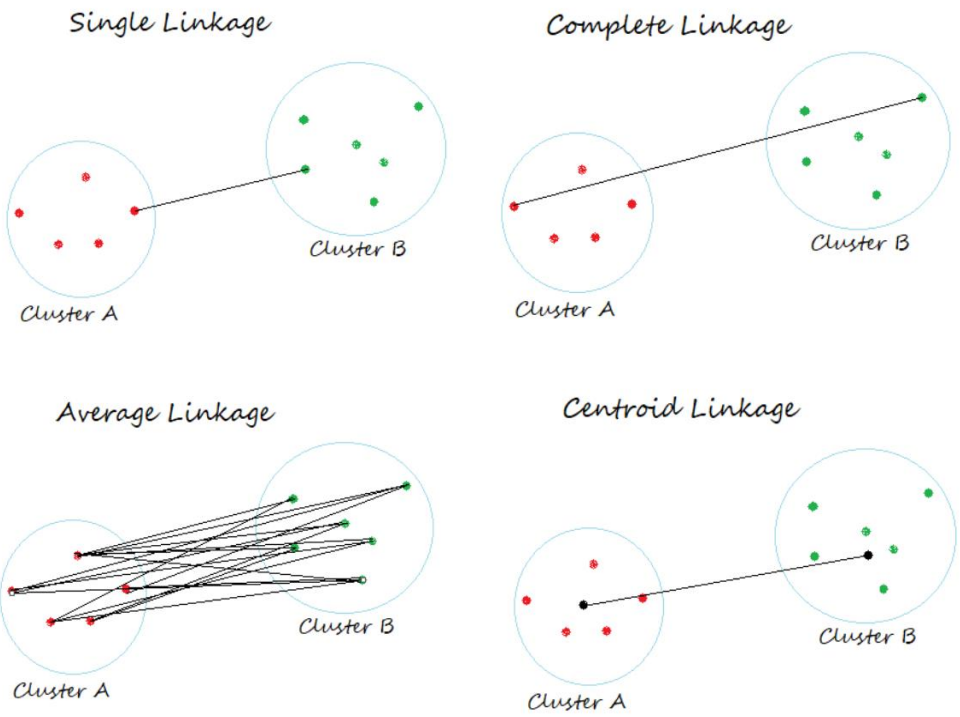
Agglomerative Hierarchical Clustering

- *Step 1: Initialization:*
  - All points are treated as their own clusters. So we start with N clusters.
  - In our example, we start with 5 clusters.



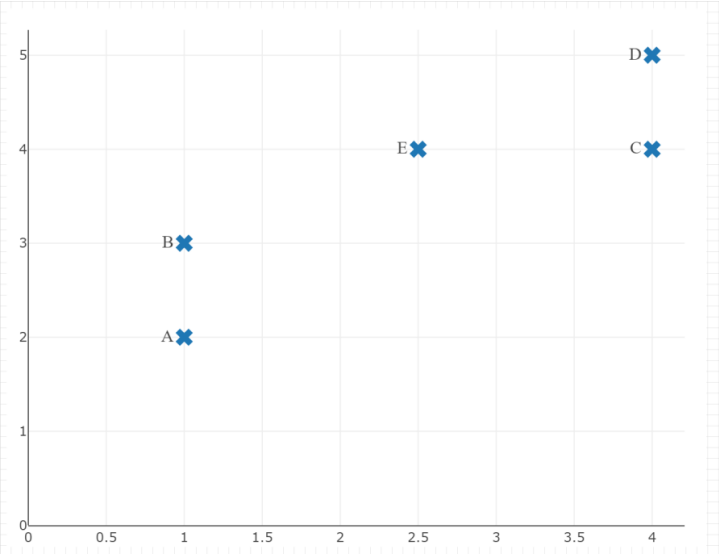
Linkage Criterions

- They determine how the distance between clusters is measured, which directly influences how the clusters are formed. The types are:
  - ***Single Linkage (Nearest Neighbor):*** Uses the minimum distance between members of the two clusters.
    - $d(A,B) = \min_{\{a \in A, b \in B\}} d(a,b)$
  - ***Complete Linkage:*** Uses the maximum distance between members of the two clusters.
    - $d(A,B) = \max_{\{a \in A, b \in B\}} d(a,b)$
  - ***Average Linkage:*** Uses the average distance between all pairs of members in the two clusters.
    - $d(A,B) = \frac{1}{N_A N_B} \sum_{a \in A} \sum_{b \in B} d(a,b)$
  - ***Centroid Linkage:*** Uses the distance between cluster means.
    - $d(A,B) = \|\mu_A - \mu_B\|^2$



Agglomerative Hierarchical Clustering

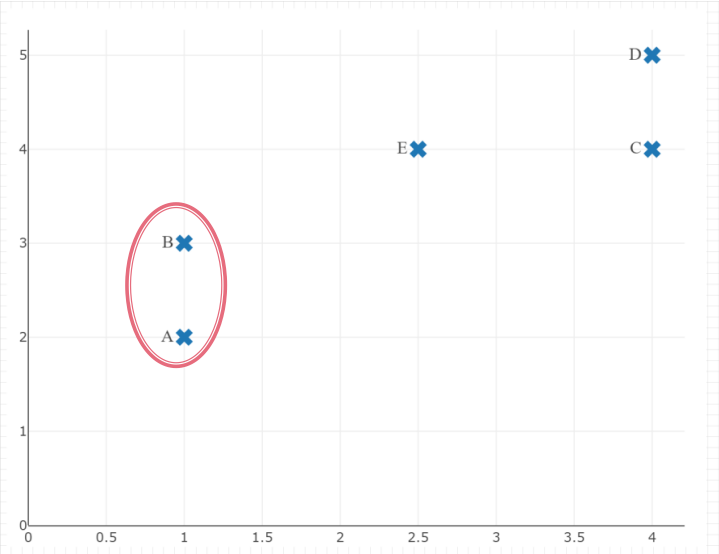
- *Step 2: Distance Matrix Computation:*
  - Calculate a similarity or distance matrix that measures the distances between all pairs of data points. This matrix is $N * N$ size.
  - You could use any distance metric discussed in class
    - Euclidean is the most famous
    - We will use Manhattan in this example
  - Any of the linkage criterion could also be used, we will use Complete Linkage in this example.



	A	B	C	D	E
A					
B	1				
C	5	4			
D	6	5	1		
E	3.5	2.5	1.5	2.5	

Agglomerative Hierarchical Clustering

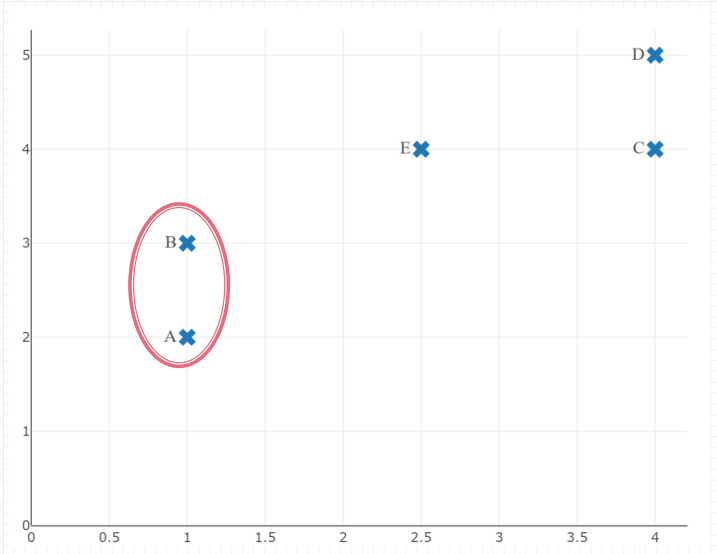
- *Step 3: Merge Closest Cluster:*
  - Find the pair of clusters that are closest to each other based on the chosen *distance metric* and *linkage criterion*.
  - In our example, A-B and C-D show the minimum distances, we could choose any of them to be merged. We'll go with A-B



	A	B	C	D	E
A					
B	1				
C	5	4			
D	6	5	1		
E	3.5	2.5	1.5	2.5	

Agglomerative Hierarchical Clustering

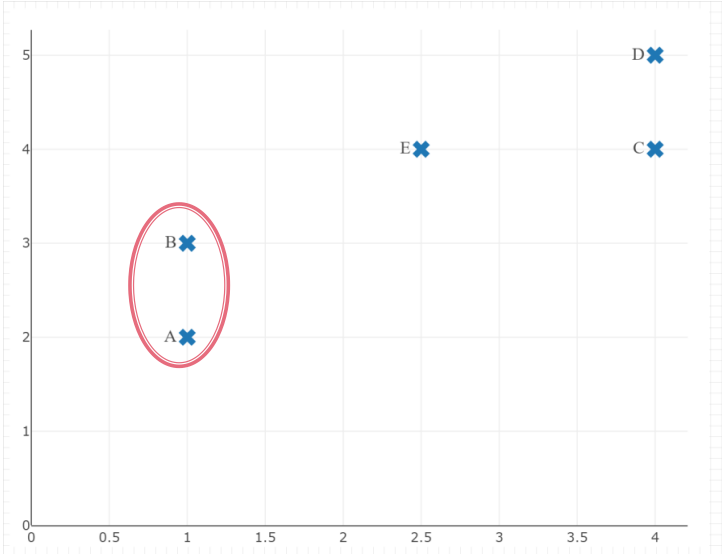
- *Step 4: Recalculate Distance Metric:*
  - Recalculate the distance matrix with the new cluster.
  - Recall that we're using complete linkage and Manhattan distance.



	A-B	C	D	E
A-B				
C	5			
D	6	1		
E	3.5	1.5	2.5	

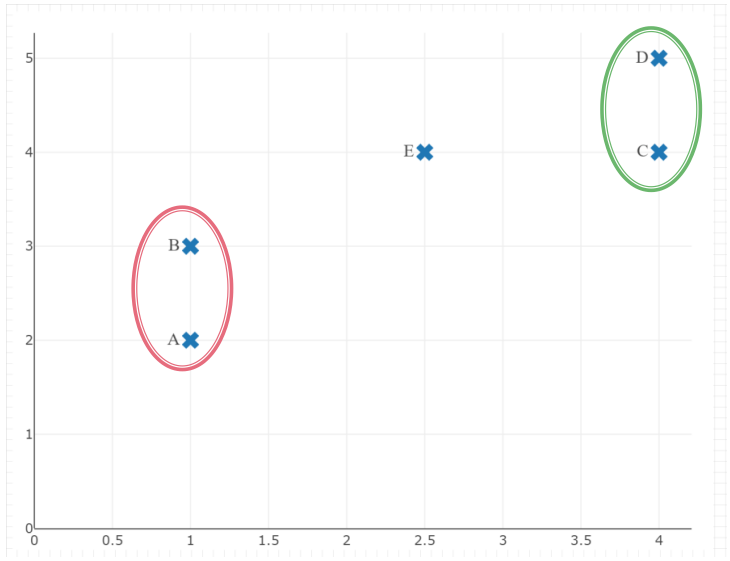
Agglomerative Hierarchical Clustering

- *Step 5: Repeat Steps 3 and 4:*
  - Repeat steps 3 and 4 until all points are merged into one cluster.

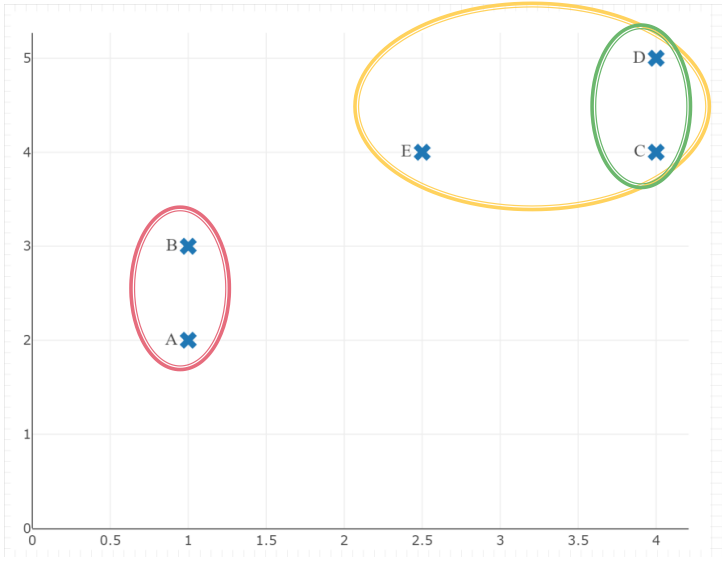


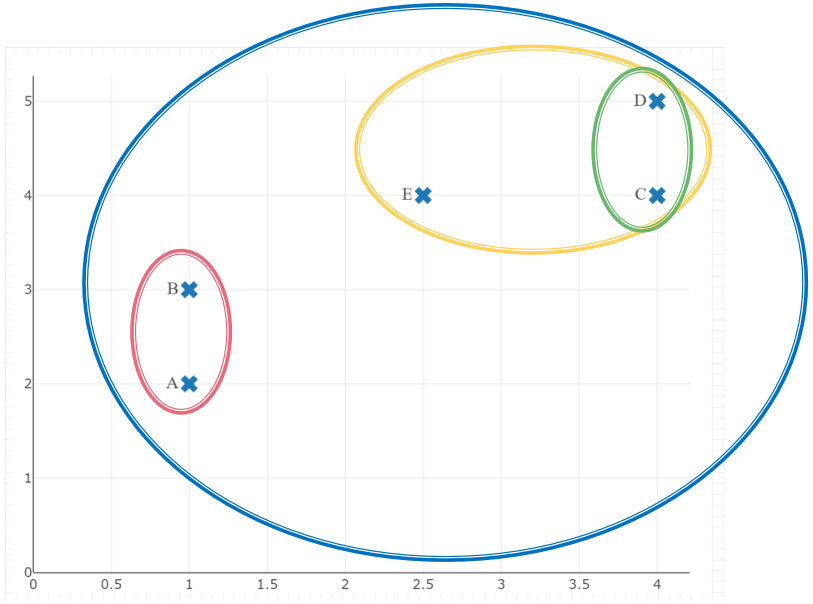
	A-B	C	D	E
A-B				
C	5			
D	6	1		
E	3.5	1.5	2.5	

Agglomerative Hierarchical Clustering

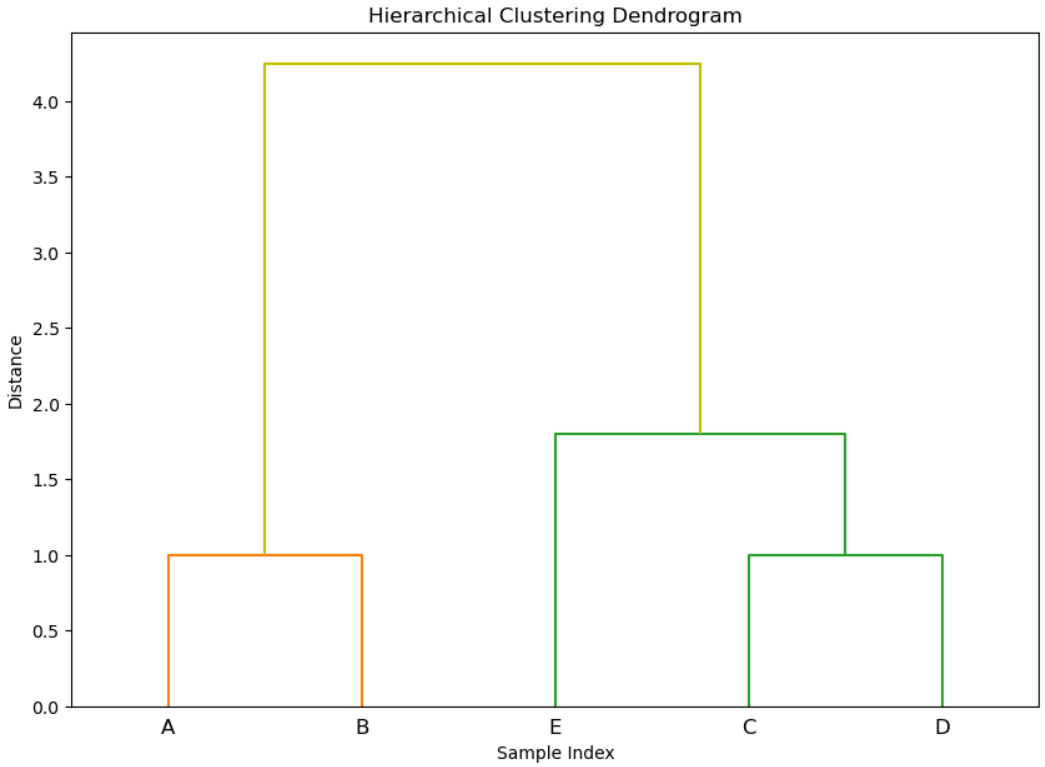


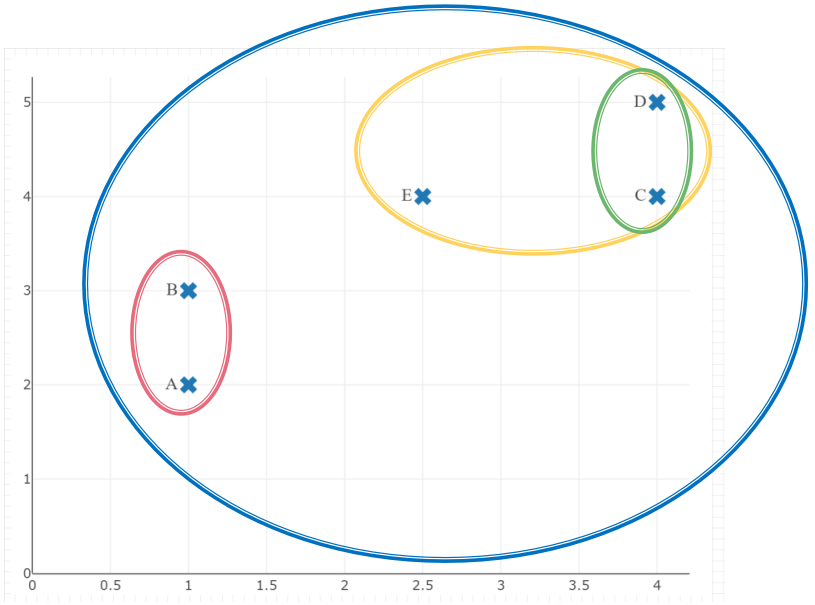
	A-B	C-D	E
A-B			
C-D	6		
E	3.5	2.5	





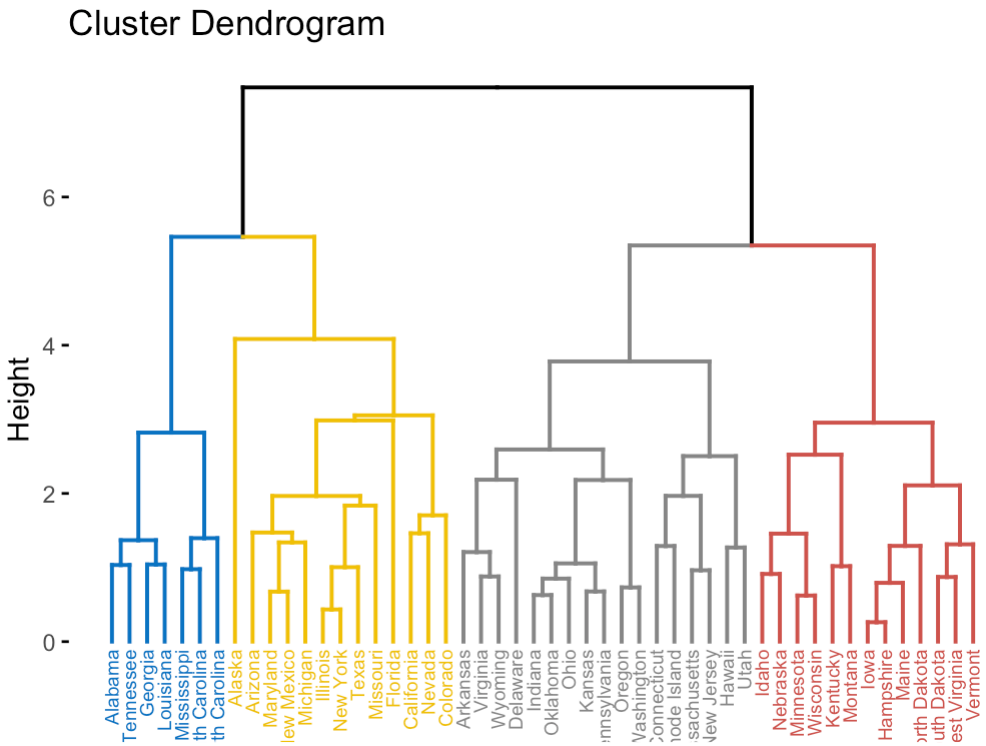
Agglomerative Hierarchical Clustering





Divisive Hierarchical Clustering

- This method is a "top-down" approach to cluster analysis. It begins with all data points in a single cluster and iteratively splits them into smaller clusters.
- Steps:
  - Initialization:
    - Start with one large cluster that includes all data points.
  - Cluster Splitting:
    - At each step, split a cluster into smaller clusters.
    - The splitting is typically based on a criterion that identifies the *'least similar'* members of the cluster.
  - Iterative Division:
    - Continue the process of splitting clusters at each step.
    - This process is repeated recursively until each data point forms its own cluster or a specified number of clusters is reached.
  - Result:
    - The result is often visualized as a dendrogram, which shows the hierarchical relationship between clusters and the order in which splits occurred.



Thank you!

- Any questions?

Disclaimer

Due to nature of the course, various materials have compiled from different open source resources with some moderation. I sincerely acknowledge their hard work and contribution

CONESTOGA

Thank You

Youssef Abdelkareem

yabdelkareem@conestogac.on.ca

**** Credits for Dr. Anasuya Bhima for assisting in the slides structure***