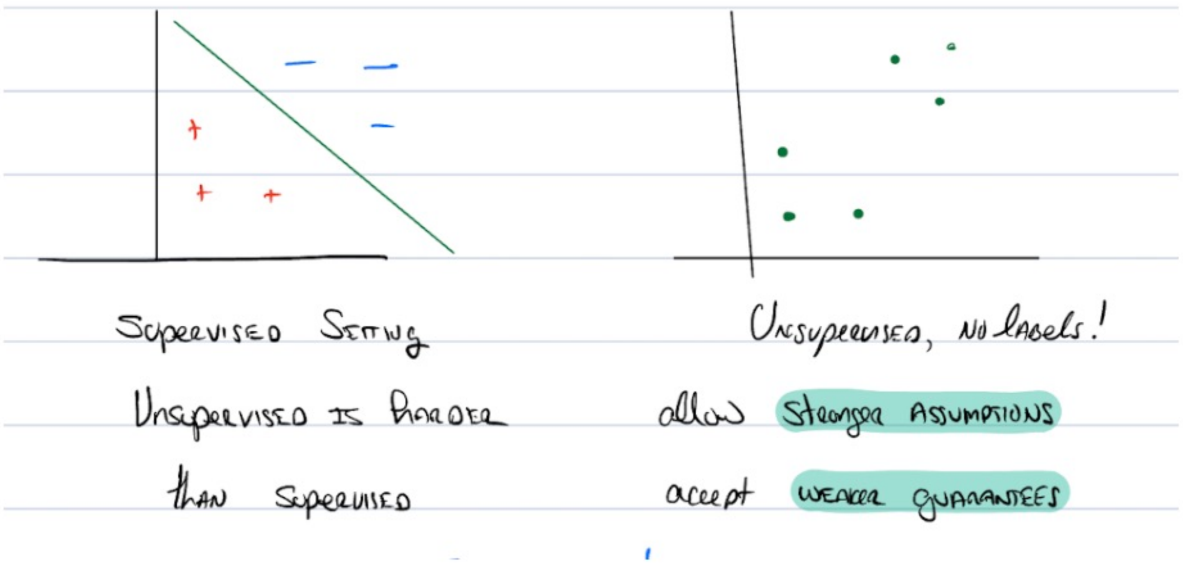
CMSC 478
Unsupervised Learning
K-means Clustering

KMA Solaiman

ksolaima@umbc.edu

Unsupervised Learning In Pictures



Unsupervised learning is "harder" than supervised, so we'll make *stronger* assumptions and accept *weaker guarantees*.

project where you need to predict the sales of a big mart:

Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales
Medium	Tier 1	Supermarket Type1	3735.1380
Medium	Tier 3	Supermarket Type2	443.4228
Medium	Tier 1	Supermarket Type1	2097.2700
NaN	Tier 3	Grocery Store	732.3800
High	Tier 3	Supermarket Type1	994.7052

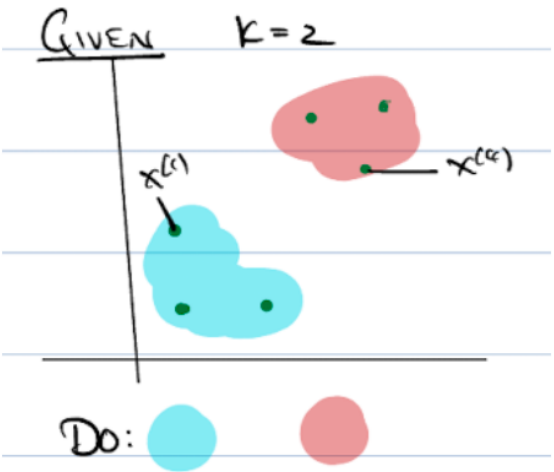
your task is to predict whether a loan will be approved or not:

Loan_ID	Gender	Married	ApplicantIncome	LoanAmount	Loan_Status
LP001002	Male	No	5849	130.0	Y
LP001003	Male	Yes	4583	128.0	N
LP001005	Male	Yes	3000	66.0	Y
LP001006	Male	Yes	2583	120.0	Y
LP001008	Male	No	6000	141.0	Y


High Income
Average Income
Low Income

k-Means (Picture)

Given $k = 2$ and the following data find clusters.



- **Given** an integer k (the number of clusters) and $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$.
- **Do** find an assignment of $x^{(i)}$ to one of the k clusters.

$$C^{(i)} = j \text{ means point } i \text{ in cluster } j$$

e.g., $C^{(2)} = 2$ and $C^{(4)} = 1$

How do we find these clusters? (Iterative Approach)

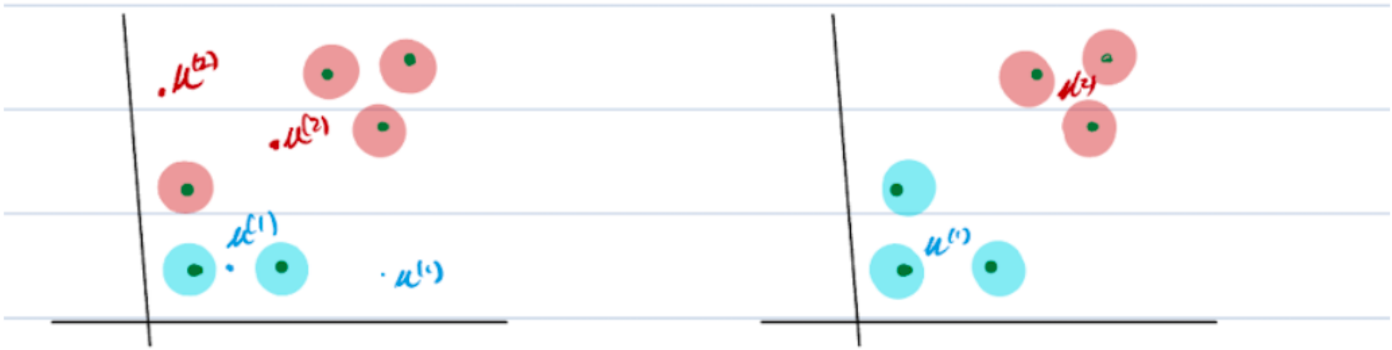
- (Randomly) Initialize Centers $\mu^{(1)}$ and $\mu^{(2)}$.

How do we find these clusters? (Iterative Approach)

- (Randomly) Initialize Centers $\mu^{(1)}$ and $\mu^{(2)}$.
- Assign each point, $x^{(i)}$, to closest cluster

$$C^{(i)} = \underset{j=1,\ldots,k}{\operatorname{argmin}} \|\mu^{(j)} - x^{(i)}\|^2 \text{ for } i = 1, \ldots, n$$

How do we find these clusters? (Iterative Approach)



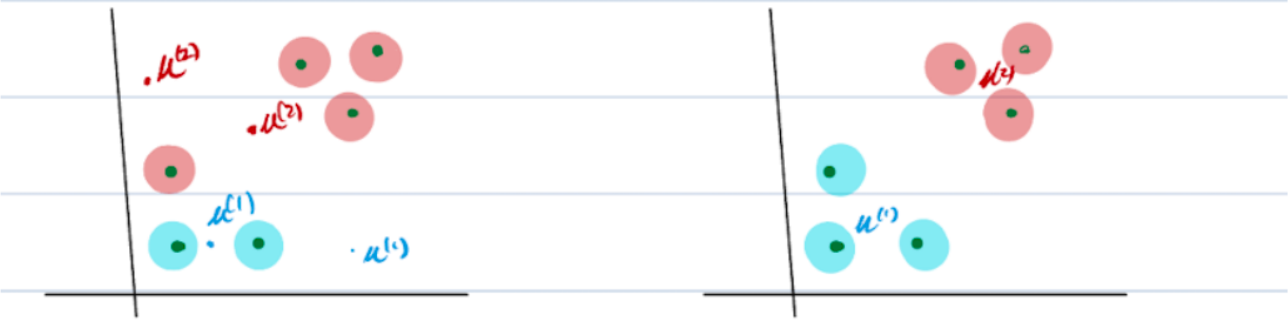
- (Randomly) Initialize Centers $\mu^{(1)}$ and $\mu^{(2)}$.
- Assign each point, $x^{(i)}$, to closest cluster

$$C^{(i)} = \underset{j=1,\dots,k}{\operatorname{argmin}} \|\mu^{(j)} - x^{(i)}\|^2 \text{ for } i = 1, \dots, n$$

- Compute new center of each cluster:

$$\mu^{(j)} = \frac{1}{|\Omega_j|} \sum_{i \in \Omega_j} x^{(i)} \text{ where } \Omega_j = \{i : C^{(i)} = j\}$$

How do we find these clusters? (Iterative Approach)



- (Randomly) Initialize Centers $\mu^{(1)}$ and $\mu^{(2)}$.
- Assign each point, $x^{(i)}$, to closest cluster

$$C^{(i)} = \operatorname*{argmin}_{j=1,\ldots,k} \|\mu^{(j)} - x^{(i)}\|^2 \text{ for } i = 1, \ldots, n$$

- Compute new center of each cluster:

$$\mu^{(j)} = \frac{1}{|\Omega_j|} \sum_{i \in \Omega_j} x^{(i)} \text{ where } \Omega_j = \{i : C^{(i)} = j\}$$

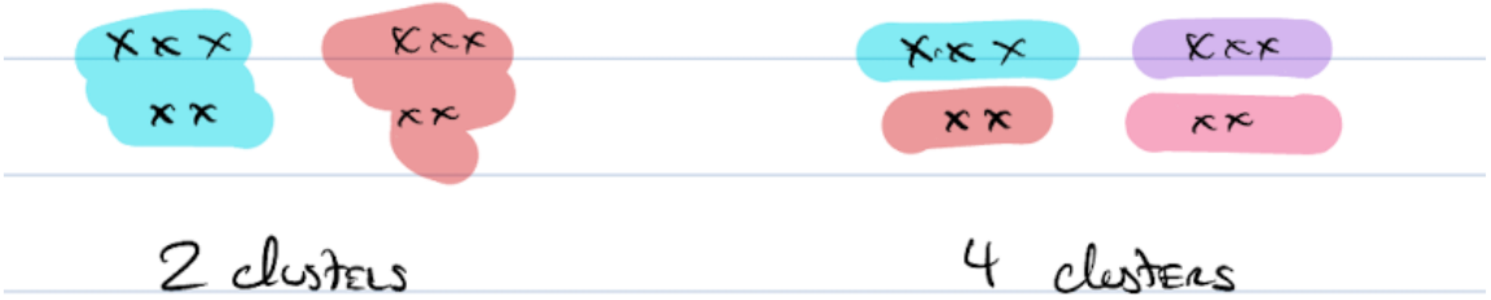
Repeat until clusters stay the same!

Comments about k-means

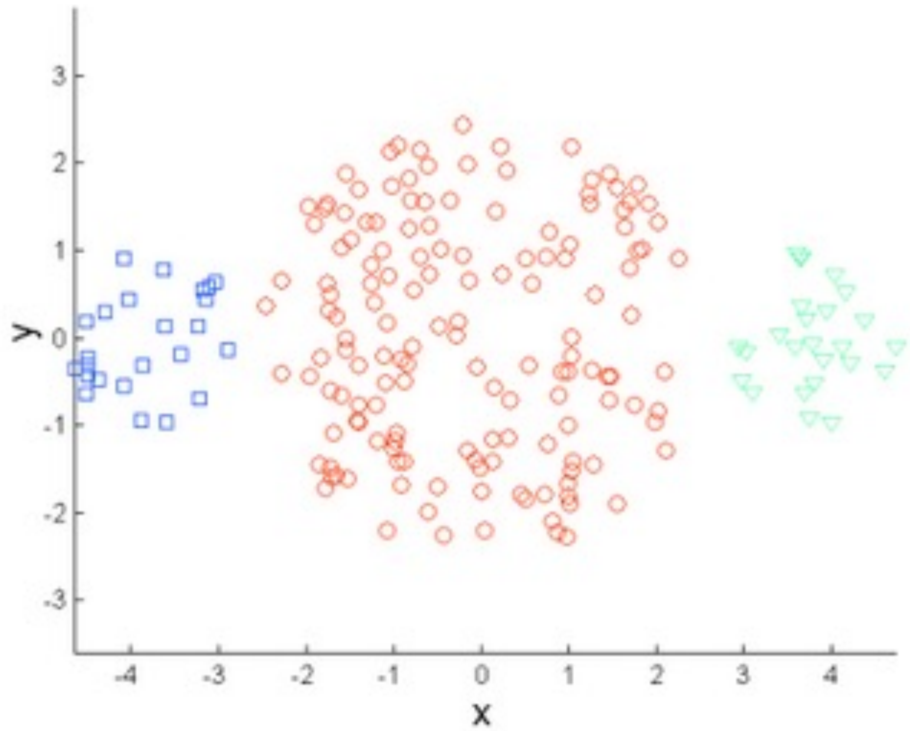
- Does it terminate? Yes, see notes! It minimizes

$$J(C,\mu) = \sum_{i=1}^{n} \|x^{(i)} - \mu^{C^{(i)}}\|^2 \text{ decreases monotonically.}$$

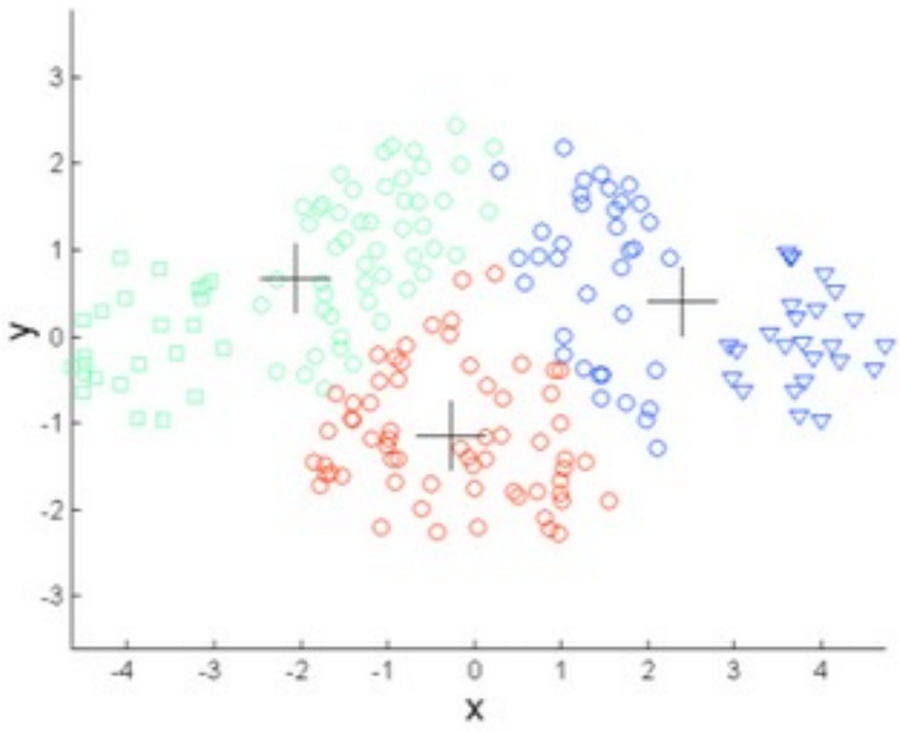
- Does it find a *global minimum*? No, it's an NP-Hard problem!
- Side Note: k-means ++ from great Stanford folks[1]
  - Improved Approximation Ratio and default in SKLearn!
- How do you choose k? *It's a modeling question!*



Different number of clusters

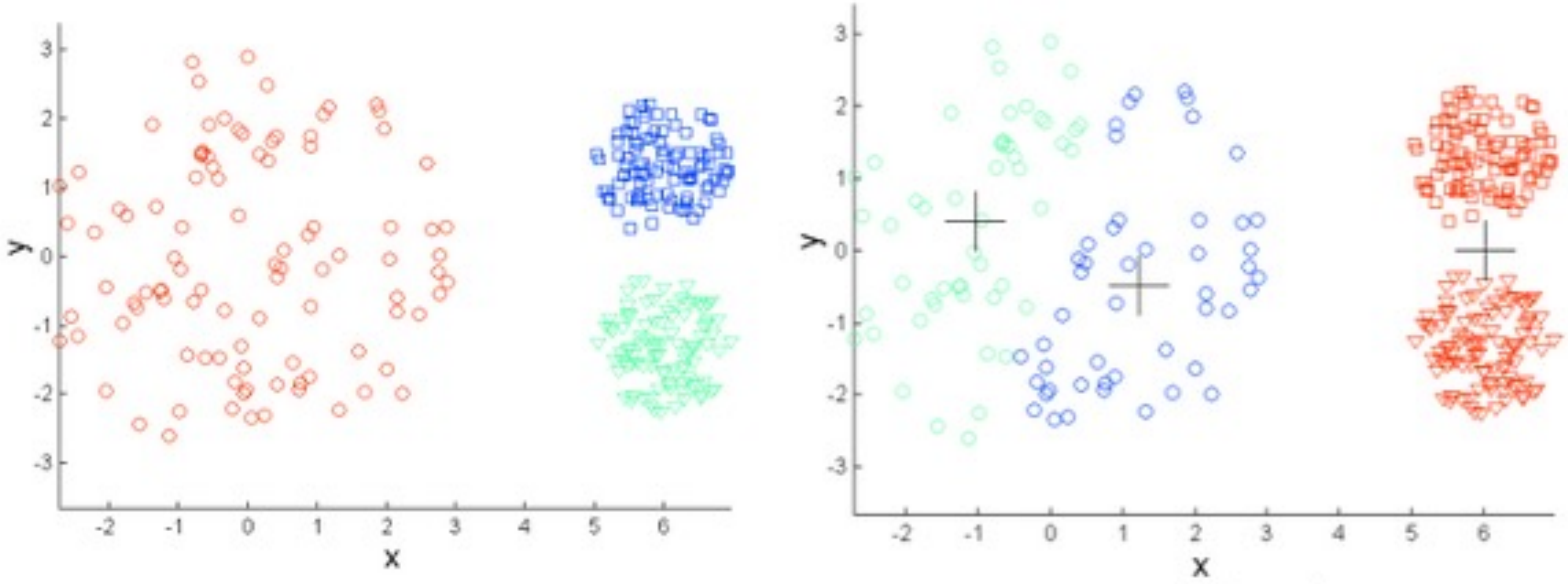


Original Points



K-means (k = 3)

Different Densities



Original Points

K-means (k = 3)

Choosing K?

- # of clusters
- Cluster centers
  - K-means++
- Sensitivity to outliers
  - identify and handle outliers before applying k-means clustering
  - removing them, transforming them, or using a robust variant of k-means clustering that is less sensitive to the presence of outliers

K-means++

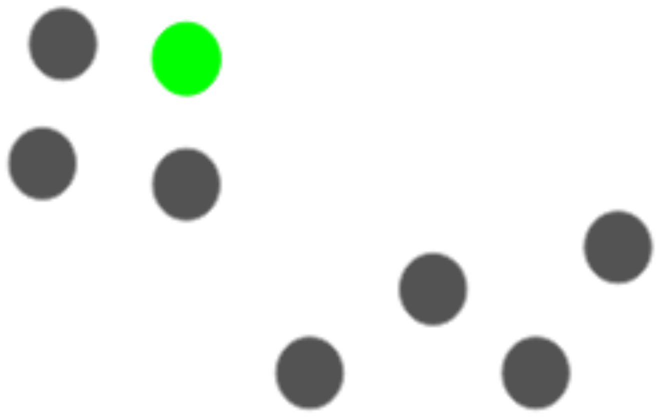
- Compute Density Estimation
- Assign centroids based on that

K-means++

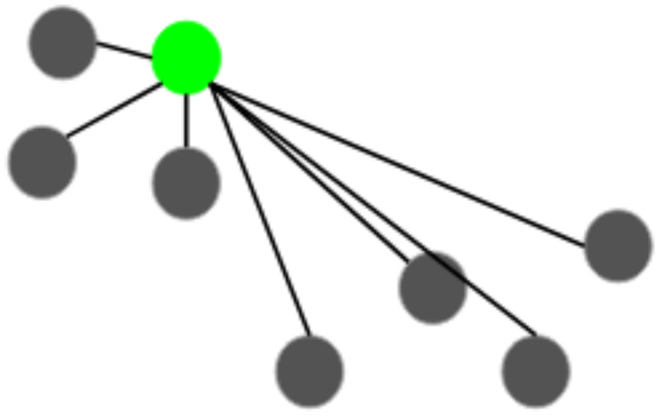
- Compute Density Estimation
- Assign centroids based on that

K-means++

- Compute Density Estimation
- Assign centroids based on that
- 3 clusters



Random Pick



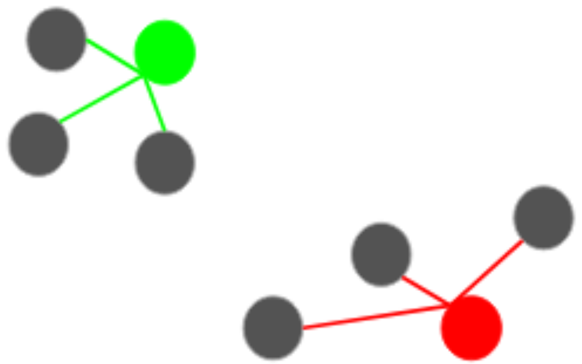
Calculate D(x)

`

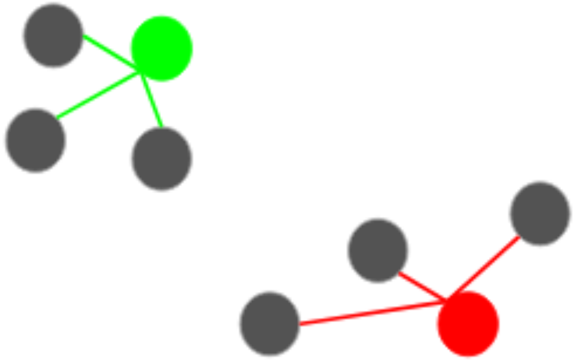
Largest $D(x)^2$

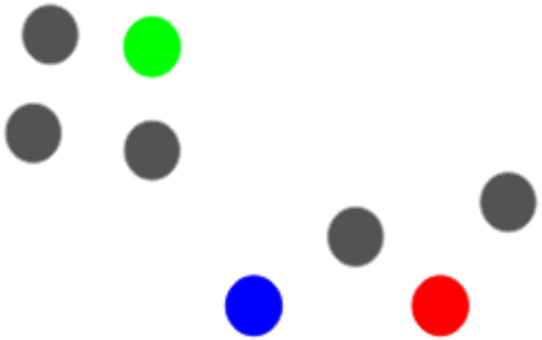
`

Largest $D(x)^2$



Largest $D(x)^2$



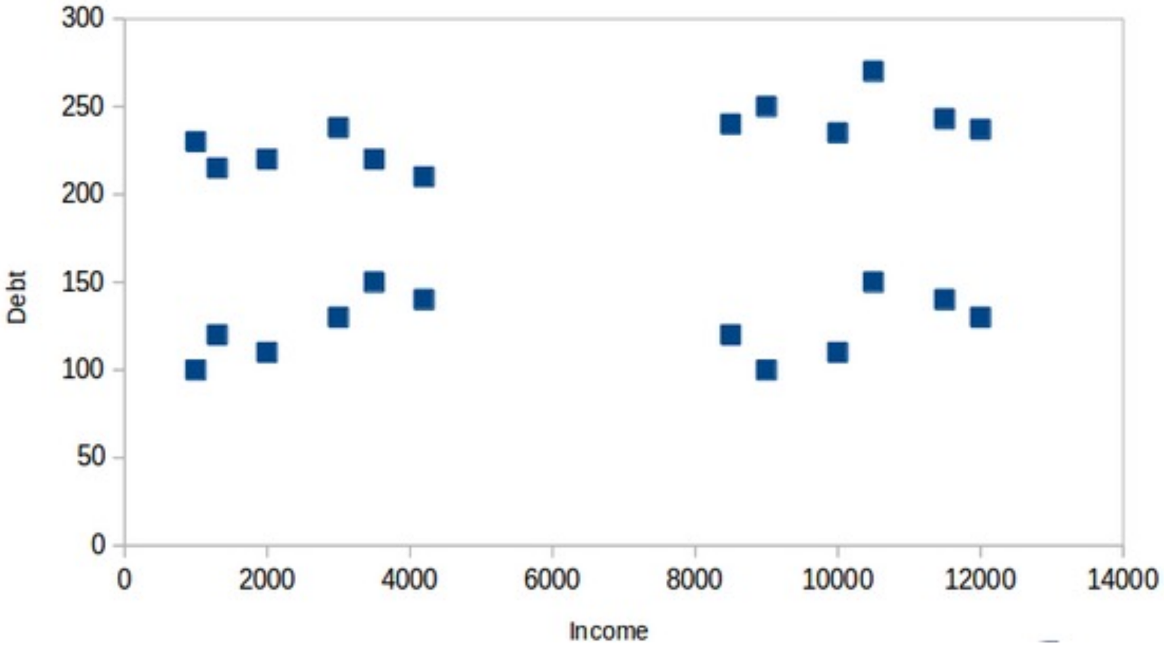


Largest $D(x)^2$
from both center

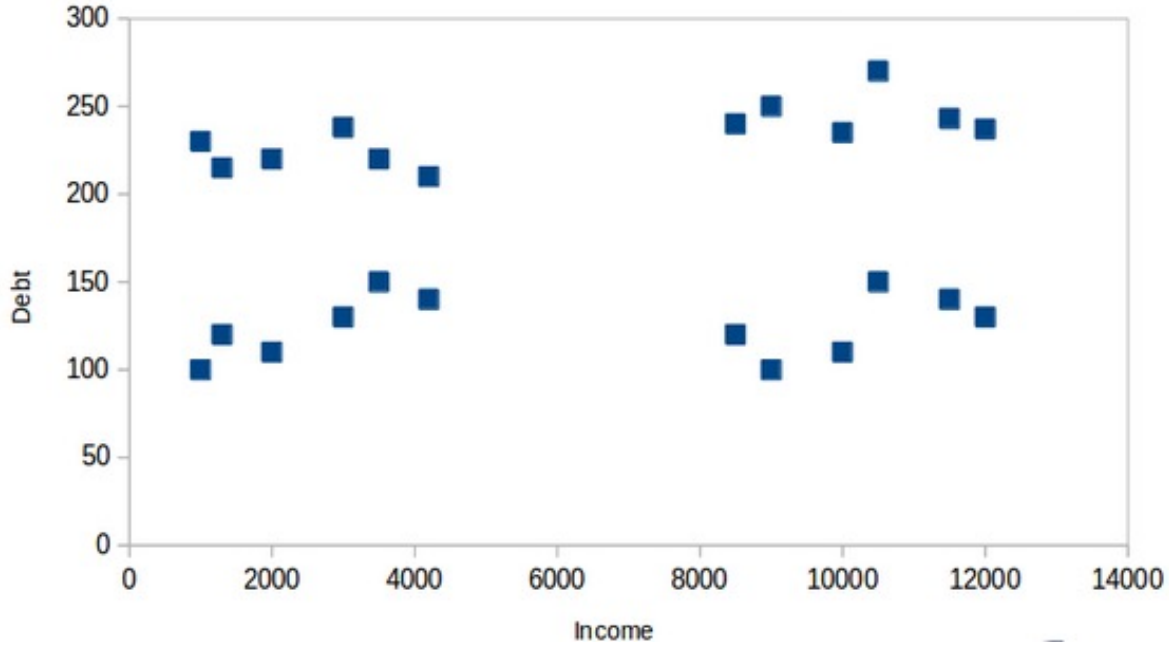
- Steps to Initialize the Centroids Using K-Means++

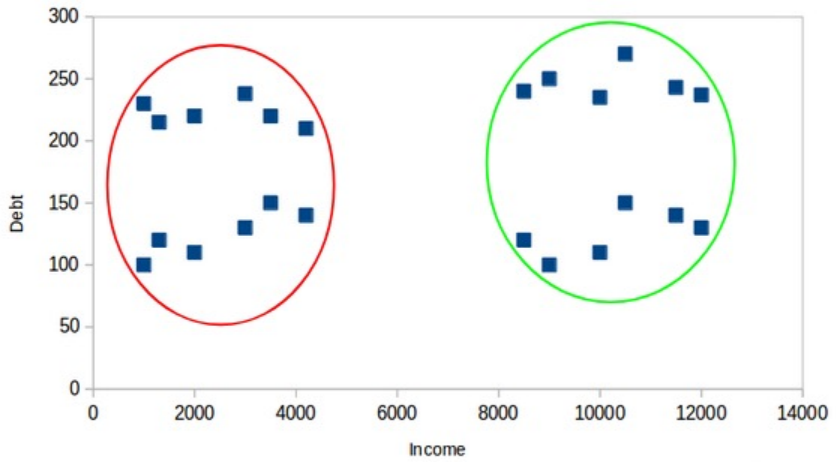
1. The first cluster is chosen uniformly at random from the data points we want to cluster. This is similar to what we do in K-Means, but instead of randomly picking all the centroids, we just pick one centroid here
2. Next, we compute the distance (D(x)) of each data point (x) from the cluster center that has already been chosen
3. Then, choose the new cluster center from the data points with the probability of x being proportional to $(D(x))^2$
4. We then repeat steps 2 and 3 until *k* clusters have been chosen

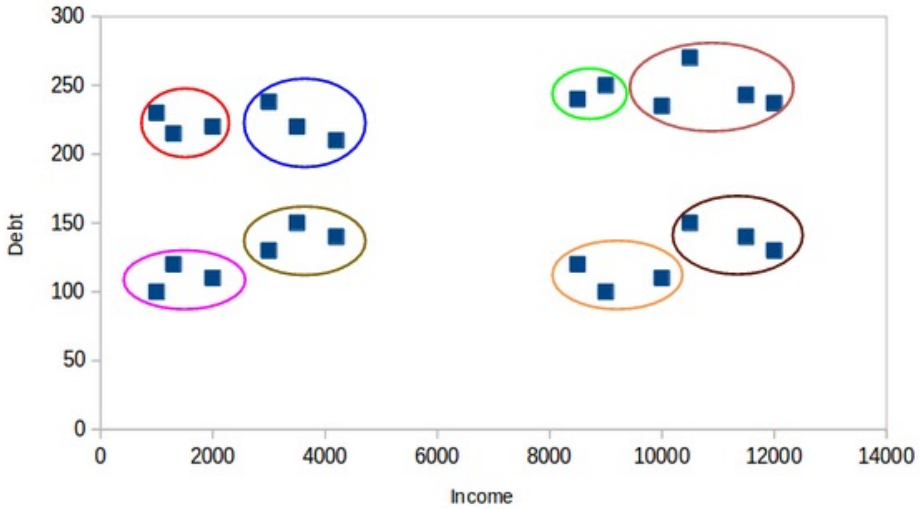
How to Choose the Right Number of Clusters?



How to Choose the Right Number of Clusters?





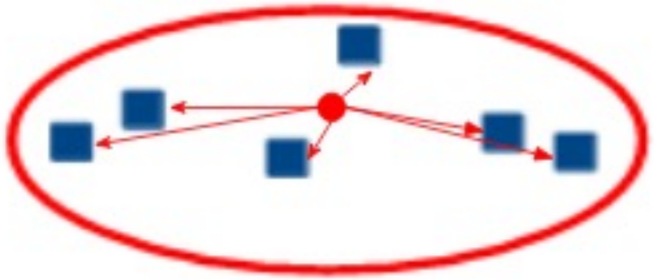


Evaluation Metrics

- Inertia
  - sum of distances of all the points within a cluster from the centroid of that cluster.
  - lesser the inertia value, the better our clusters are.
- Silhouette Score
  - high silhouette score = clusters are well separated
  - 0 = overlapping clusters,
  - negative score suggests poor clustering solutions.
  - For each data,

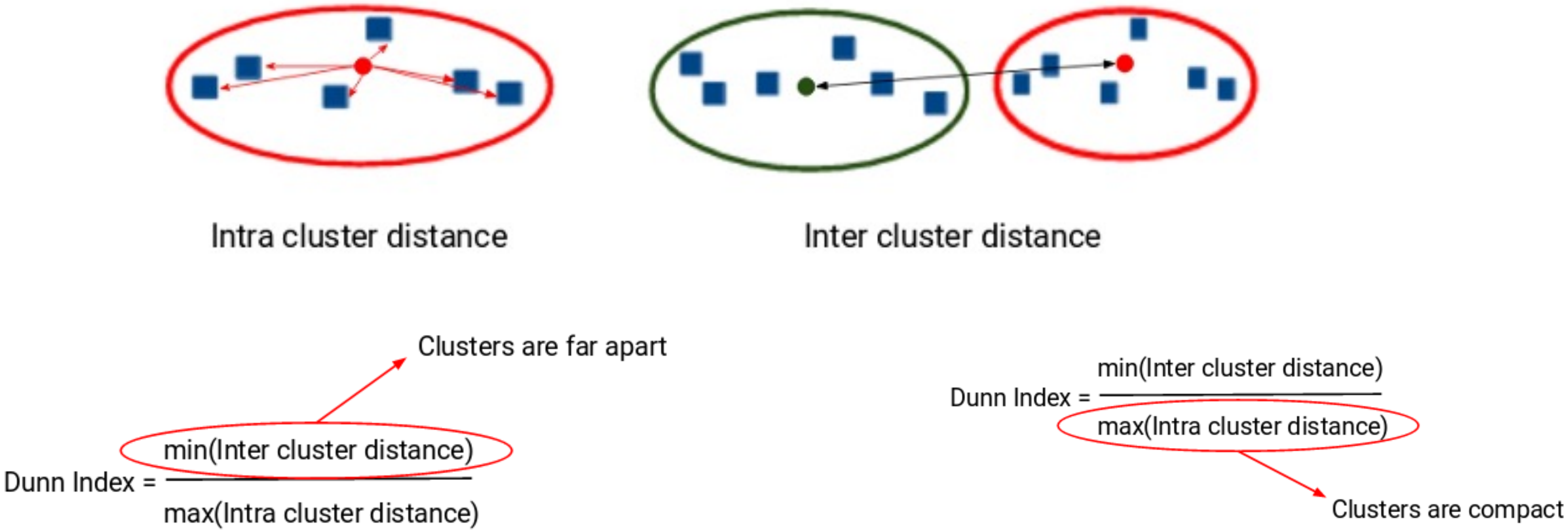
$$s = (b - a) / \max(a, b)$$

    - 'a' is the average distance within the cluster, 'b' is the average distance to the nearest cluster, and 'max(a, b)' is the maximum of 'a' and ‘b’
  - Mean for all points

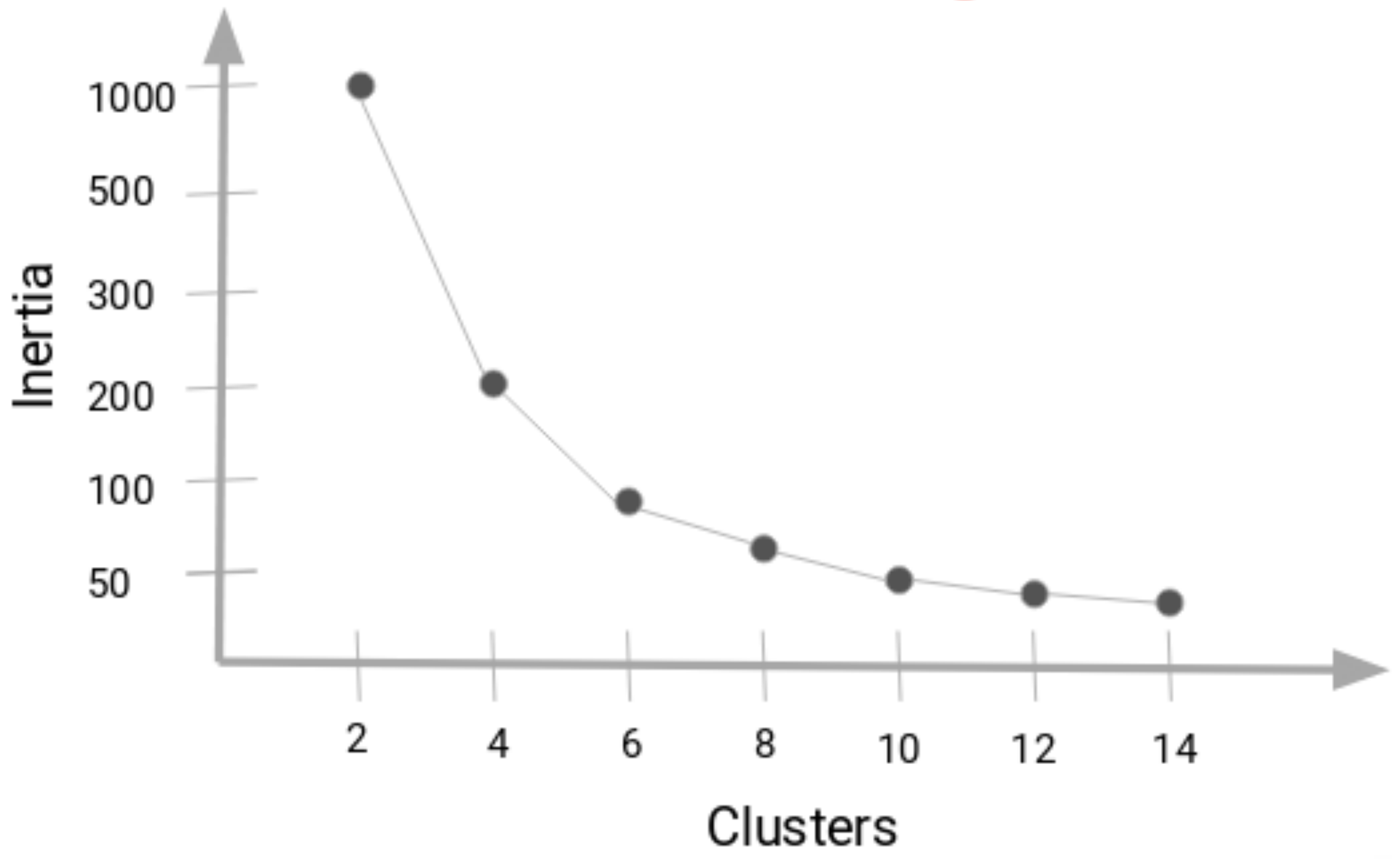


Intra cluster distance

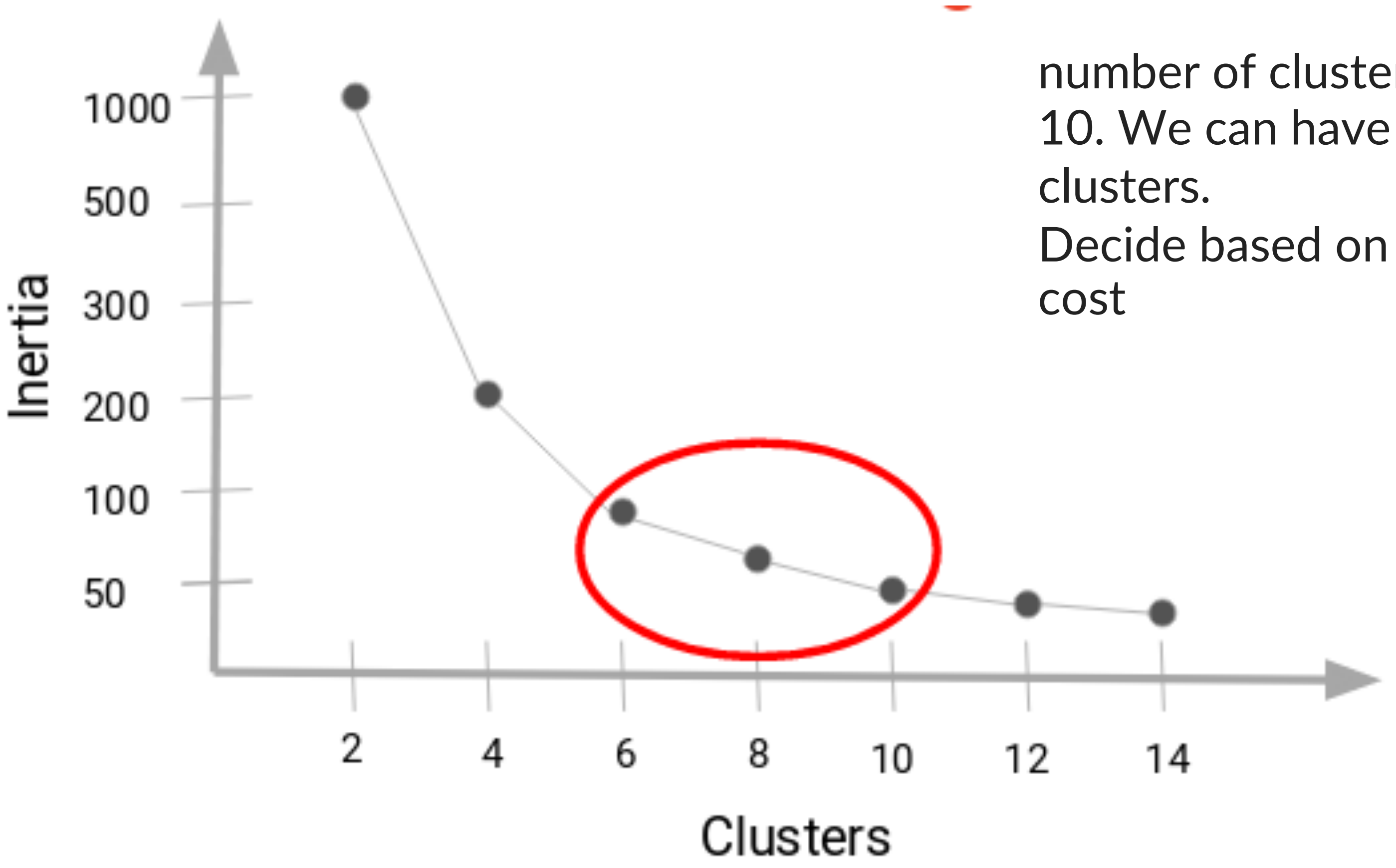
- Dunn index


Intra cluster distance
Inter cluster distance
Clusters are far apart
min(Inter cluster distance)
Dunn Index =
max(Intra cluster distance)
min(Inter cluster distance)
Dunn Index =
max(Intra cluster distance)
Clusters are compact

Empirical Choice of K



Empirical Choice of K



number of clusters between 6 and 10. We can have 7, 8, or even 9 clusters.
Decide based on computational cost