K-Means Clustering

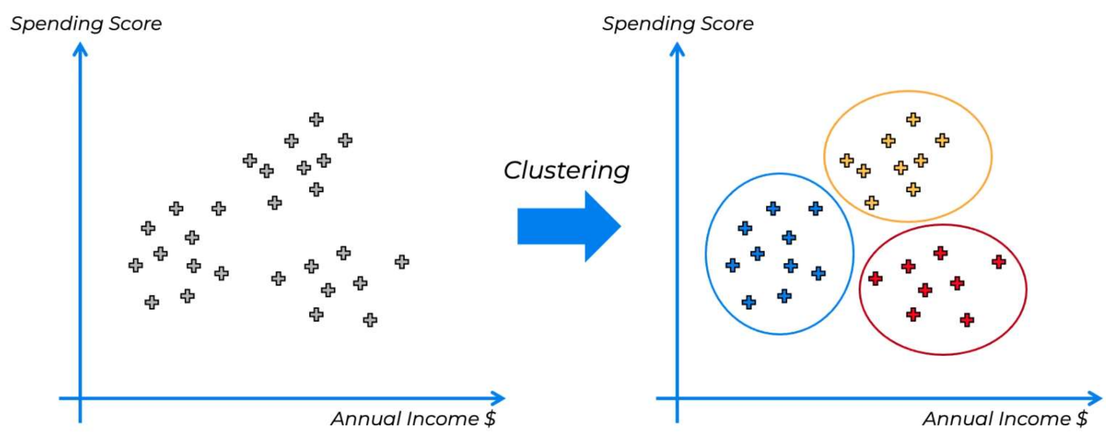
Friday, December 13, 2024 10:29 AM

Unsupervised Machine Learning

Clustering
- Grouping unlabeled data

Supervised Learning
- You have training data and answers to that data

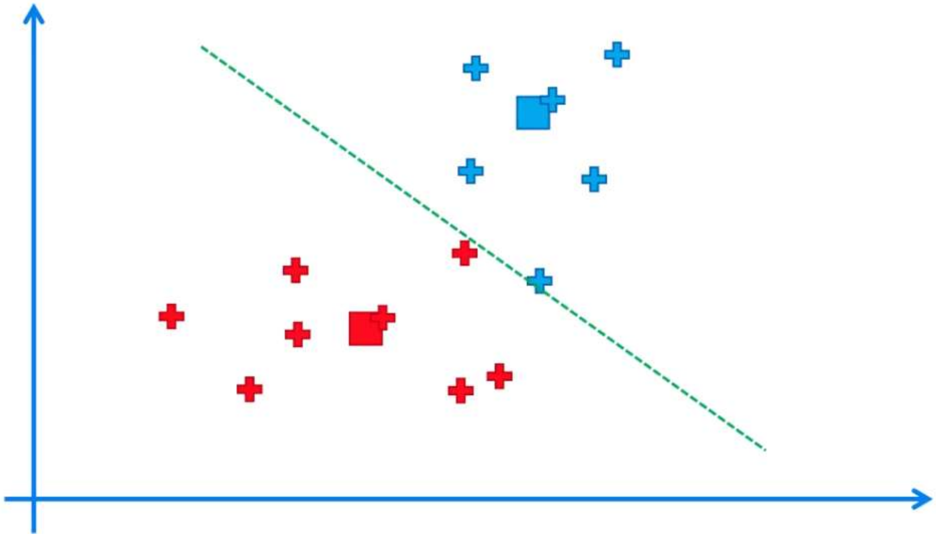
Unsupervised Learning
- We don't have answers and the model needs to figure it out itself
- The model can still group fruits together, but they don't know the fruit category



- You can dig deeper to learn more about those groups

K-Means Clustering
- Given a scatter plot, we want to make clusters

- You want to decide how many clusters you want
- For each cluster you can place a random point on the graph
- Find the equadistant line between each randomly placed points
- You need to calcuate the central mass or gravity of each point and find the average
  - Move the randomly places points to those positions
- Now find the equadistant line and do the same things with finding the average
  - Do this until doing the process again doesn't change anything



The Elbow Method

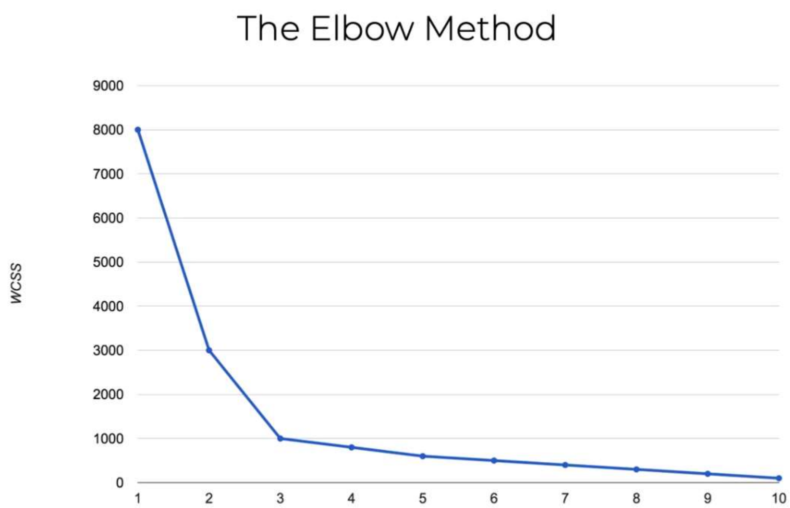
- K-means clustering doesn't need to always be in 2 dimensions, it can work in multiple dimensions

- How to decide how many clusters you want

  Within Cluster Sum of Squares:

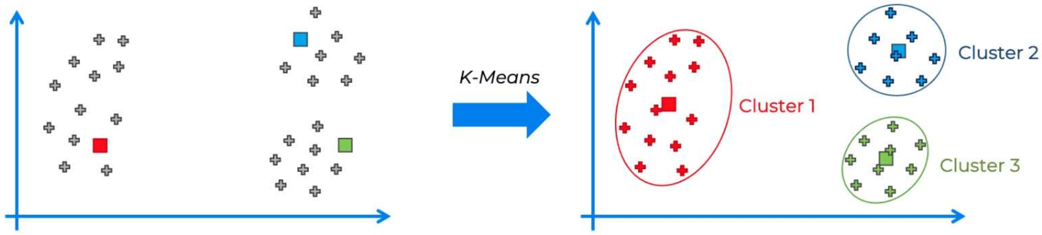
  - $$\mathrm{WCSS} = \sum_{\mathrm{P_i\ in\ Cluster\ 1}} \mathrm{distance}(\mathrm{P_i}, \mathrm{C_1})^2 + \sum_{\mathrm{P_i\ in\ Cluster\ 2}} \mathrm{distance}(\mathrm{P_i}, \mathrm{C_2})^2 + \ldots$$

  - The more clusters we have the smaller WCSS

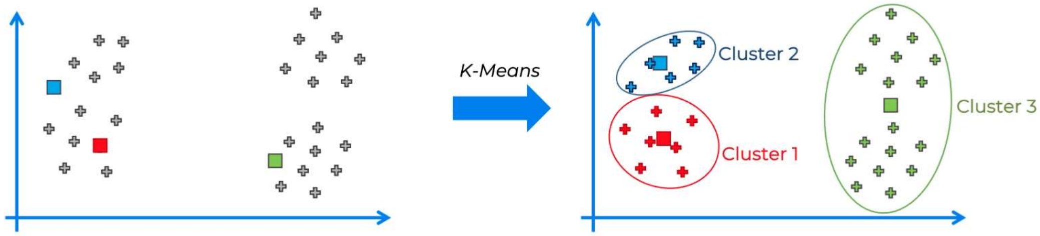


- Look for where is the kink in this chart
- 3 is the optimal number of clusters

K-Means++



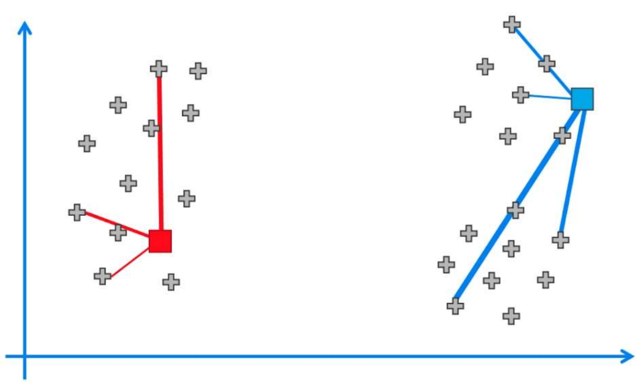
-



- The bottom graph is not good. Two of the random points are placed in the first cluster, making them into two separate clusters
- Results are different because the initialization of the random points are different

K-Means++ Initialization Algoriithm

- Step 1: Chose first centroid at random among points
- Step 2: For each of the remaining data points compute the distance (D) to the <u>nearest</u> out of already selected centroids
- Step 3: Choose next centroid among remaining data points using <u>weighted</u> random selection -- weighted by D^2
- Step 4: Repeat Steps 2 and 3 until all k centroids have been selected
- Step 5: Proceed with standard k-means clustering

- What this will do:
  - After placing a centroid, find all data points and find the distance of each. When you square the distance, which ever has the highest distance you place another centroid
  - 
- This initialization does not guarantee there won't be an issue
  - This is because it is still random, but it's an weighted random to the probability of working is much higher

Hierarchical Clustering

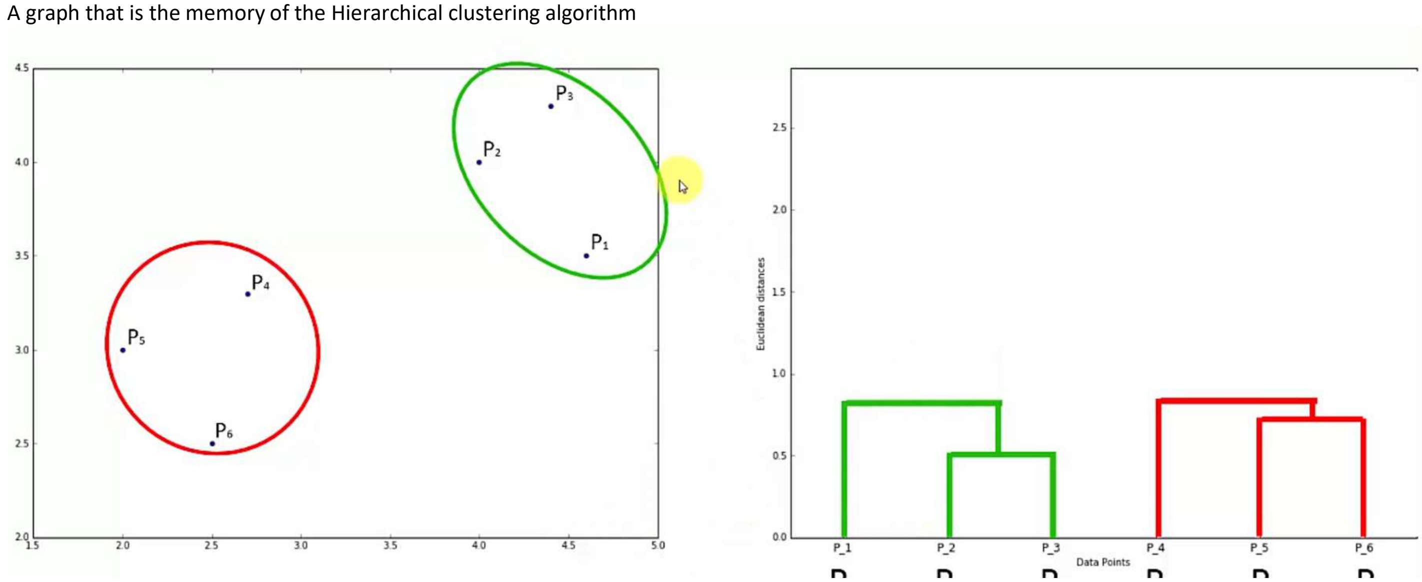
Friday, December 13, 2024 12:36 PM

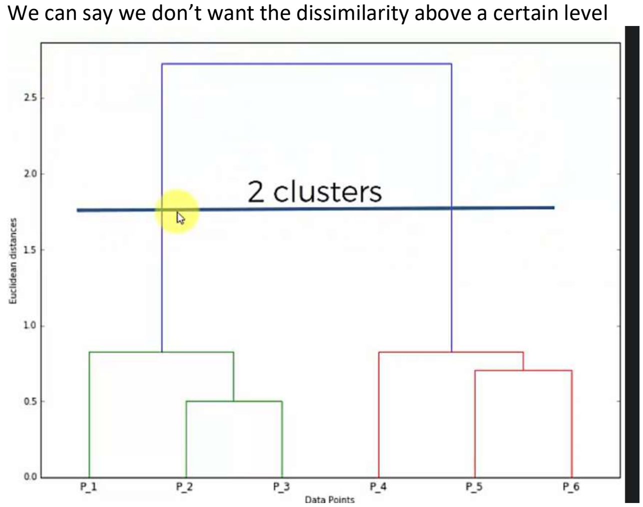
Can result the same as K-Means but a different process

Agglomerative and Divisive Approaches

Agglomerative
- STEP 1: Make each data point a single-point cluster
    - This forms N clusters
- STEP 2: Take the two closest data points and make them one cluster
    - That forms N-1 clusters
- STEP 3: Take the two closest clusters and make them one cluster
    - Forms N -2 clusters
- STEP 4: Repeat STEP 3 until there is only one cluster

Distance Between Two Clusters:
- Option 1: Closest points
- Option 2: Furthest points
- Option 3: Average Distance
- Option 4: Distance between centroids

Dendrograms
- A graph that is the memory of the Hierarchical clustering algorithm
- 

Using Dendrograms
- Look at the vertical levels and create thresholds
- We can say we don't want the dissimilarity above a certain level
- 

- We can tell how many clusters we have by seeing how many vertical lines the threshold crosses

How to find the optimal number of clusters

- Find the highest vertical distance, and add the threshold through that line