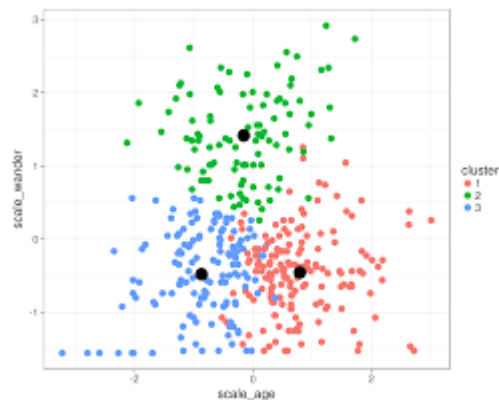


Supervised Learning Vs. Unsupervised Learning

- **Supervised Learning**
 - Use training set: points which are classified prior to passing the point through the network
 - Example: Circle Lab
- **Unsupervised Learning**
 - We don't set what the points should be classified as ahead of time
 - Find the mean of groups of points and use the mean to classify data set



There are 3 Means

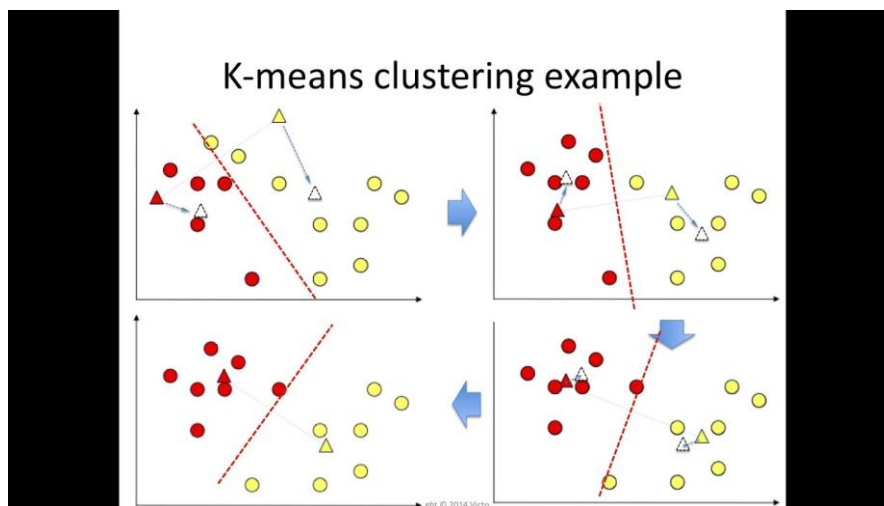
K-means: algorithm to determine the K means which minimize the 'error'

1. Find K starting 'means (values)' from S (set of points we'd like to classify).

For learning K-means, pick your starting 'means' at random from your points set

2. For each point in S, decide which of the K means it is closest to, and put it in that group.
3. Figure out the means of all the K groups and these become the new means.

Loop (step 2 and 3) until no point is group hopping.



Example:

K-Means is widely used for many applications: Image Segmentation, Clustering Gene Segmentation Data, News Article Clustering, Clustering Languages, Species Clustering, Anomaly Detection

Lab assignment: With a given K and a picture by URL or filename, use the given K and cluster the image with the K-means algorithm, and then re-draw the picture with K number of colors.

Input Samples: `kmeans.py 6 'http://www.blahblah.com/image.jpg'`

`kmeans.py 4 'image.jpg'`

Sample URL: `'https://cf.geekdo-images.com/imagepage/img/5lvEWGGTqWDFmJq_MaZvVD3sPuM=/fit-in/900x600/filters:no_upscale()/pic260745.jpg'`

1. Start with the number of pixels belongs to each mean:
[73766, 62655, 51544, 112035]

2. Trace the number of pixels group hopping:

diff 1: [7498, -9510, 1768, 244]

...

diff 22: [-951, 0, 871, 80]

...

diff 40: [0, 0, 0, 0] # No point is group hopping! Done!

3. Inspect image and for every pixel, and if it's surrounded by pixels all of one color, make current pixel that color.