Welcome to General Assembly

- **WiFi** GA Guest
- **Password** yellowpencil

GENERAL ASSEMBLY

DATA SCIENCE

DAT11SYD

Lesson-08: Clustering

Course Plan

UNITS

UNIT 1: FOUNDATIONS OF DATA MODELING	Introduction to Data Science	Lesson 1
	Elements of Data Science	Lesson 2
	Data Visualisation	Lesson 3
	Linear Regression	Lesson 4
	Logistic Regression	Lesson 5
	Model Evaluation	Lesson 6
	Regularisation	Lesson 7
	Clustering	Lesson 8
UNIT 2: DATA SCIENCE IN THE REAL WORLD	Recommendations	Lesson 9
	SQL + Productivity	Lesson 10
	Decision Trees	Lesson 11
	Ensembles	Lesson 12
	Natural Language Programming	Lesson 13
	Cloud Computing	Lesson 14
	Time Series	Lesson 15
	Soft Skills	Lesson 16
	Network Analysis	Lesson 17
	Neural Networks	Lesson 18
	Final Projects Presentations	Lesson 19
	Final Projects Presentations	Lesson 20

Paul & James review final project ideas

Git & GitHub – 1 Pager Guide!

(Part B) EVERY CLASS:

At the START of the class, you'll need to sync the latest materials from the COURSE repo:

(1) **Make sure you are in the dat11syd directory:**

```
cd ~/workspace/dat11syd
```

(2) **Make sure to select the "master" branch of your repo:**

```
git checkout master
```

(3) **Fetch the latest changes from the UPSTREAM repo (i.e the course repo)**

```
git fetch upstream
```

(4) **Merge the changes from the upstream repo to your master branch:**

```
git merge upstream/master
```

DURING the class:

(5) **Before editing, either copy files to your "students/" folder, or rename them**

At the END of every class:

(6) **Make sure you are in the dat11syd directory:**

```
cd ~/workspace/dat11syd
```

(7) **Add any files that you've updated to your git registry:**

```
git add -A
```

(8) **Commit the changes with a sensible comment:**

```
git commit -m "my updates for lesson 7"
```

(9) **Push your changes to your PERSONAL repo:**

```
git push origin master
```

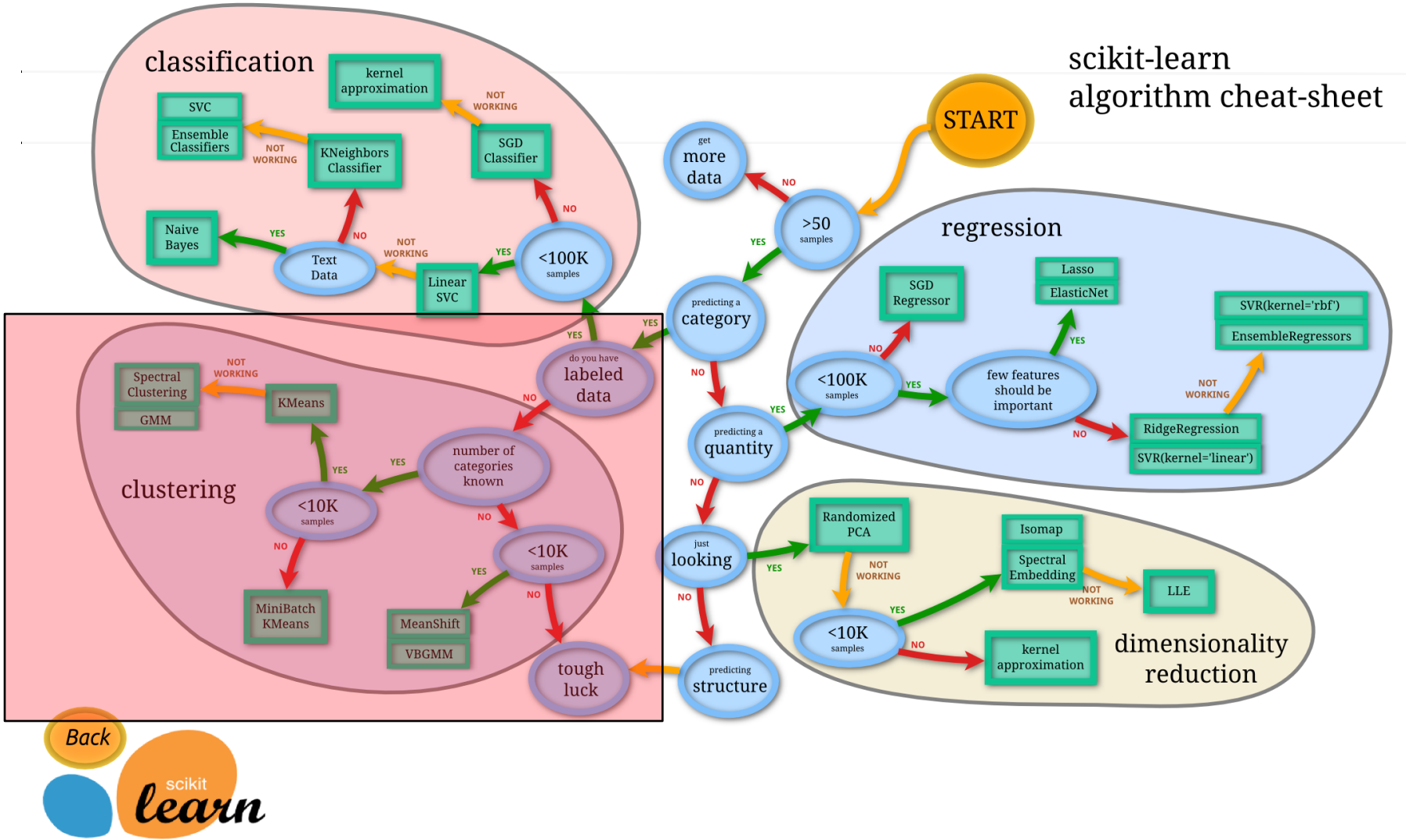
DONE!!!!!

AGENDA

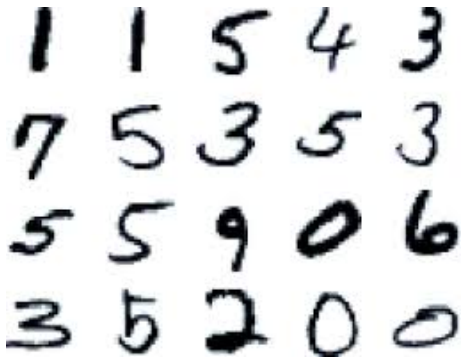
1. Motivation / Review
2. What is Clustering?
3. What is K-Means and how does it work?
4. Lab
5. Discussion

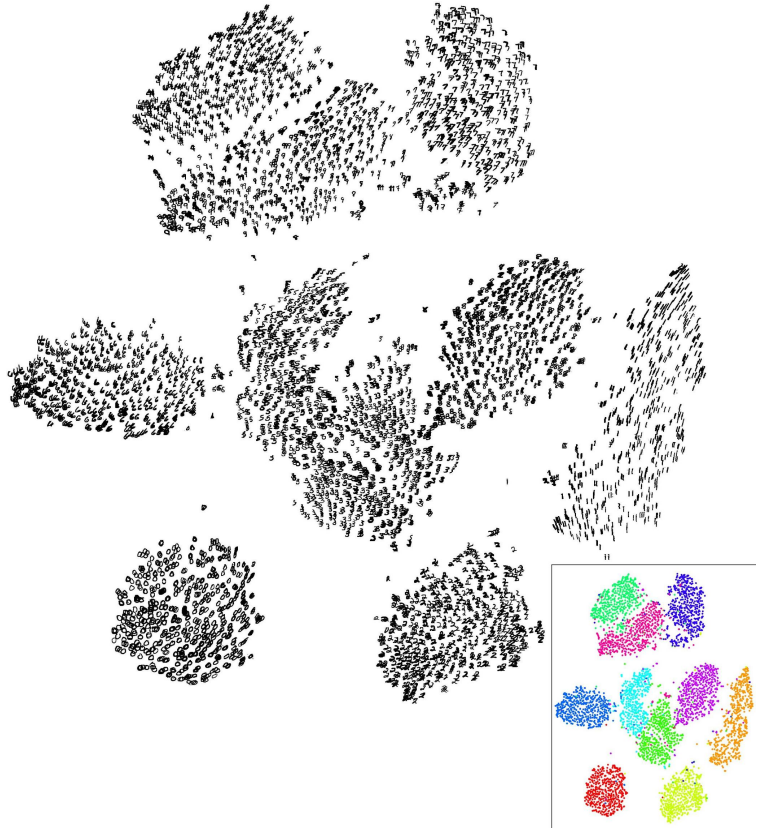
DATA SCIENCE PART TIME COURSE

WHAT IS CLUSTERING AND WHY DO IT?

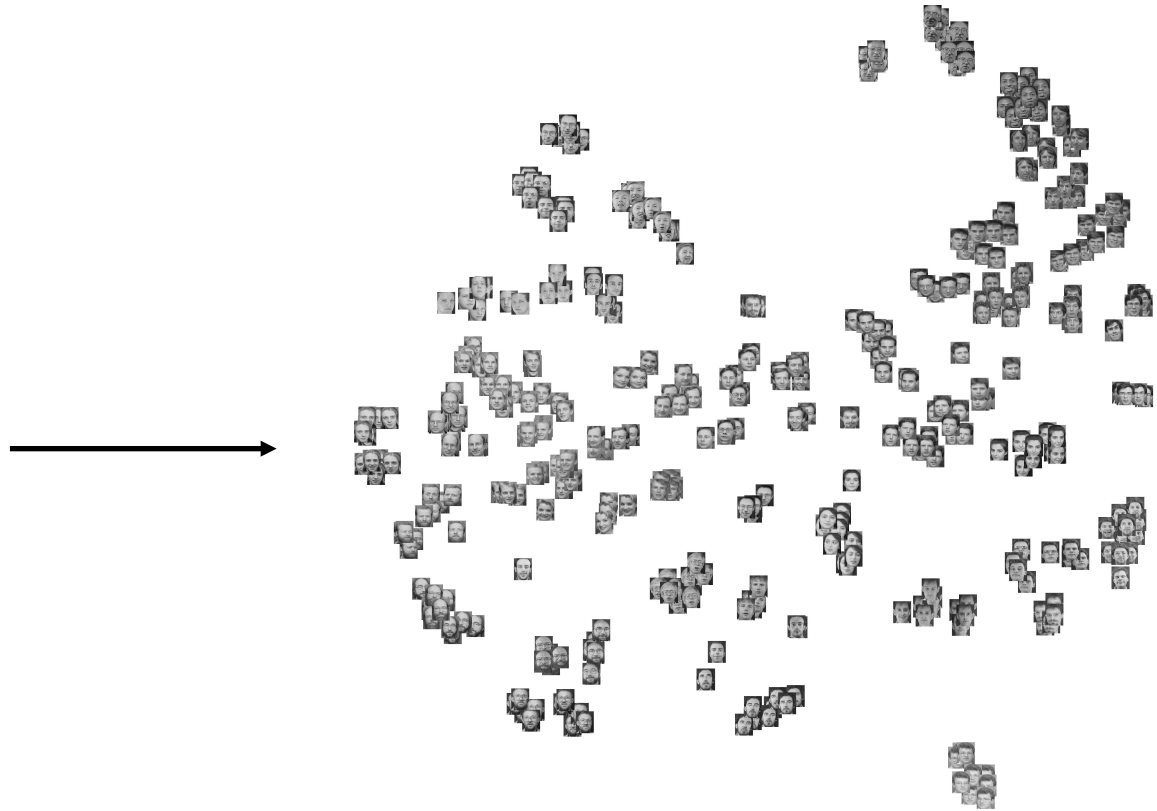

scikit-learn
algorithm cheat-sheet
START
get more data
NO
>50 samples
YES
predicting a category
NO
predicting a quantity
NO
just looking
NO
predicting structure
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
NO
YES
regression
SGD Regressor
NO
Lasso
ElasticNet
<100K samples
YES
few features should be important
YES
NO
NOT WORKING
SVR(kernel='rbf')
EnsembleRegressors
RidgeRegression
SVR(kernel='linear')
clustering
do you have labeled data
YES
NO
Spectral Clustering
GMM
NOT WORKING
KMeans
YES
number of categories known
YES
NO
<10K samples
NO
MiniBatch KMeans
<10K samples
YES
NO
MeanShift
VBGMM
tough luck
dimensionality reduction
Randomized PCA
YES
NOT WORKING
<10K samples
YES
NO
Isomap
Spectral Embedding
NOT WORKING
LLE
kernel approximation
Back
scikit learn

MNIST

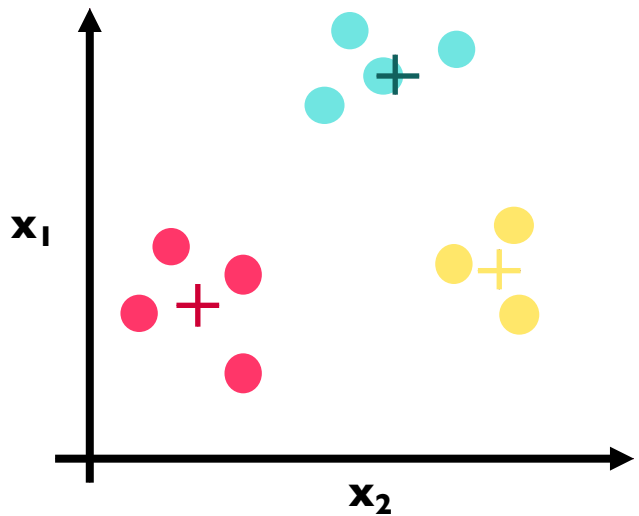




Olivetti Faces



- What is a Cluster?
- Why would we do this?
- What is K-Means?



WHAT IS A CLUSTER?

- Unsupervised learning => **find interesting patterns** or groups in data.
- **No** variable we are trying to predict (a **Y value**).
- Clustering discovers **subgroups** in data where the points are similar to each other.
- All points in the same group are similar.
- Points in different groups are different to each other.
- What variables to make groups on. What makes them different (or similar)?

WHY WOULD WE CLUSTER DATA?

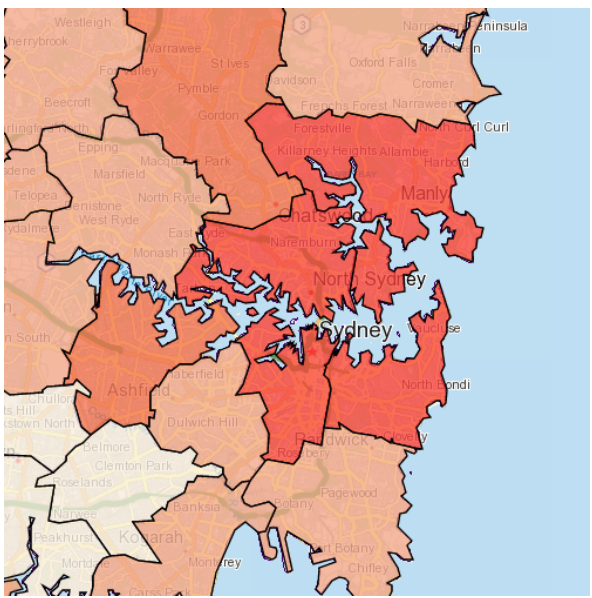
*To enhance understanding of a data by **dividing into groups** (behavioural customer segmentation).*

*Clustering provides a **layer of abstraction** from individual data points (cluster number).*

*The goal is to extract and **enhance the natural structure** of the data*

WHY WOULD WE CLUSTER DATA? 13

Marketing teams might want to group customers into like groups as a way of summarising the data

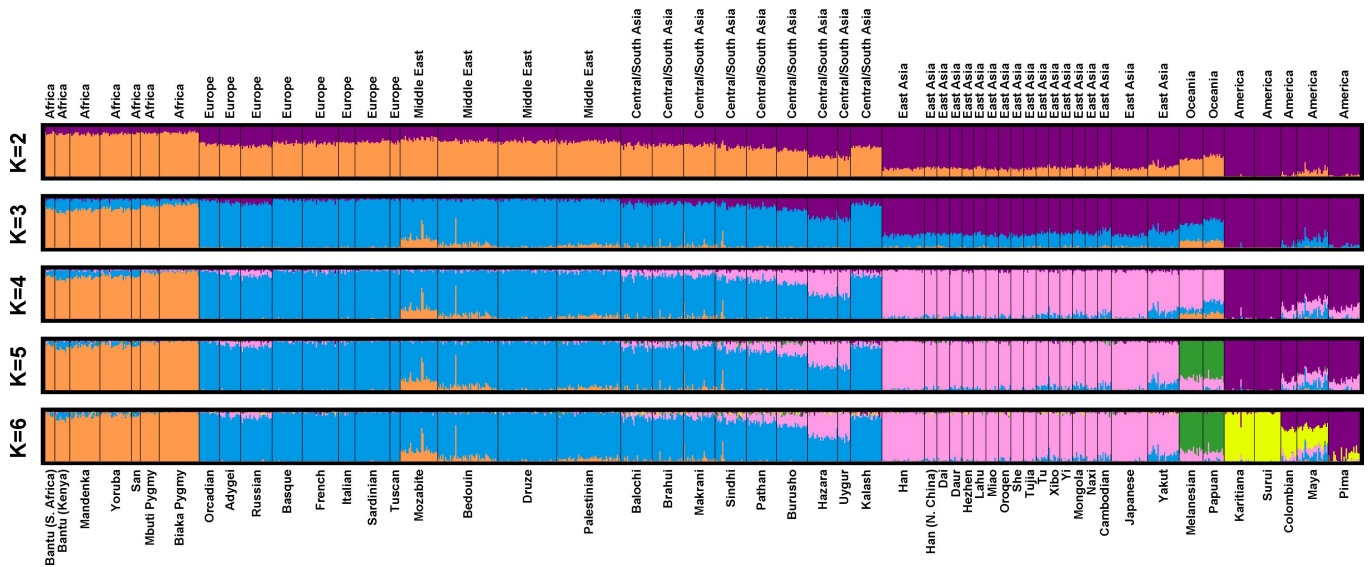


WHY WOULD WE CLUSTER DATA?

Financial groups may want to group transactions into like groups as a way to find unusual payments

WHY WOULD WE CLUSTER DATA?

Genetics data can be clustered to identify ancestry



DATA SCIENCE PART TIME COURSE

HOW DO WE CLUSTER DATA?

1) Choose k initial centroids (note that k is an input)

2) For each point:

- find distance to each centroid

- assign point to nearest centroid

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met

There are several options:

- randomly (but may yield divergent behaviour)

- perform alternative clustering task, use resulting centroids as initial k-means centroids **(warm start)**

- start with global centroid, choose point at max distance, repeat (but might select outlier)

The similarity criterion is determined by the measure we choose.

In the case of k-means clustering, the similarity metric is the **Euclidian distance:**

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{N} (x_{1i} - x_{2i})^2}$$

Whiteboard distantce btwn 2 pts example

STEP 3 - RECALCULATE CENTROID POSITIONS

Q: How do we re-compute the positions of the centres at each iteration of the algorithm?

Q: How do we re-compute the positions of the centres at each iteration of the algorithm?

A: By calculating the centroid (i.e., the geometric centre)

We iterate until some stopping criteria are met; in general, suitable convergence is achieved in a small number of steps.

Stopping criteria can be based on the centroids (eg, if positions change by no more than ε) or on the points (eg, if no more than x% change clusters between iterations).

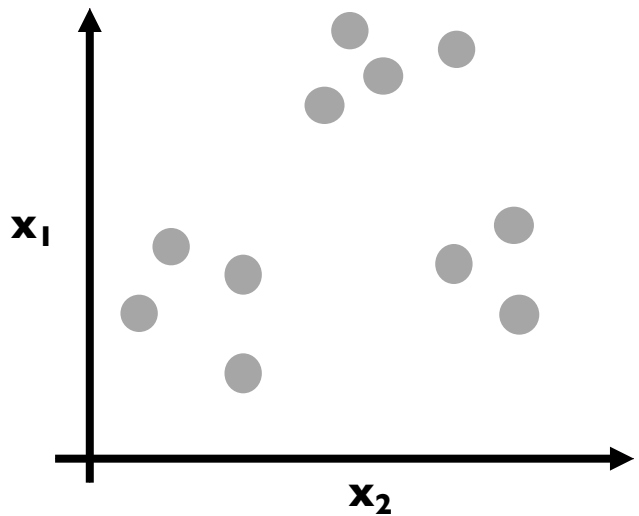
1) Choose k initial centroids

2) For each point:

- *find distance to each centroid*
- *assign point to nearest centroid*

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



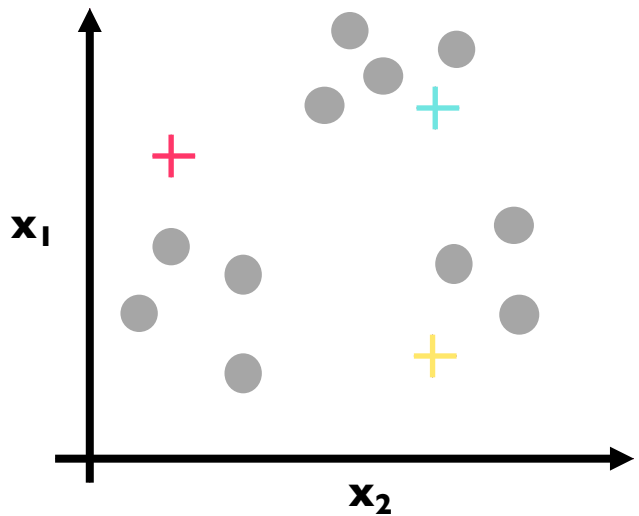
1) Choose k initial centroids

2) For each point:

- *find distance to each centroid*
- *assign point to nearest centroid*

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



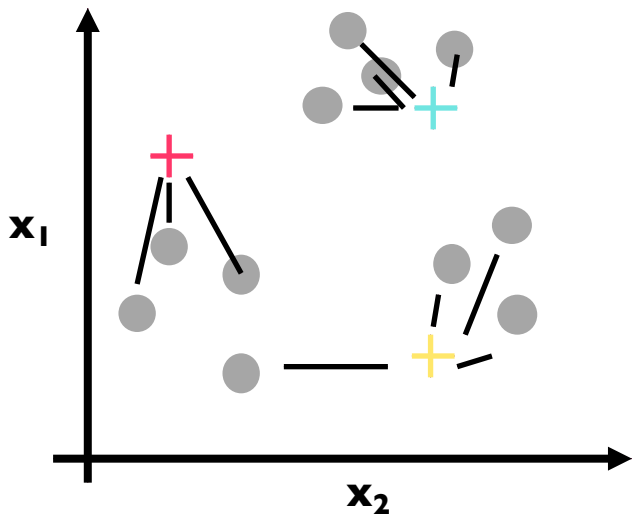
1) Choose k initial centroids

2) For each point:

- *find distance to each centroid*
- *assign point to nearest centroid*

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



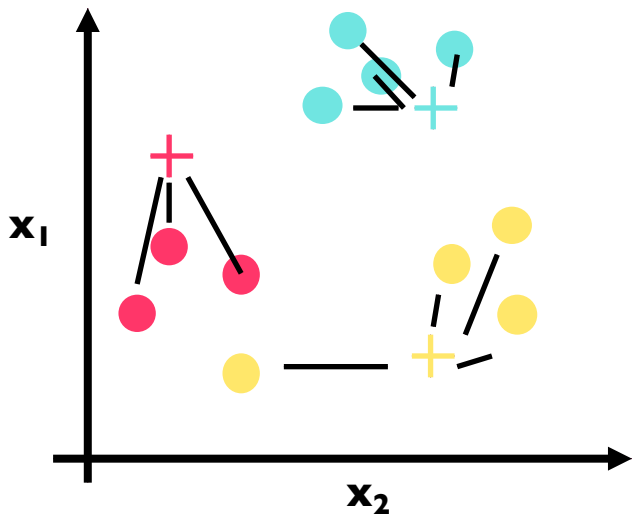
1) Choose k initial centroids

2) For each point:

- *find distance to each centroid*
- *assign point to nearest centroid*

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



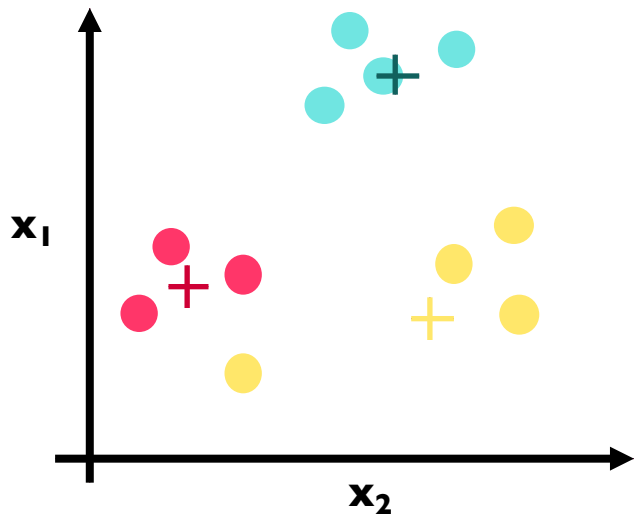
1) Choose k initial centroids

2) For each point:

- *find distance to each centroid*
- *assign point to nearest centroid*

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



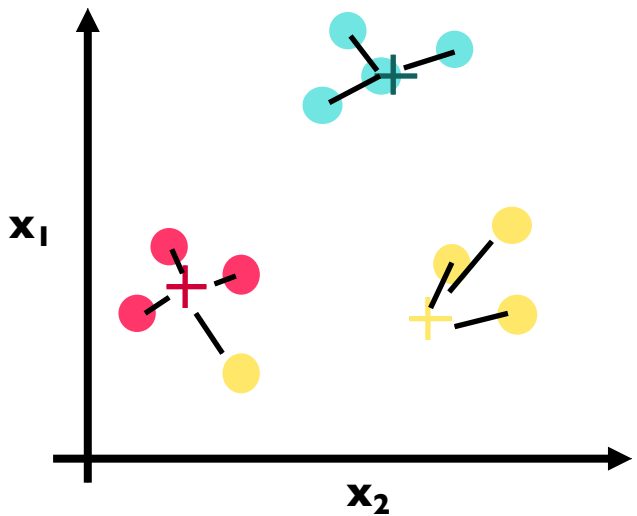
1) Choose k initial centroids

2) For each point:

- *find distance to each centroid*
- *assign point to nearest centroid*

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



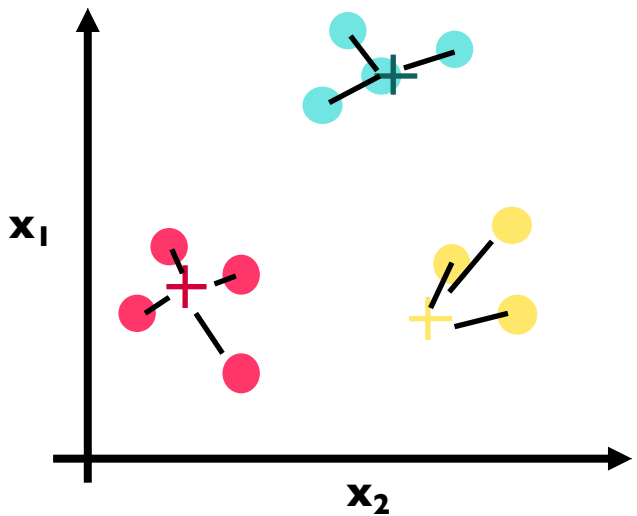
1) Choose k initial centroids

2) For each point:

- *find distance to each centroid*
- *assign point to nearest centroid*

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



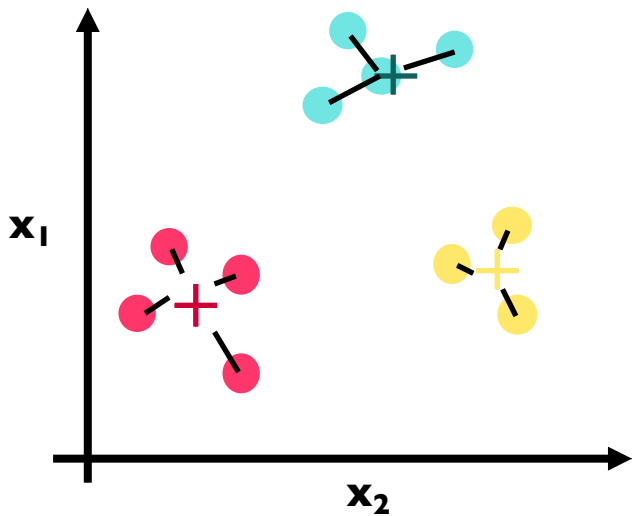
1) Choose k initial centroids

2) For each point:

- *find distance to each centroid*
- *assign point to nearest centroid*

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



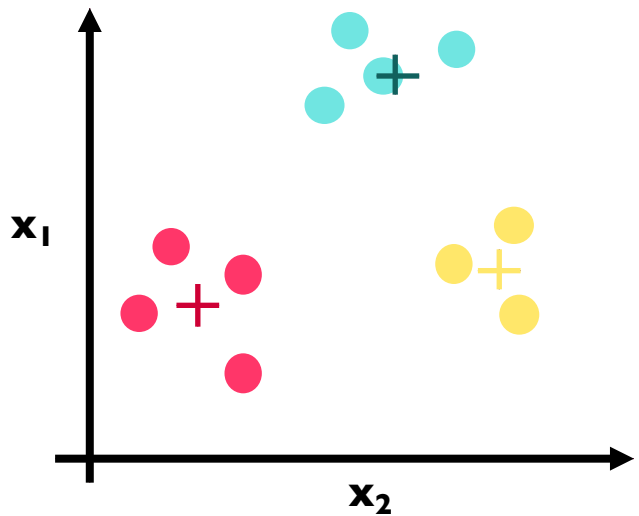
1) Choose k initial centroids

2) For each point:

- *find distance to each centroid*
- *assign point to nearest centroid*

3) Recalculate centroid positions

4) Repeat steps 2-3 until stopping criteria met



KMEANS WEB DEMO

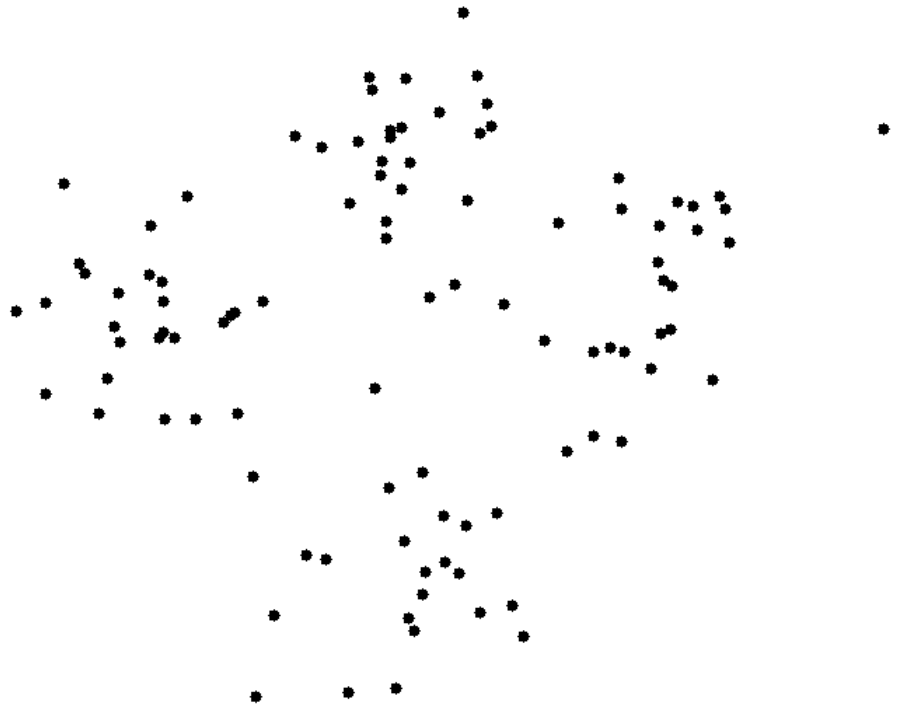
http://shabal.in/visuals/kmeans/1.html

Qn: How many clusters do you see?

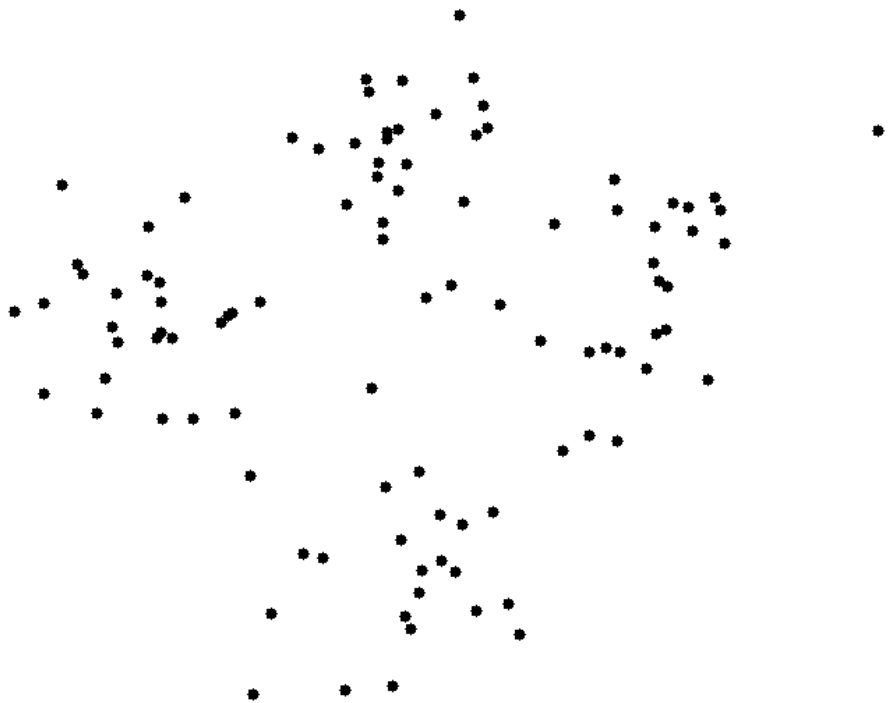
The animation demonstrates the effects of different starting points on the clustering algorithm:

- 4 left-most points
- 4 right-most points
- 4 top-most points
- 4 bottom-most points

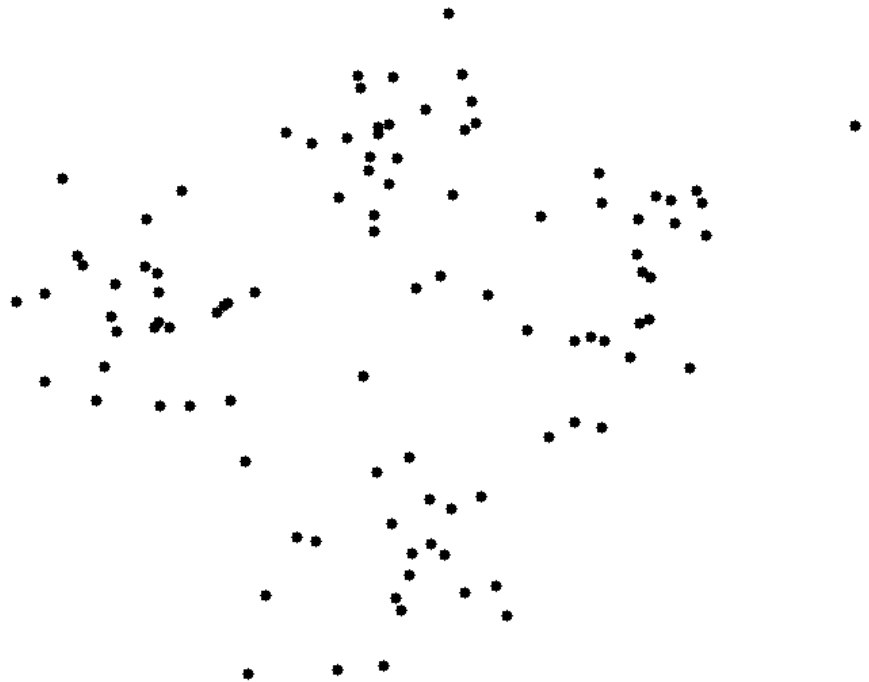
Using the 4 left-most points as the starting points:



Using the 4 right-most points as the starting points:



Using the 4 top-most points as the starting points:



Using the 4 bottom-most points as the starting points:

KMEANS WEB DEMO

Other good demos:

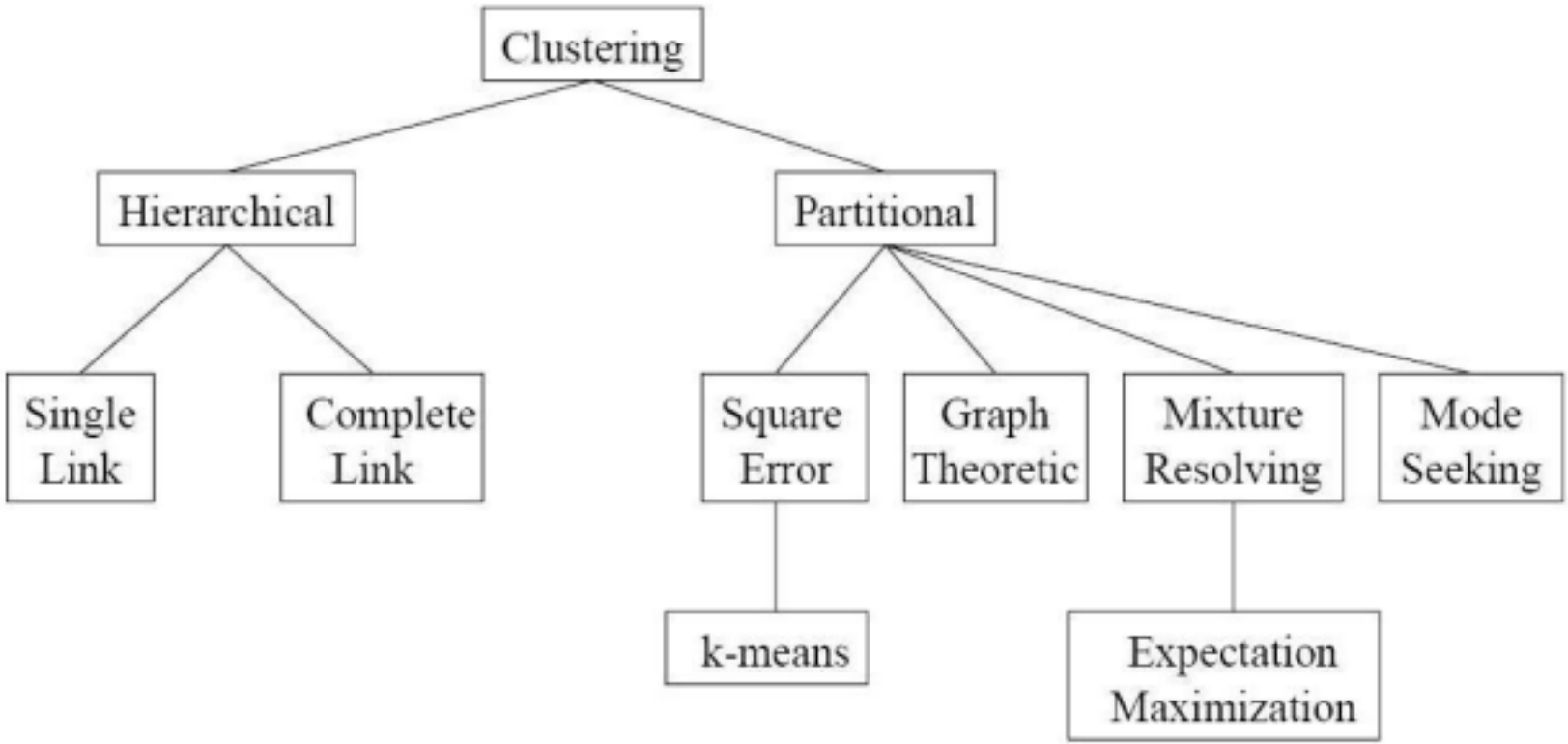
https://www.youtube.com/watch?v=mtkWR8sx0NA

https://www.youtube.com/watch?v=_aWzGGNrcic (especially from timestamp 4:22)

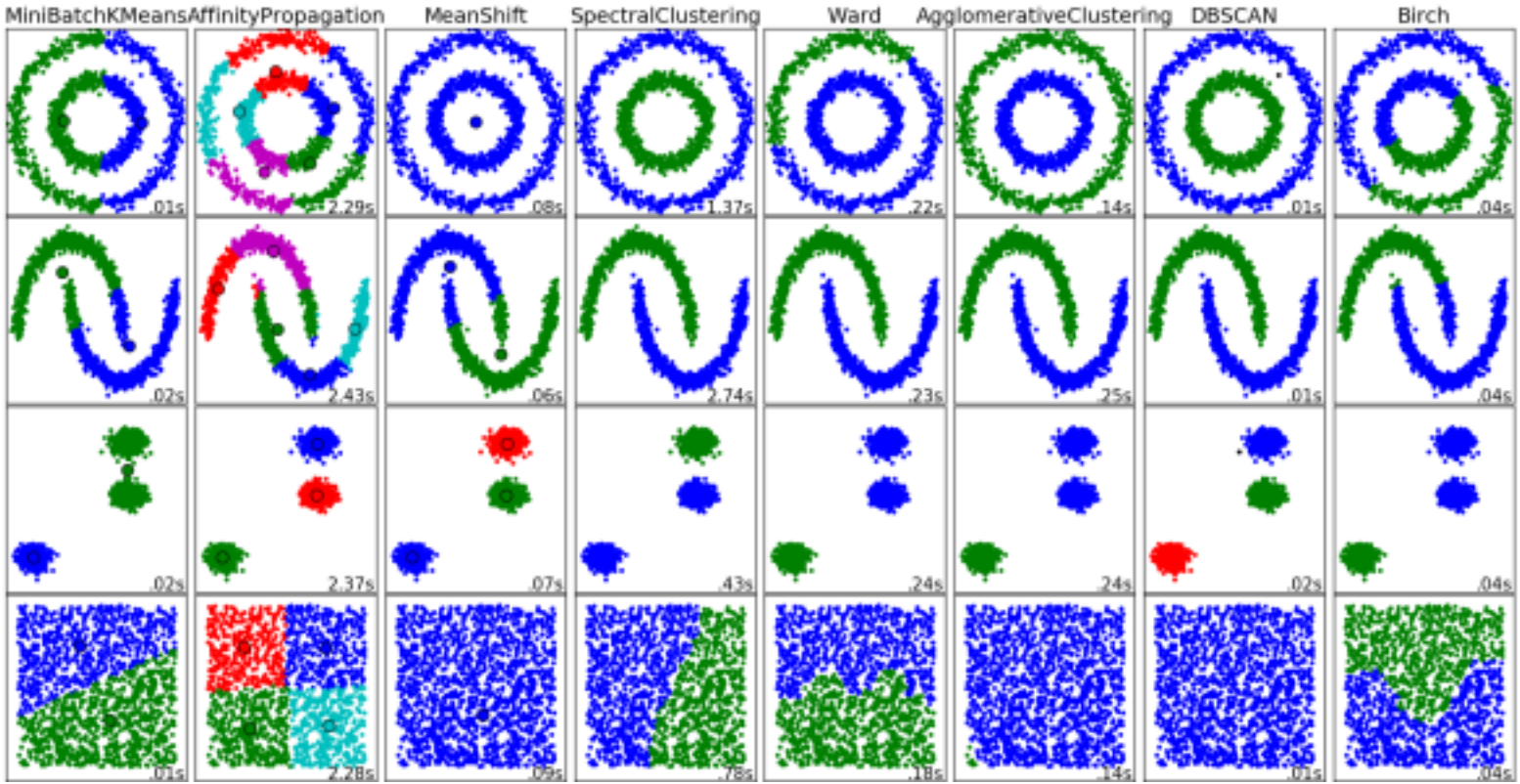
http://www.onmyphd.com/?p=k-means.clustering

DATA SCIENCE PART TIME COURSE

OTHER CLUSTERING ALGORITHMS


Clustering
Hierarchical
Partitional
Single Link
Complete Link
Square Error
Graph Theoretic
Mixture Resolving
Mode Seeking
k-means
Expectation Maximization

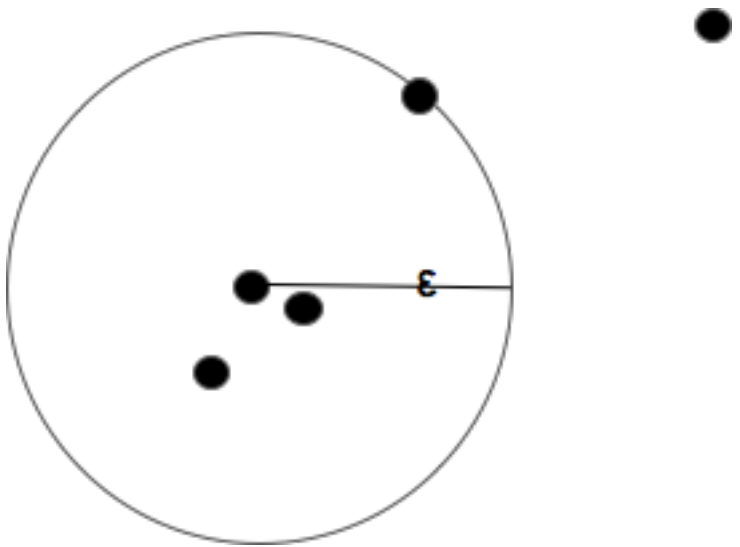
VARIETY OF CLUSTERING OPTIONS



Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

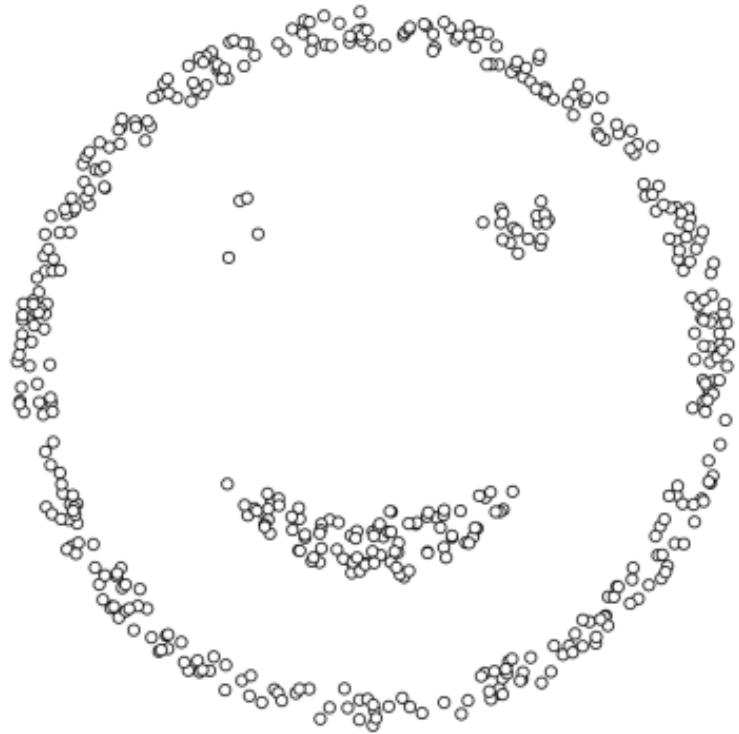
Criteria

- **ε (or Epsilon)** is the radius
- **minPoints** (number of points within the ε-Neighborhood required for classification)

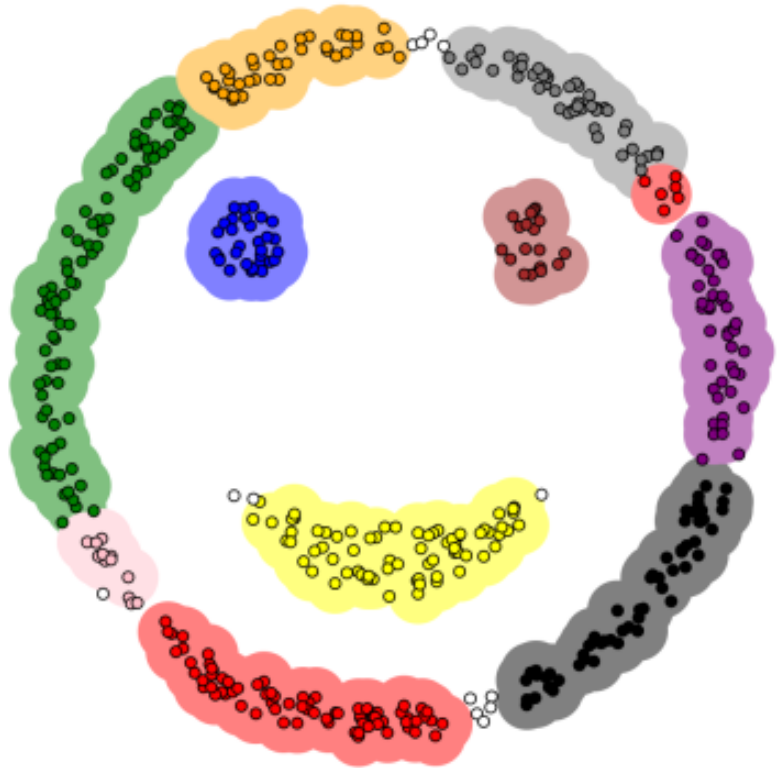


Note

- DBSCAN iterates through every point
- Core object (point meeting the criteria)
- Outlier (outside the radius)

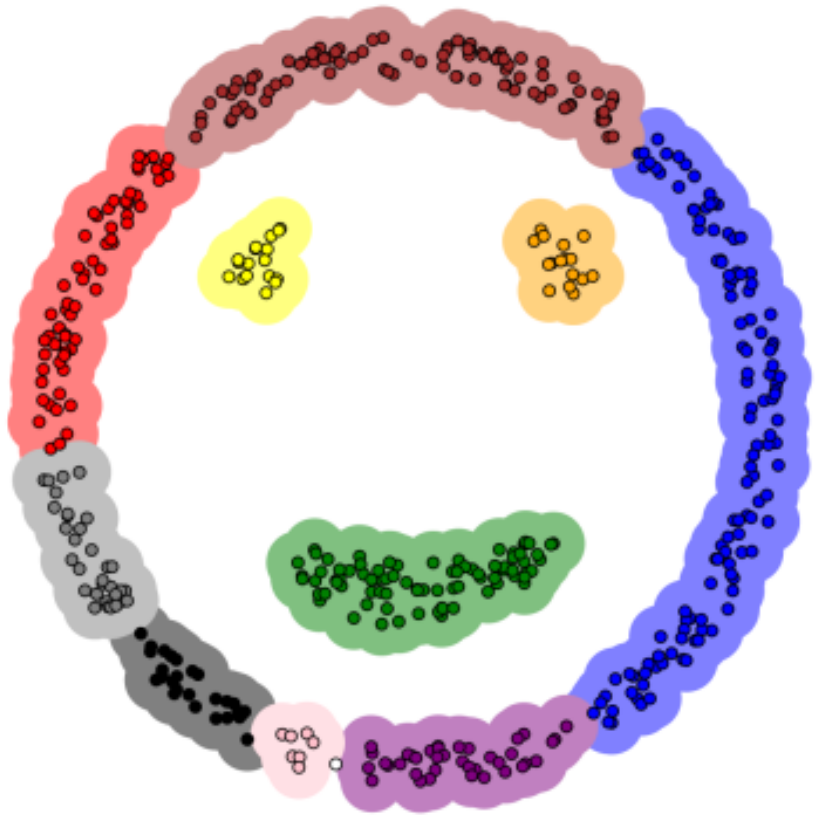


11 Clusters
Patchy



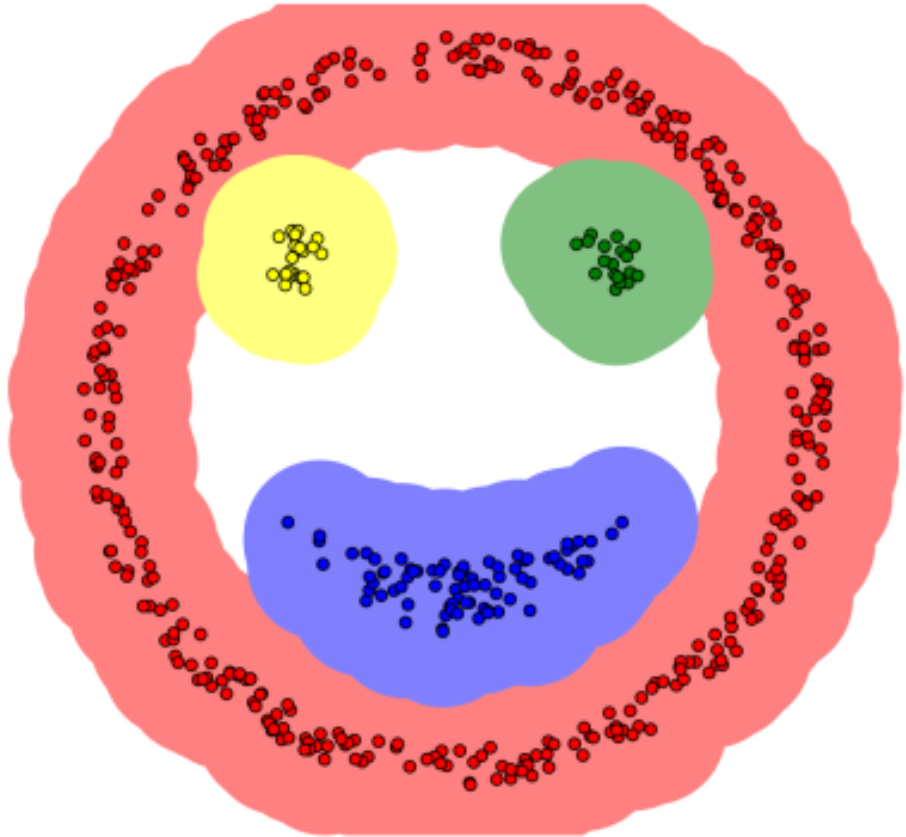
epsilon = 0.80
minPoints = 6

10 Clusters
Less Patchy



epsilon = 0.80
minPoints = 2

4 Clusters
Lion King



epsilon = 1.98
minPoints = 6

https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

Pros

Recovers more complex cluster shapes
Finds the number of clusters
Automatically find outliers

Cons

Requires a distance function
Not as scalable as K-means
Calculating connected components can be difficult

DATA SCIENCE PART TIME COURSE

HOW DO WE KNOW OUR CLUSTERS ARE ANY GOOD?

In general, k-means will converge to a solution and return a partition of k clusters, even if no natural clusters exist in the data.

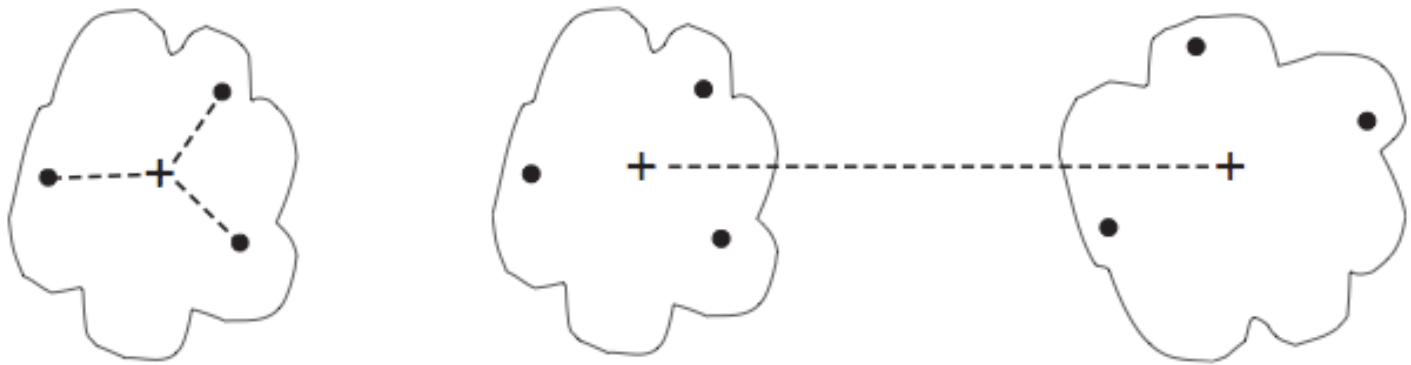
We will look at two validation metrics useful for partitional clustering, cohesion and separation.

Cohesion measures clustering effectiveness within a cluster.

$$\hat{C}(C_i) = \sum_{x \in C_i} d(x, c_i)$$

Separation measures clustering effectiveness between clusters.

$$\hat{S}(C_i, C_j) = d(c_i, c_j)$$



(a) Cohesion. (b) Separation.

Figure 8.28. Prototype-based view of cluster cohesion and separation.

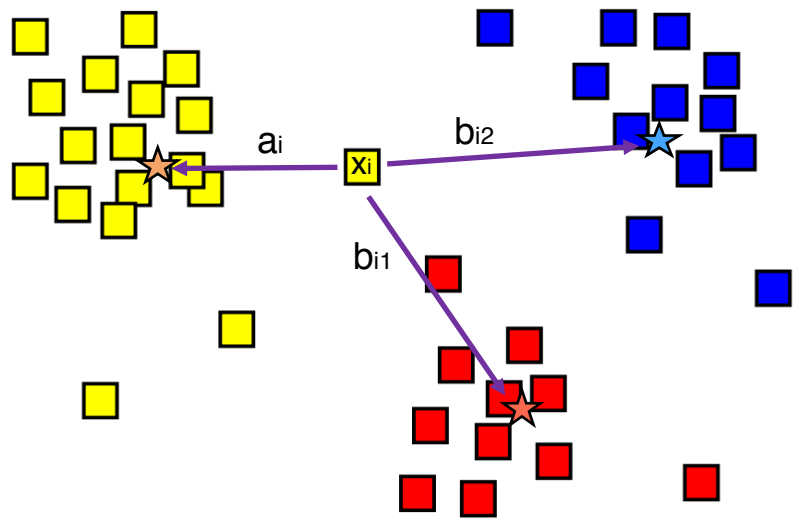
source: http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf

A useful measure that combines the ideas of cohesion and separation is the silhouette coefficient. For point x_i, this is given by:

$$SC_i = \frac{b_i - a_i}{max(a_i, b_i)}$$

such that:

- a_i = average intra-cluster distance to x_i
- b_{ij} = average extra-cluster distance to x_i
- $b_i = \min_j(b_{ij})$



The silhouette coefficient can take values between -1 and 1.

In general, we want **separation to be high** and **cohesion to be low**. This corresponds to a value of **SC close to +1.**

A negative silhouette coefficient means the cluster radius is larger than the space between clusters, and thus **clusters overlap**

One useful application of cluster validation is to determine the best number of clusters for your dataset.

Q: How would you do this?

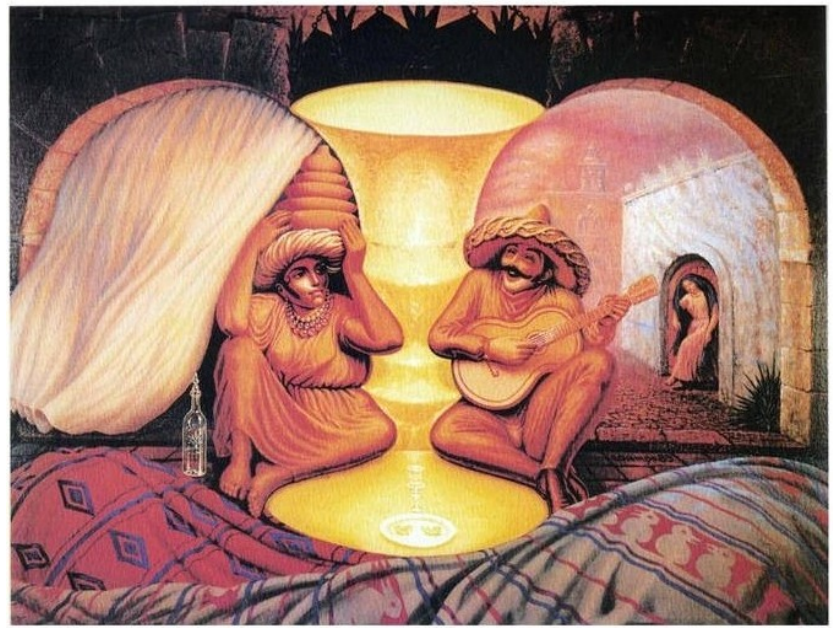
One useful application of cluster validation is to determine the best number of clusters for your dataset.

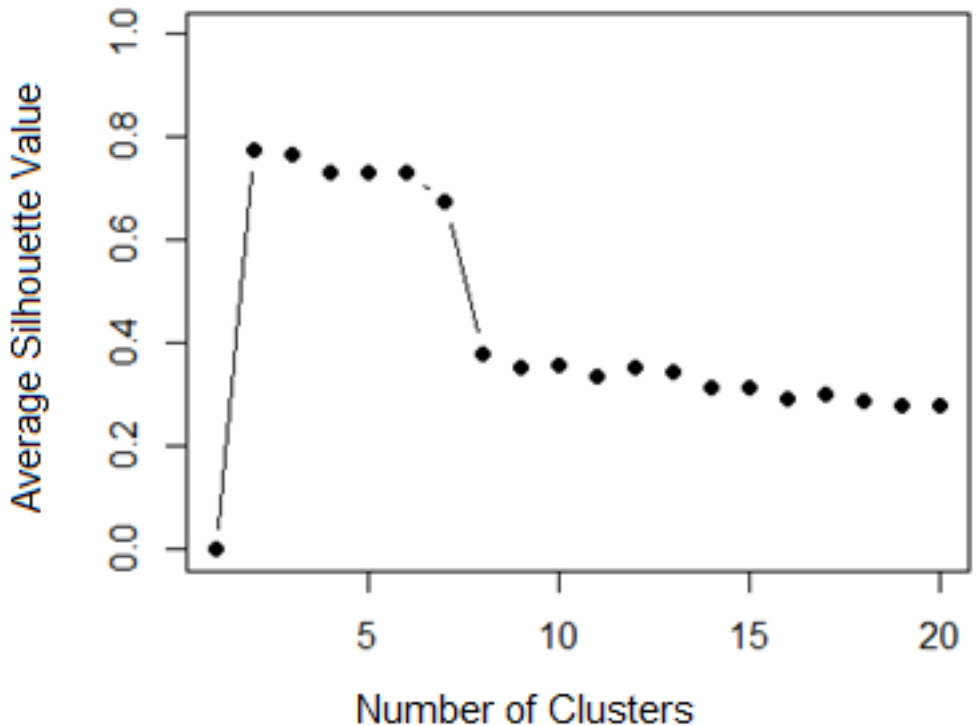
Q: How would you do this?

A: By computing the Silhouette Coefficient for different values of k.

Ultimately, cluster validation and clustering in general are subjective techniques that rely on human interpretation to be meaningful.

Art




Average Silhouette Value
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
20
Number of Clusters

Strengths:

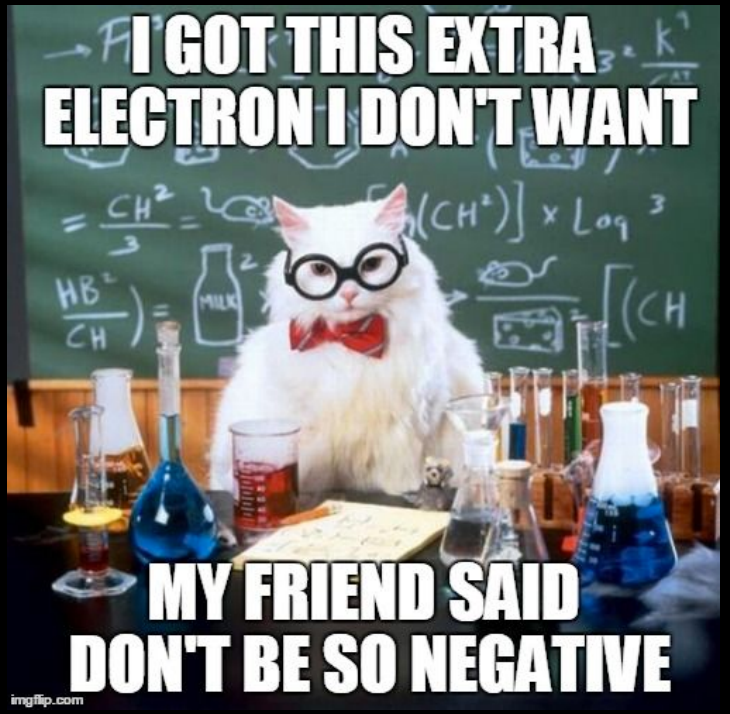
K-means is a popular algorithm because of its computational efficiency and simple and intuitive nature.

Weaknesses:

However, K-means is highly scale dependent, and is not suitable for data with widely varying shapes and densities.

DATA SCIENCE PART TIME COURSE

LAB



DATA SCIENCE

HOMEWORK

Read the following

Chapter 10.3 of Introduction to Statistical Learning - Clustering Methods in Introduction to Statistical Learning (15 pages)

...OR...

DATA SCIENCE - Week 4 Day 1

HOMEWORK

PCA: https://youtu.be/Zbr5hyJNGCs, https://www.youtube.com/watch?v=cnCzY5M3txk

Clustering (Siraj Raval): https://youtu.be/9991JIKnFmk

Python Notebook on Clustering => link

Pre Reading:

http://www.slideshare.net/MrChrisJohnson/algorithmic-music-recommendations-at-spotify

http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html