GENERAL ASSEMBLY

CLUSTERING

Angelo Klin
Katra Analytics

LEARNING OBJECTIVES

- Supervised vs unsupervised algorithms
- Understand and apply k-means clustering
- Density-based clustering: DBSCAN
- Silhouette Metric

PRE-WORK

PRE-WORK REVIEW

- Logit / Sigmoid
- How to optimise for lower false positives or negatives? ROC Curve
- Can logistic regression work with more than two classes?
  - Yes and see here
- How would we measure the accuracy of the classification of any given point with logistic regression?

OPENING

UNSUPERVISED LEARNING

UNSUPERVISED LEARNING

- So far all the algorithms we have used are supervised
  - Each observation came with one or more labels, either
    - Classes (Classification), or
    - Measurements (Regression)

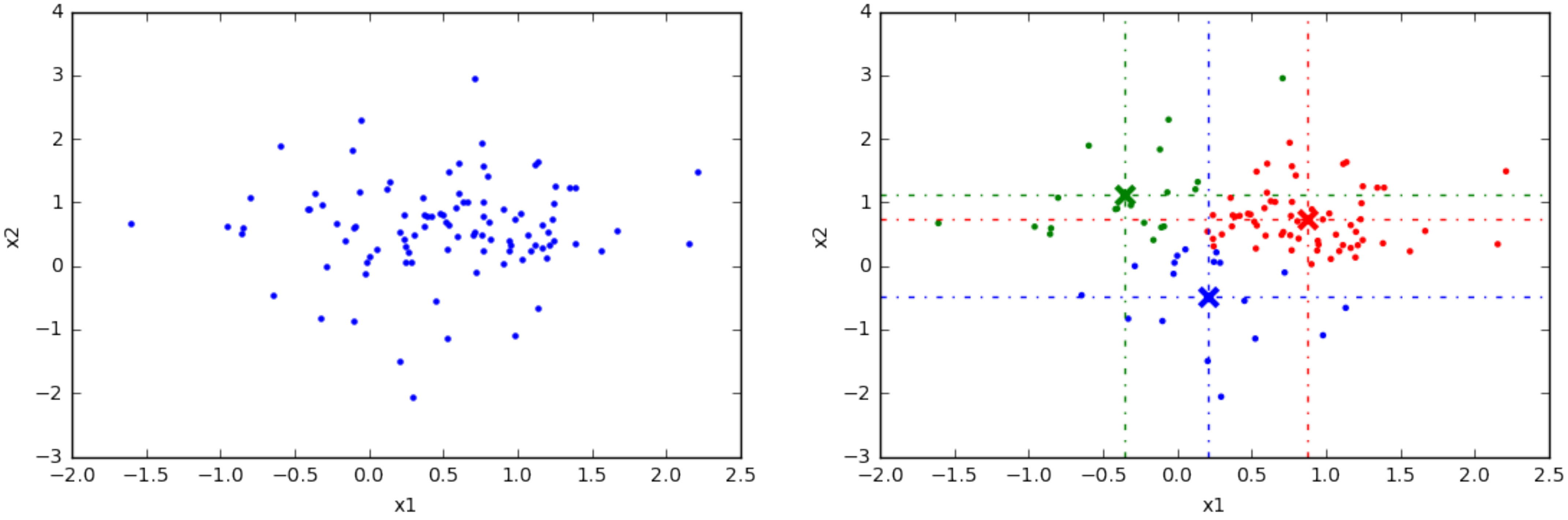
- Unsupervised learning has a different goal: Feature discovery

- Clustering is a common and fundamental example of Unsupervised Learning

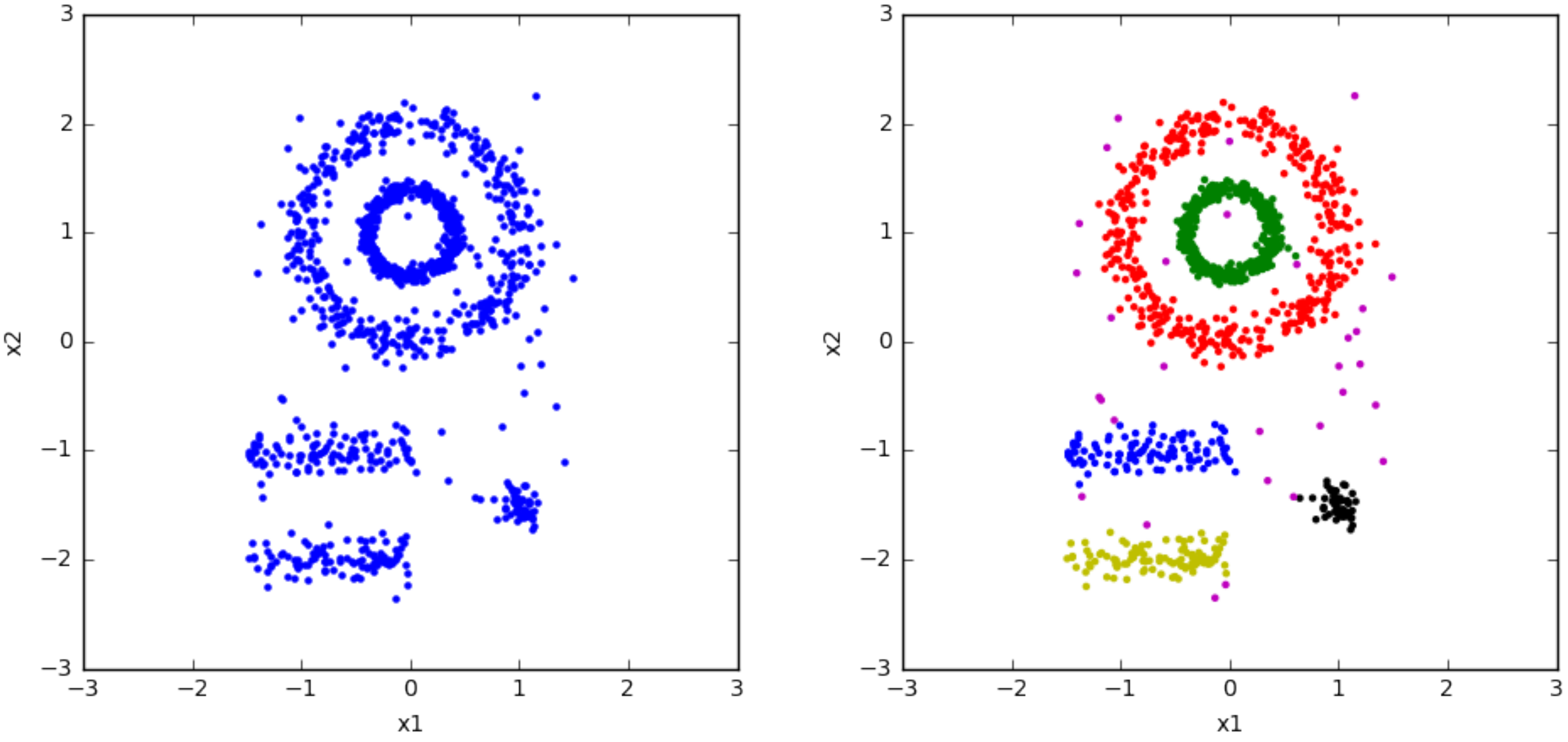
- Clustering algorithms try to find Meaningful Groups within data

CLUSTERING

CLUSTERING: CENTROIDS

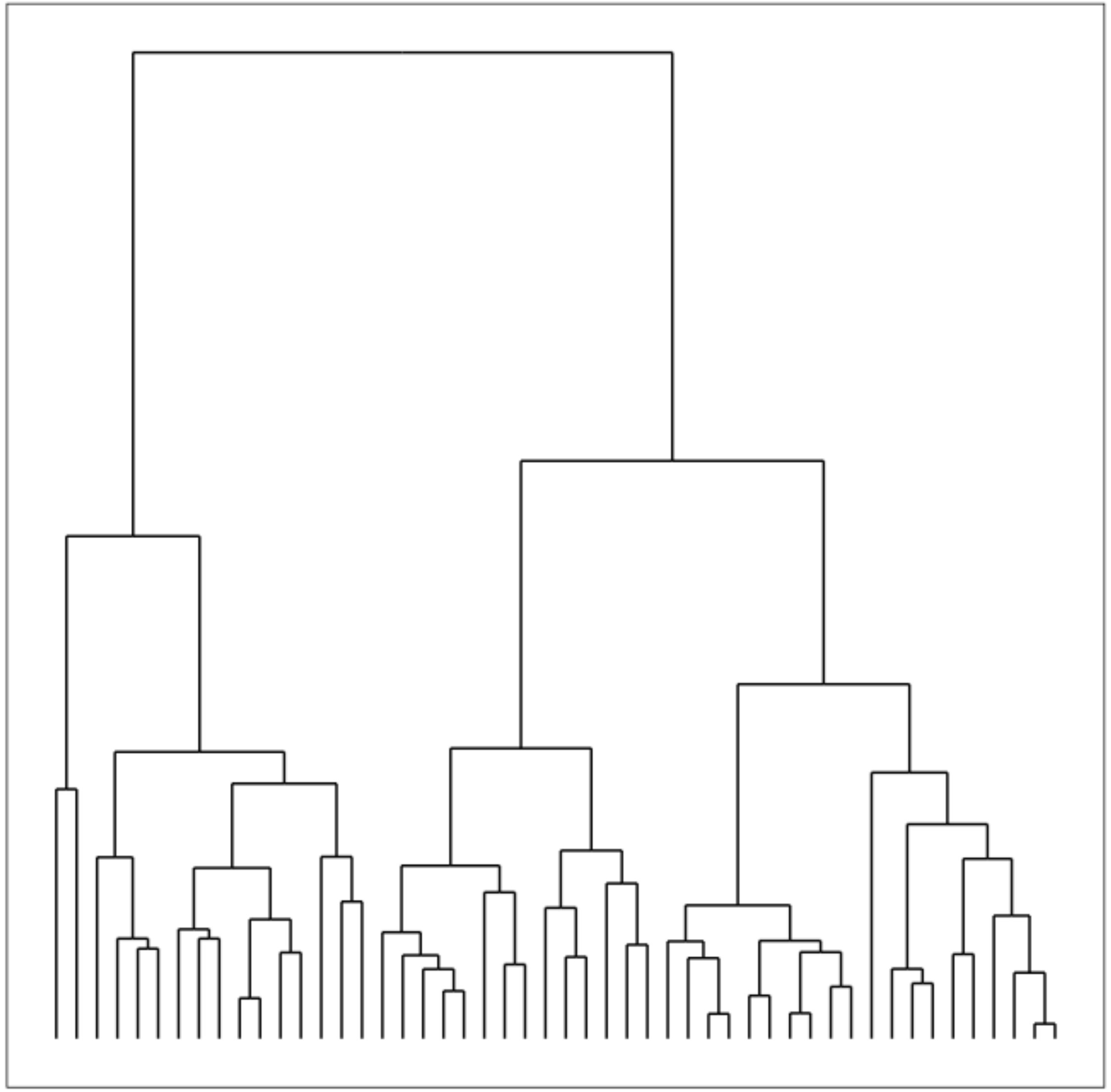


CLUSTERING: DENSITY BASED



CLUSTERING: HIERARCHICAL

- Build hierarchies that form clusters
- Based on classification trees (next lesson)



ACTIVITY: KNOWLEDGE CHECK

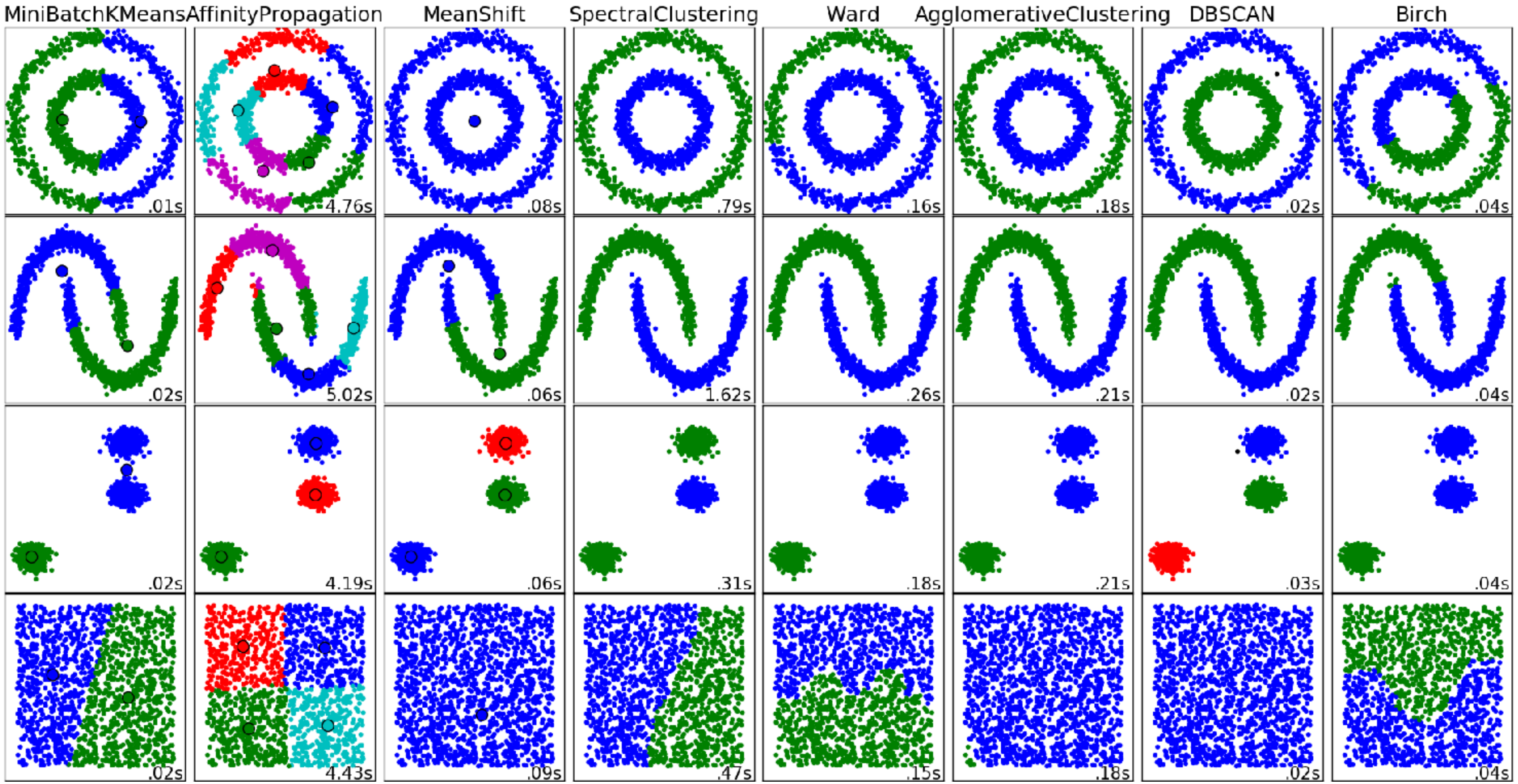
EXERCISE

DIRECTIONS: ANSWER THE FOLLOWING QUESTIONS

1. How is unsupervised learning different from classification?

CLUSTERING: DENSITY BASED

There are many clustering algorithms



ACTIVITY: KNOWLEDGE CHECK

EXERCISE

DIRECTIONS: ANSWER THE FOLLOWING QUESTIONS

1. Can you think of a real-world clustering application?
   a. Recommendation Systems, e.g. Netflix genres
   b. Medical Imaging: differentiate tissues
   c. Identifying market segments
   d. Discover communities in social networks
   e. Lots of applications for genomic sequences (homologous sequences, genotypes)
   f. Earthquake epicentres
   g. Fraud detection

K-MEANS: CENTROID CLUSTERING

K-MEANS: CLUSTERING

- k-Means clustering is a popular centroid-based clustering algorithm
- Basic idea: find k clusters in the data centrally located around various mean points

K-MEANS: CLUSTERING

- k-Means seeks to minimise the sum of squares about the means
- Precisely, find k subsets S_1, …, S_k of the data with means μ_1, …, μ_k that minimises:

$$arg\,\underset{S}{min} \sum_{i=1}^{k} \sum_{x \in S_i} \left\| x - \mu_i \right\|^2$$

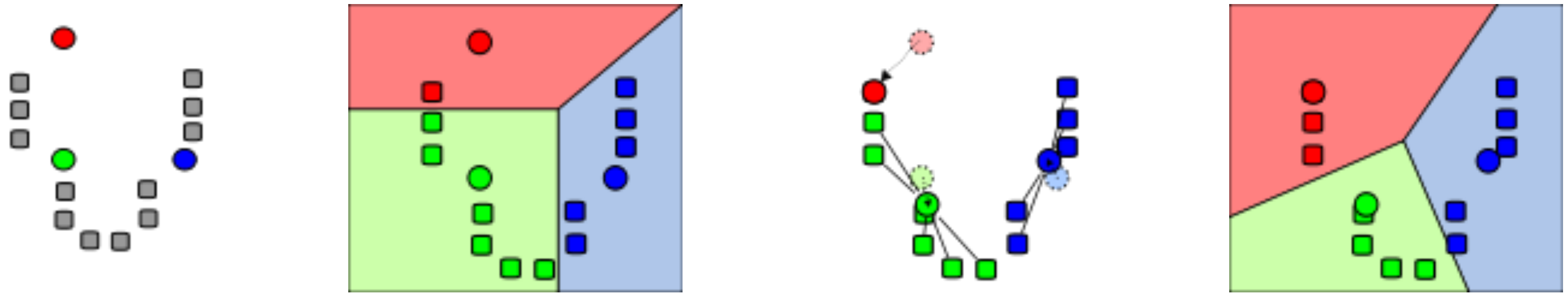
K-MEANS: CLUSTERING

- This is a computationally difficult problem to solve so we rely on heuristics

- The "standard" heuristic is called "Lloyd's Algorithm":
  - Start with k initial mean values
  - Data points are then split up into a Voronoi diagram
    - Each point is assigned to the "closest" mean
  - Calculate new means based on centroids of points in the cluster
  - Repeat until clusters do not change

K-MEANS: CLUSTERING

- Start with k initial mean values
- Data points are then split up into a Voronoi diagram
- Calculate new means based on centroids of points in the cluster



K-MEANS: CLUSTERING

- Awesome Demo

- Let's try it out!

```
from sklearn.cluster import KMeans

est = KMeans(n_clusters = 3)
est.fit(X)

labels = est.labels_
```

ACTIVITY: KNOWLEDGE CHECK

EXERCISE

DIRECTIONS: ANSWER THE FOLLOWING QUESTIONS

1. How do we assign meaning to the clusters we find?
2. Do clusters always have meaning?

K-MEANS: CLUSTERING

- Assumptions are important! k-Means assumes:
  - k is the correct number of clusters
  - the data is isotropically distributed (circular/spherical distribution)
  - the variance is the same for each variable
  - clusters are roughly the same size

- Nice counter examples / cases where assumptions are not met:
  - K-means clustering is not a free lunch
  - Scikit-Learn Examples

K-MEANS: CLUSTERING

- Netflix prize: Predict how users will rate a movie
  - How might you do this with clustering?
  - Cluster similar users together and take the average rating for a given movie by users in the cluster (which have rated the movie)
  - Use the average as the prediction for users that have not yet rated the movie
- In other words, fit a model to users in a cluster for each cluster and make predictions per cluster
- k-Means for the Netflix Prize (pdf)

DBSCAN: DENSITY BASED CLUSTERING

DBSCAN CLUSTERING

- DBSCAN: Density-based spatial clustering of applications with noise (1996)

- Main idea: Group together closely-packed points by identifying
  - Core points
  - Reachable points
  - Outliers (not reachable)

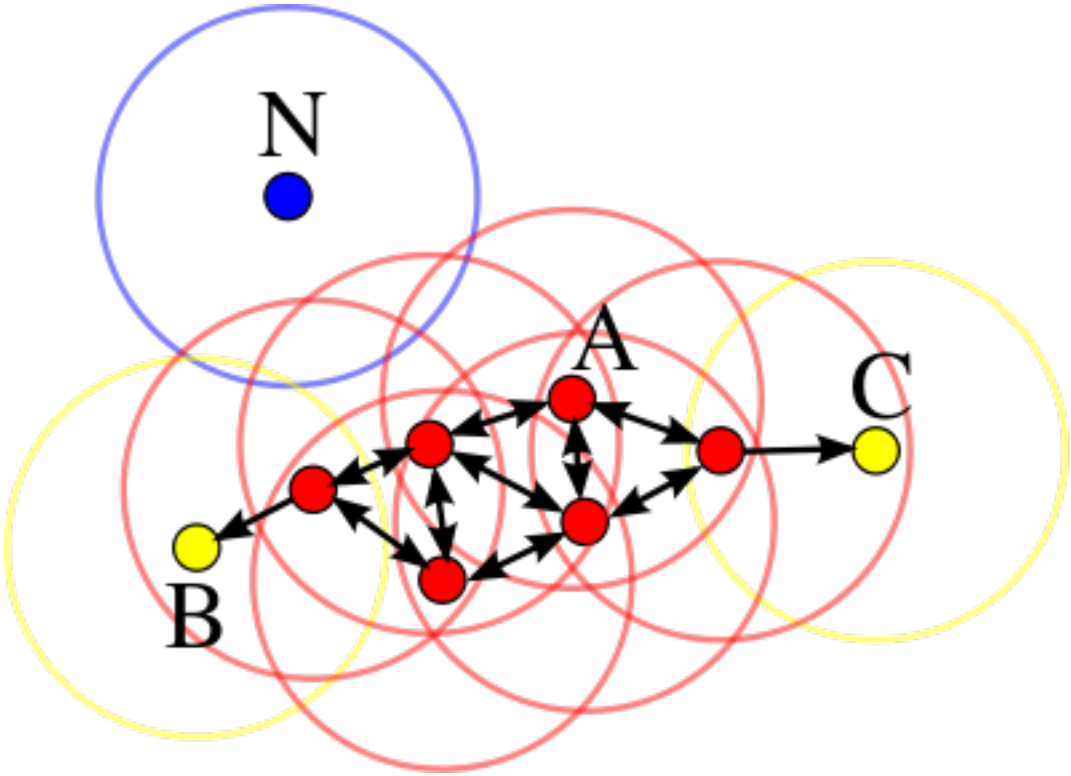
- Two parameters:
  - min_samples
  - eps

DBSCAN CLUSTERING

- Core points: at least min_samples points within eps of the core point
  - Such points are **directly reachable** from the core point

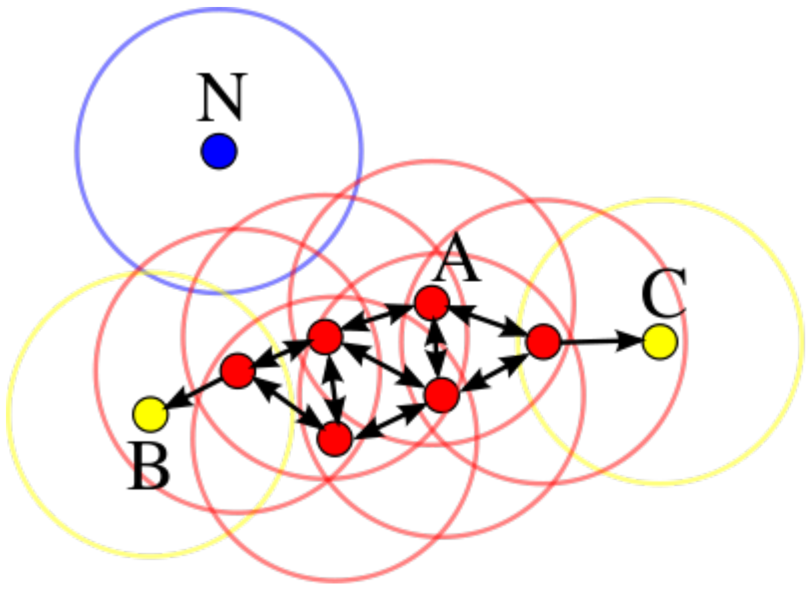
- Reachable: point q is reachable from p if there is a path of core points from p to q
  - On the image p = {A}, q = {B, C}

- Outlier: not reachable (N)



DBSCAN CLUSTERING

- A cluster is a collection of connected core and reachable points
- Awesome Demo
- Let's try it out!




```
from sklearn.cluster import DBSCAN

est = DBSCAN(eps = 0.5, min_samples = 10)
est.fit(X)
labels = est.labels_
```

ACTIVITY: KNOWLEDGE CHECK

EXERCISE

DIRECTIONS: ANSWER THE FOLLOWING QUESTIONS

1. How does DBSCAN differ from k-Means?

DBSCAN CLUSTERING

- DBSCAN advantages:
  - Can find arbitrarily-shaped clusters
  - Do not have to specify number of clusters
  - Robust to outliers

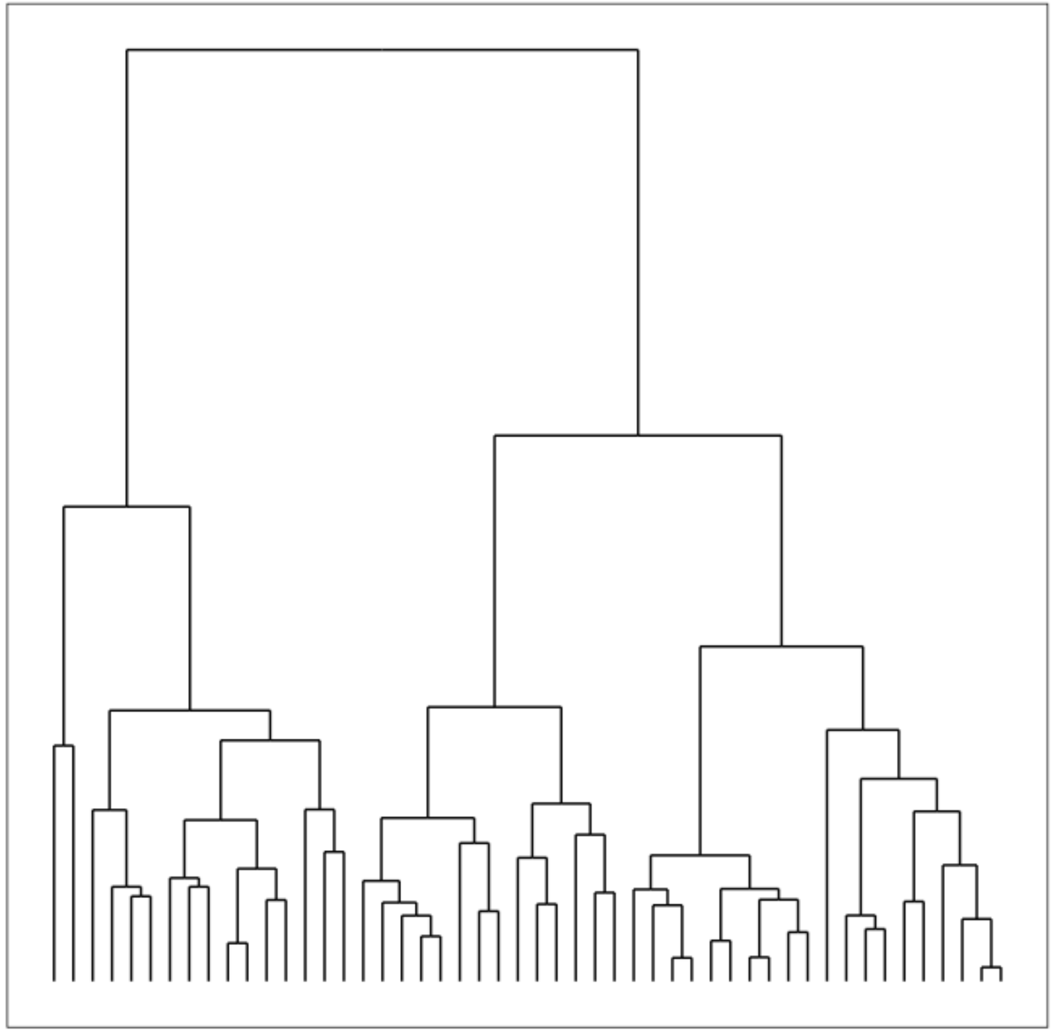
- DBSCAN disadvantages:
  - Does not work well when clusters are of varying densities
    - Hard to chose parameters that work for all clusters
  - Can be hard to chose correct parameters regardless

HIERARCHICAL CLUSTERING

HIERARCHICAL CLUSTERING

- Build hierarchies that form clusters

- Based on classification trees (next lesson)



HIERARCHICAL CLUSTERING

- We will discuss the details once we cover decision trees. For now we can black box the model and fit with scikit-learn
- Let's try it out!

```
from sklearn.cluster import AgglomerativeClustering

est = AgglomerativeClustering(n_clusters = 4)
est.fit(X)

labels = est.labels_
```

CLUSTERING METRICS

CLUSTERING METRICS

- As usual we need a metric to evaluate model fit
- For clustering we use a metric called the Silhouette Coefficient
    - a is the mean distance between a sample and all other points in the cluster
    - b is the mean distance between a sample and all other points in the *nearest* cluster

- The Silhouette Coefficient is:

$$\frac{b - a}{max(a, b)}$$

    - Ranges between 1 and -1
    - Average over all points to judge the cluster algorithm

CLUSTERING METRICS

```
from sklearn import metrics
from sklearn.cluster import KMeans

kmeans_model = KMeans(n_clusters = 3, random_state = 1).fit(X)
labels = kmeans_model.labels_

metrics.silhouette_score(X, labels, metric = "euclidean")
```

CLUSTERING METRICS

- There are a number of other metrics based on:
  - Mutual Information
  - Homogeneity
  - Adjusted Rand Index (when you know the labels on the training data)

PUTTING IT TOGETHER

CLUSTERING, CLASSIFICATION AND REGRESSION

ACTIVITY: KNOWLEDGE CHECK

EXERCISE

DIRECTIONS: ANSWER THE FOLLOWING QUESTIONS

1. How might we combine clustering and classification?

CLUSTERING, CLASSIFICATION AND REGRESSION

- We can use clustering to discover new features and then use those features for either classification or regression
- For classification, we could use e.g. k-NN to classify new points into the discovered clusters
- For regression, we could use a dummy variable for the clusters as a variable in our regression

ACTIVITY: CLUSTERING + CLASSIFICATION

EXERCISE

DIRECTIONS: ANSWER THE FOLLOWING QUESTIONS (15 MINUTES)

1. Using the starter code, perform a k-means clustering on the flight delay data
2. Use the clustering to create a classifier

TOPIC REVIEW

TOPIC REVIEW

- Clustering is used to discover features, e.g. segment users or assign labels (such as species)

- Clustering may be the goal or a step in a data science pipeline

BEFORE NEXT CLASS

DUE DATE

- Project
  - Unit Project 4

Q & A