Unsupervised ML - Clustering: K-Means

Brock Tibert
btibert@bu.edu

Class Overview

- Quick Review of Unsupervised ML
- Review of K-Means Clustering
- Walk-through of K-Means clustering in python
- Discussion of a density-based alternative
- Hands-on in python
- K-Means Exercise

CourseKey

Team Projects

I will start to post some supplemental tools that you might want to consider using.

- Based on capstone groups
- Example deliverables on QuestromTools
- Pick a real problem that you want to explore
- You will need to identify dataset(s) that will help you work through the problem
  - You may need to create datasets!
- Your projects will need to include work from the themes we cover in class, but your problem will/should dictate what method(s) you use
  - NOTE: You are not only limited to this class though. You should draw on what you have learned throughout the program so far
- Your deliverables
  - Brief in-class presentation
  - An executive summary detailing your findings and recommendations

K-Means Clustering

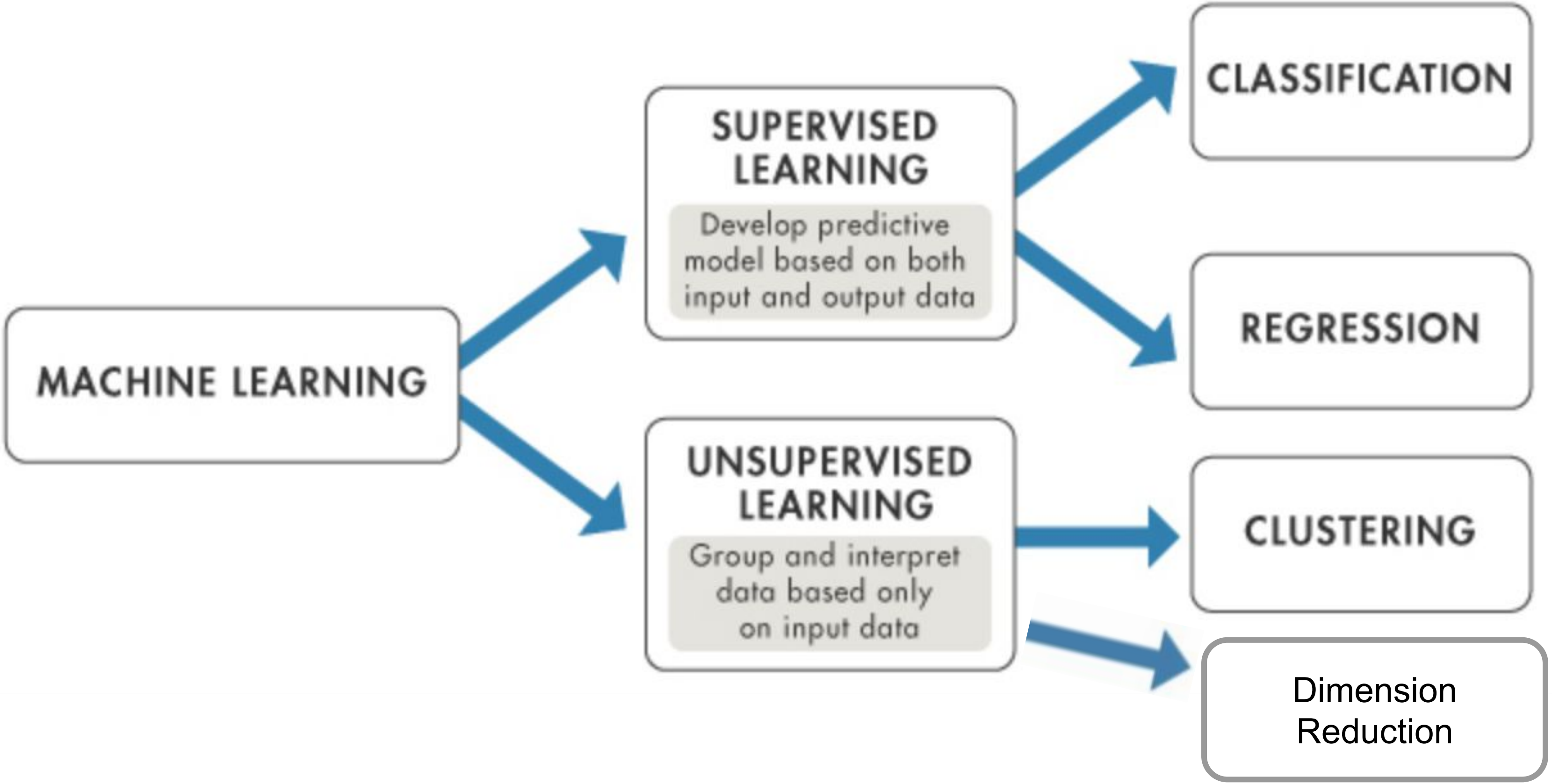
Pattern Discovery

Pattern Discovery

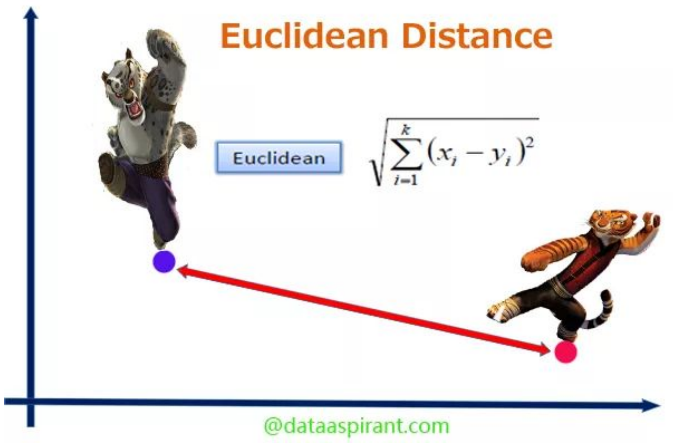
"...the discovery of interesting, unexpected, or valuable structures in large data sets."

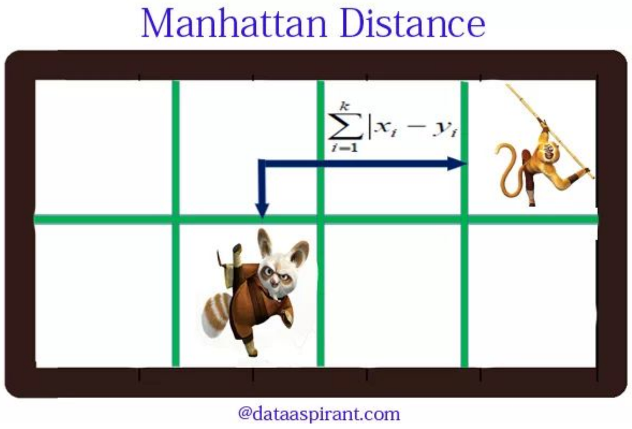
- David Hand

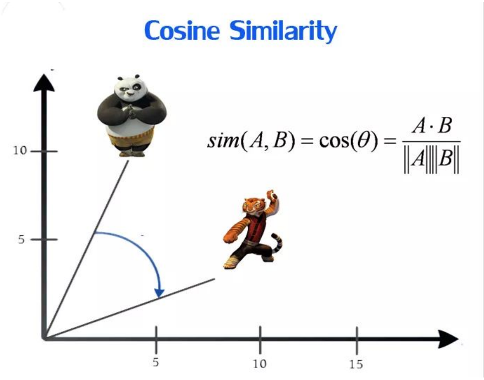
Review


MACHINE LEARNING
SUPERVISED LEARNING
Develop predictive model based on both input and output data
UNSUPERVISED LEARNING
Group and interpret data based only on input data
CLASSIFICATION
REGRESSION
CLUSTERING
Dimension Reduction

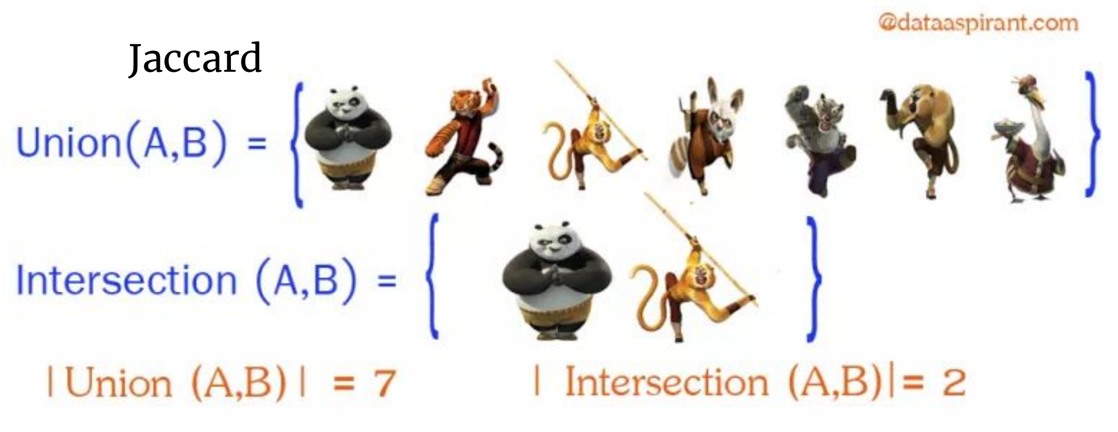
Coming back to the concept of "distance"





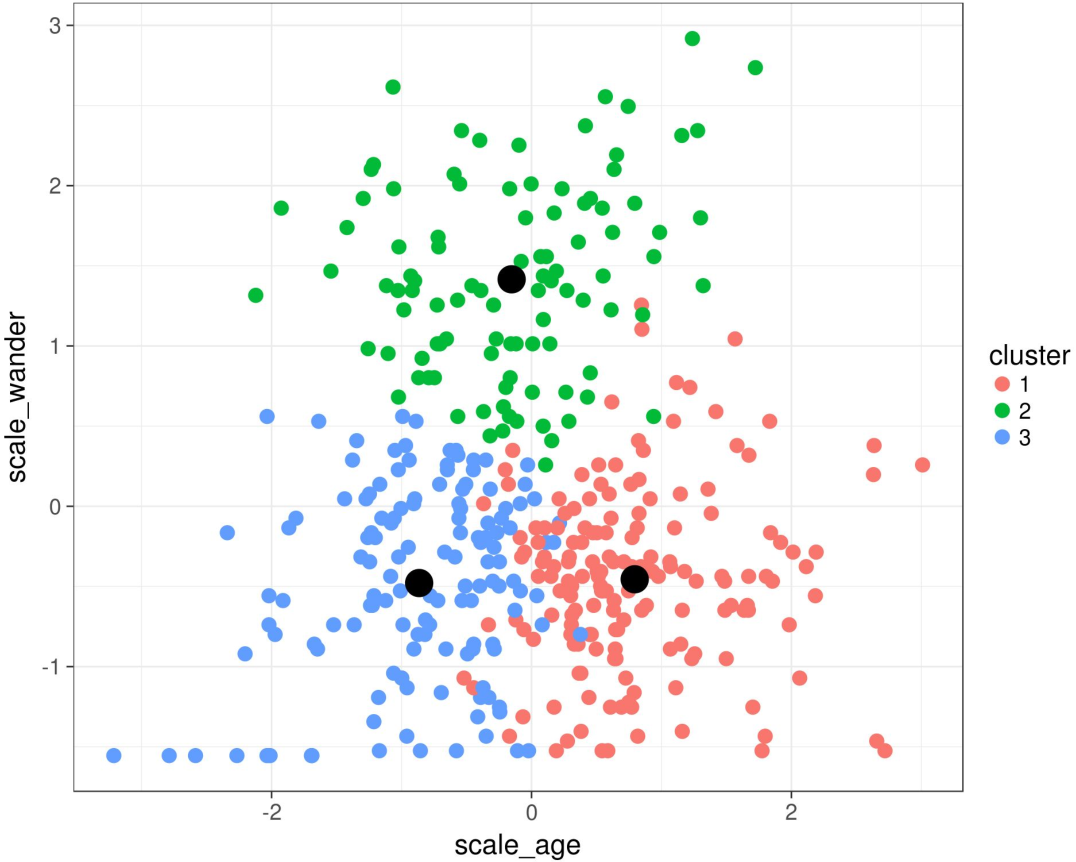


Jaccard

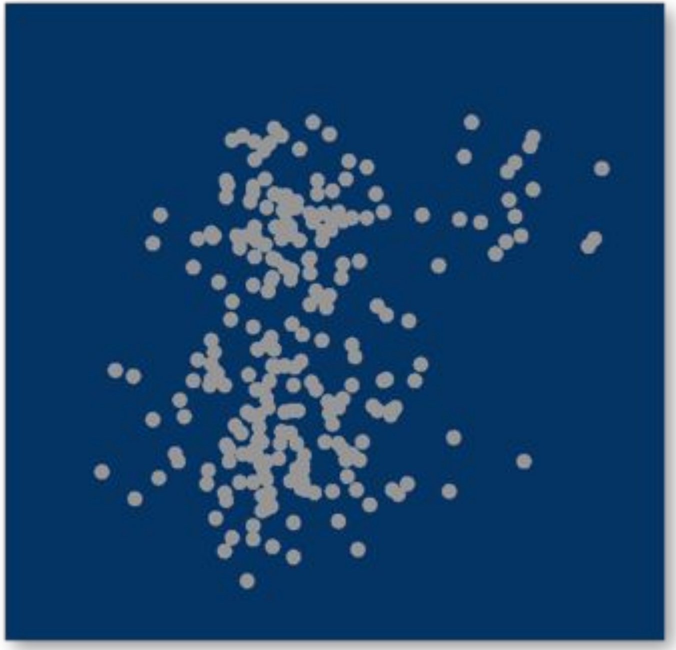


K-Means Overview

K-Means

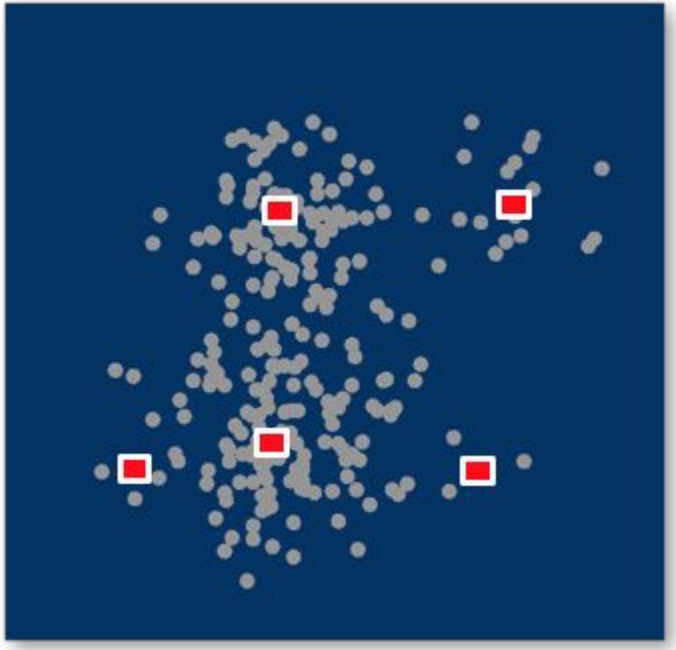


k-Means Clustering Algorithm



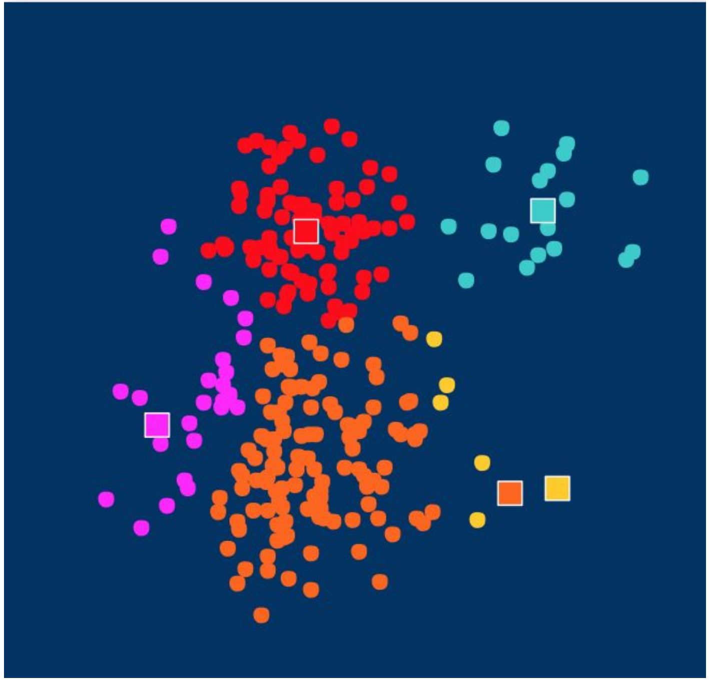
1. Select inputs
2. Select k cluster centers
3. Assign cases to closest center
4. Update cluster centers
5. Re-assign cases
6. Repeat steps 4 + 5 until convergence

k-Means Clustering Algorithm



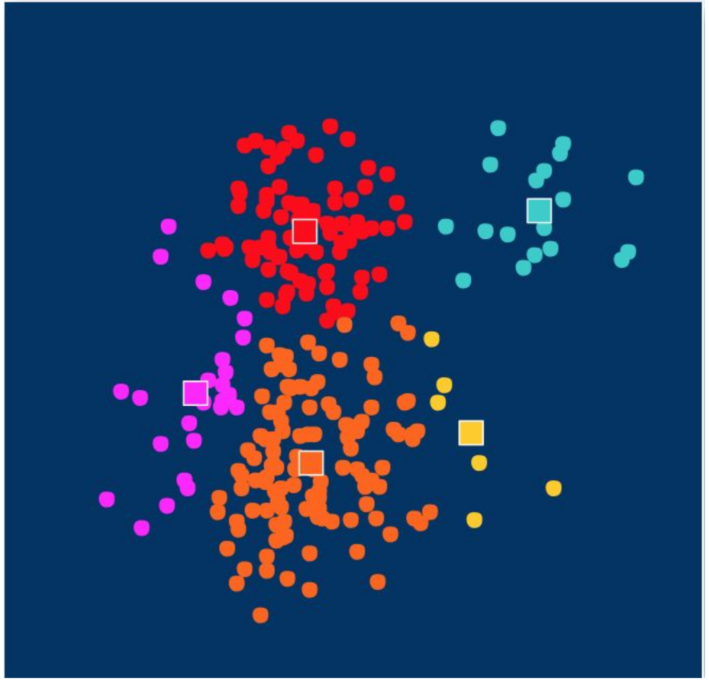
1. Select inputs
2. Select *k* cluster centers
3. Assign cases to closest center
4. Update cluster centers
5. Re-assign cases
6. Repeat steps 4 + 5 until convergence

k-Means Clustering Algorithm



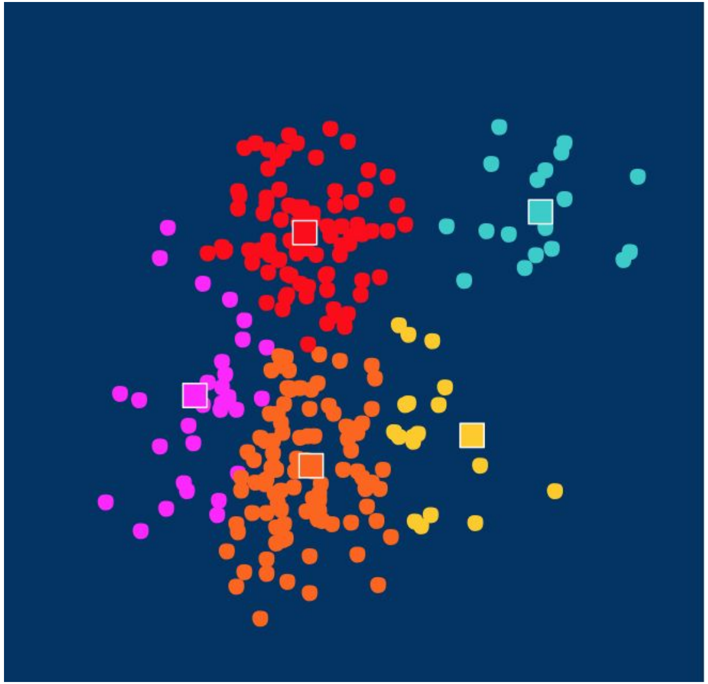
1. Select inputs
2. Select *k* cluster centers
3. Assign cases to closest center
4. Update cluster centers
5. Re-assign cases
6. Repeat steps 4 + 5 until convergence

k-Means Clustering Algorithm



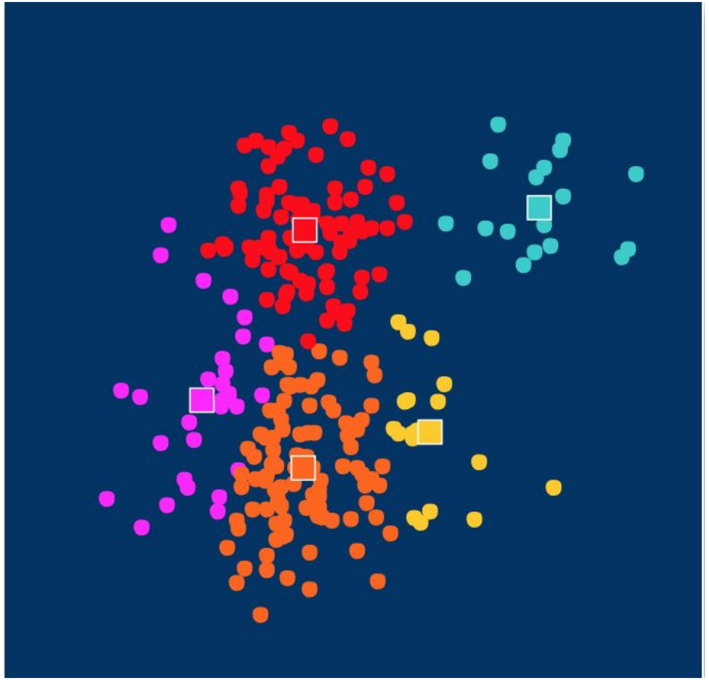
1. Select inputs
2. Select *k* cluster centers
3. Assign cases to closest center
4. Update cluster centers
5. Re-assign cases
6. Repeat steps 4 + 5 until convergence

k-Means Clustering Algorithm



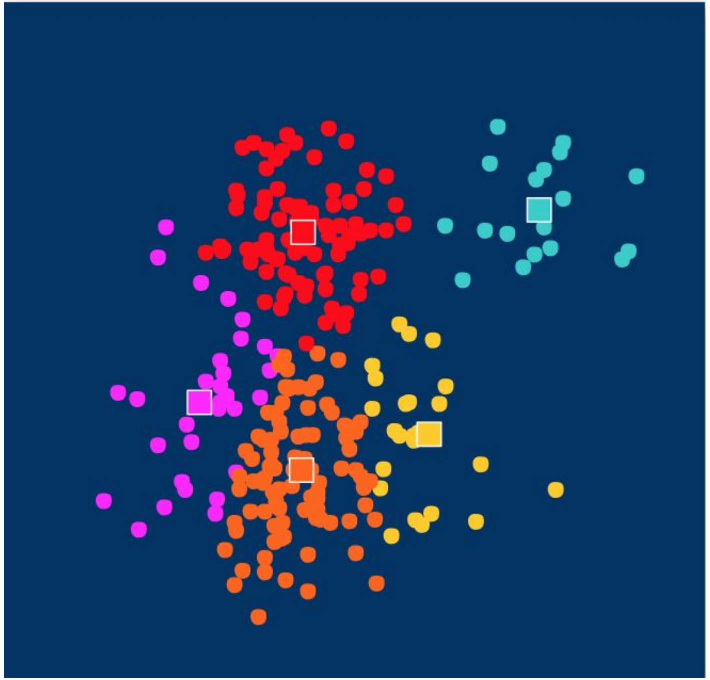
1. Select inputs
2. Select k cluster centers
3. Assign cases to closest center
4. Update cluster centers
5. Re-assign cases
6. Repeat steps 4 + 5 until convergence

k-Means Clustering Algorithm



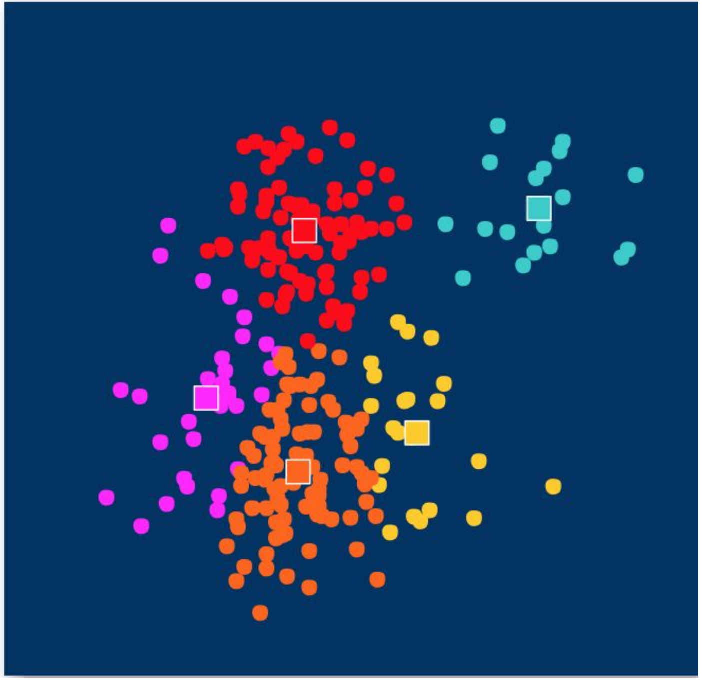
1. Select inputs
2. Select k cluster centers
3. Assign cases to closest center
4. Update cluster centers
5. Re-assign cases
6. Repeat steps 4 + 5 until convergence

k-Means Clustering Algorithm



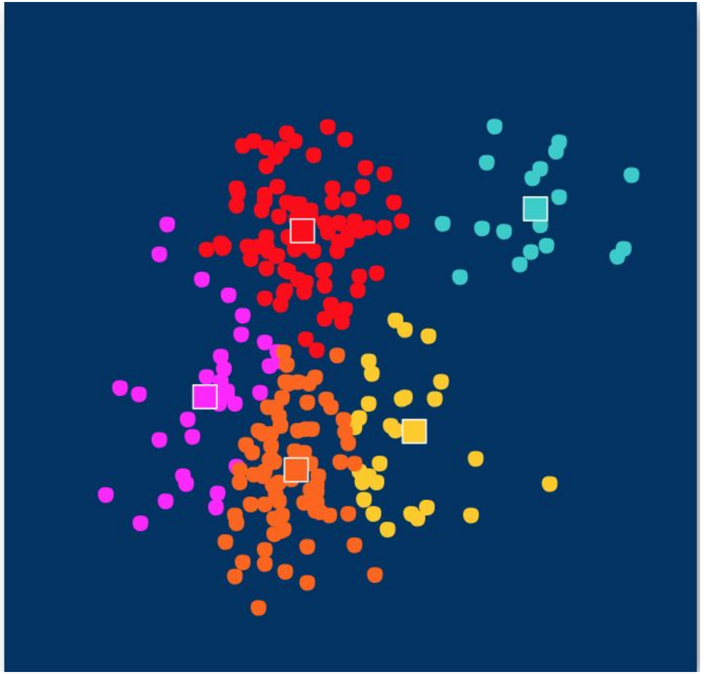
1. Select inputs
2. Select k cluster centers
3. Assign cases to closest center
4. Update cluster centers
5. Re-assign cases
6. Repeat steps 4 + 5 until convergence

k-Means Clustering Algorithm



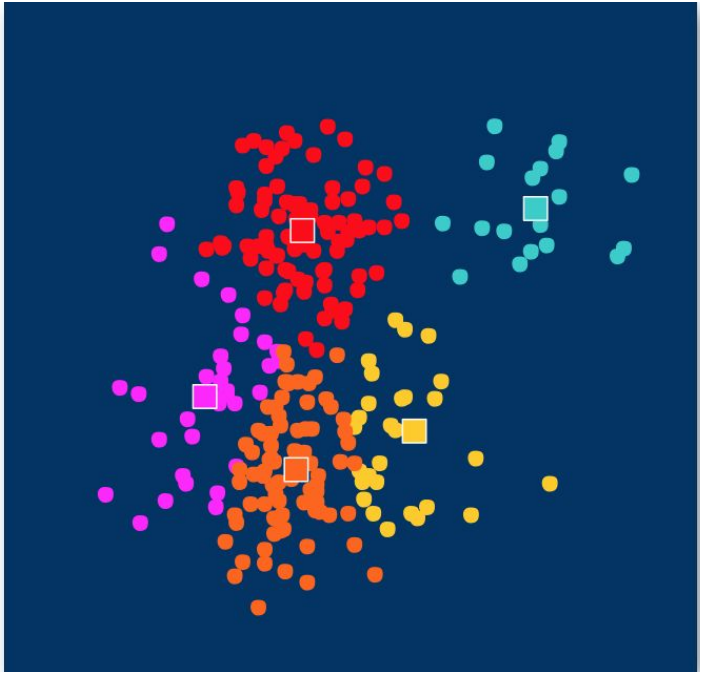
1. Select inputs
2. Select *k* cluster centers
3. Assign cases to closest center
4. Update cluster centers
5. Re-assign cases
6. Repeat steps 4 + 5 until convergence

k-Means Clustering Algorithm



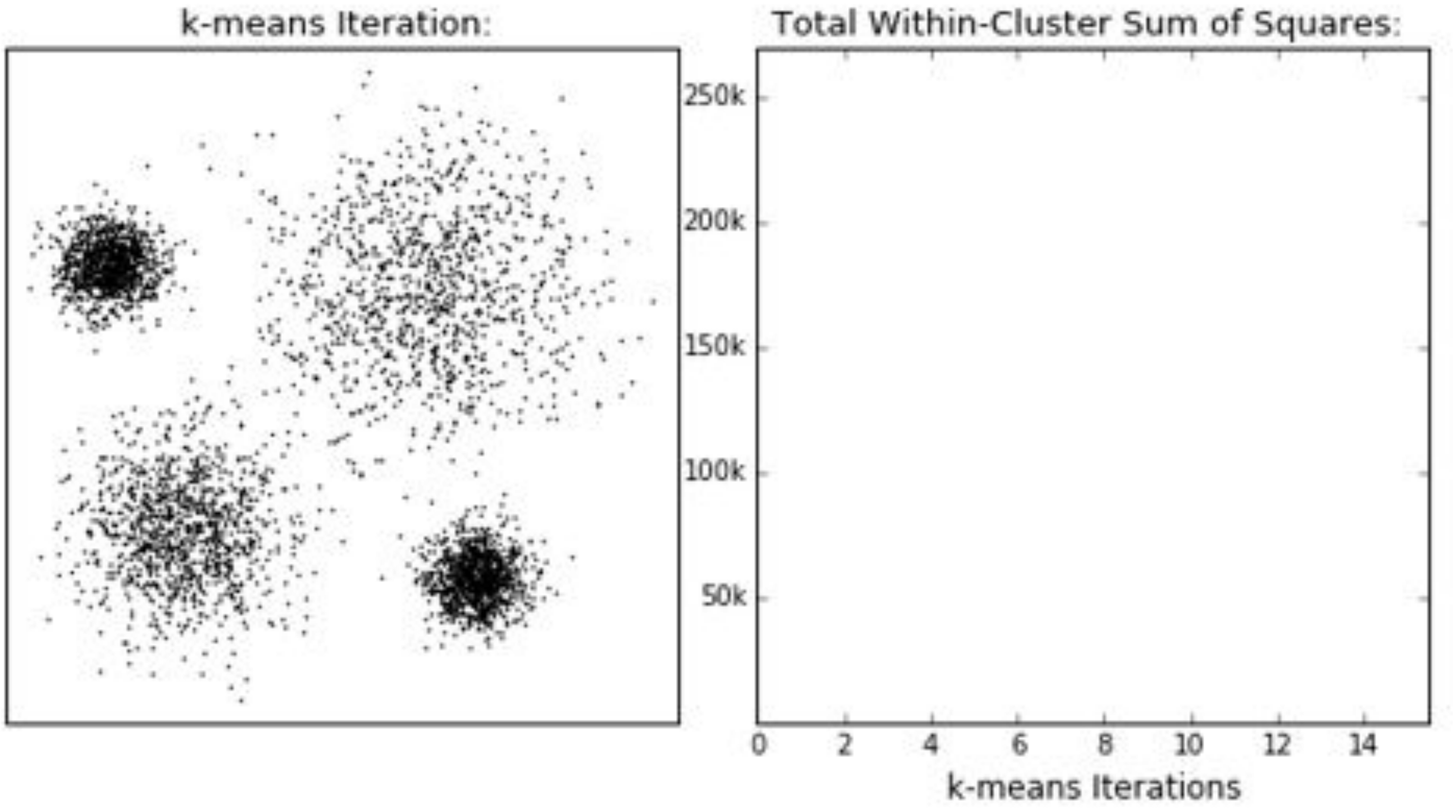
1. Select inputs
2. Select k cluster centers
3. Assign cases to closest center
4. Update cluster centers
5. Re-assign cases
6. Repeat steps 4 + 5 until convergence

k-Means Clustering Algorithm



1. Select inputs
2. Select k cluster centers
3. Assign cases to closest center
4. Update cluster centers
5. Re-assign cases
6. Repeat steps 4 + 5 until convergence

K-Means - An Animated Approach



What Value of *k* to Use

The number of seeds, k, typically translates to the final number of clusters that are obtained. The choice of k can be made using a variety of methods.

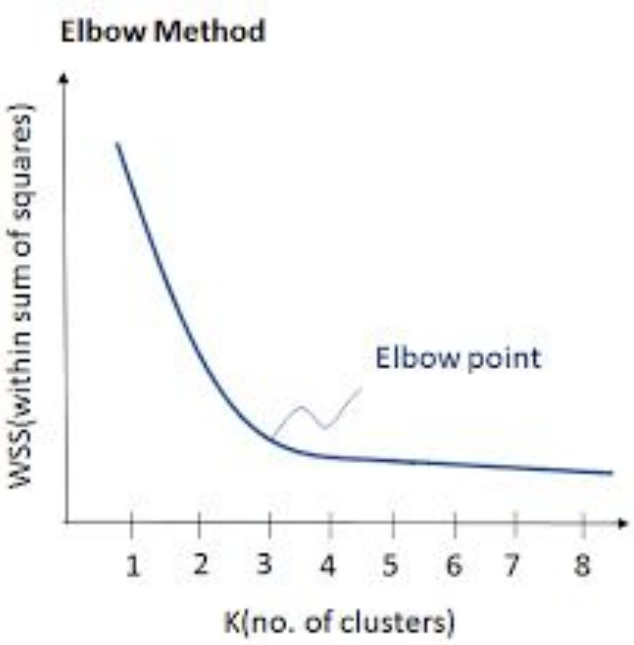
- Subject-matter knowledge (E.g. There are most likely 5 groups)
- Convenience (It is convenient to market to 3 or 4 groups)
- Business Constraints (6 different products need six segments)
- Arbitrary (Always pick 20)

However, we can apply data-driven approaches to help guide the selection of K

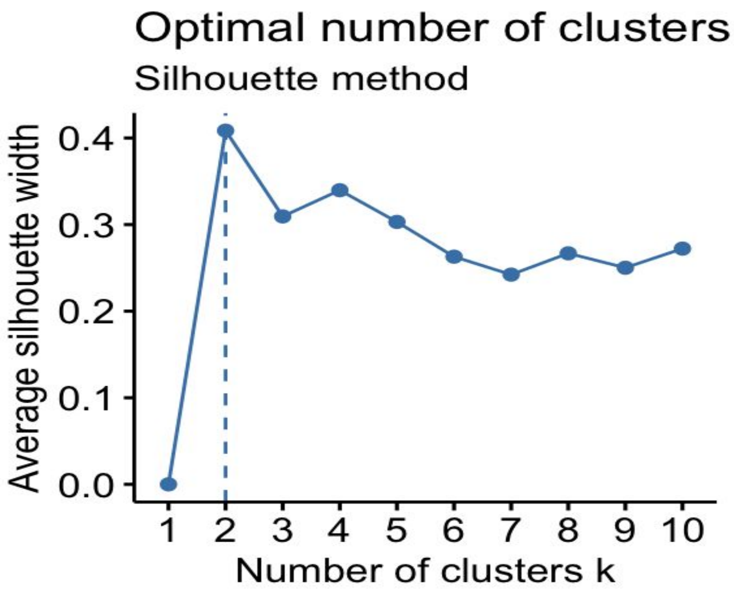
Evaluation of K

Evaluation of K

Total WSS



Silhouette Score



Evaluation of K: Discussion

Total WSS

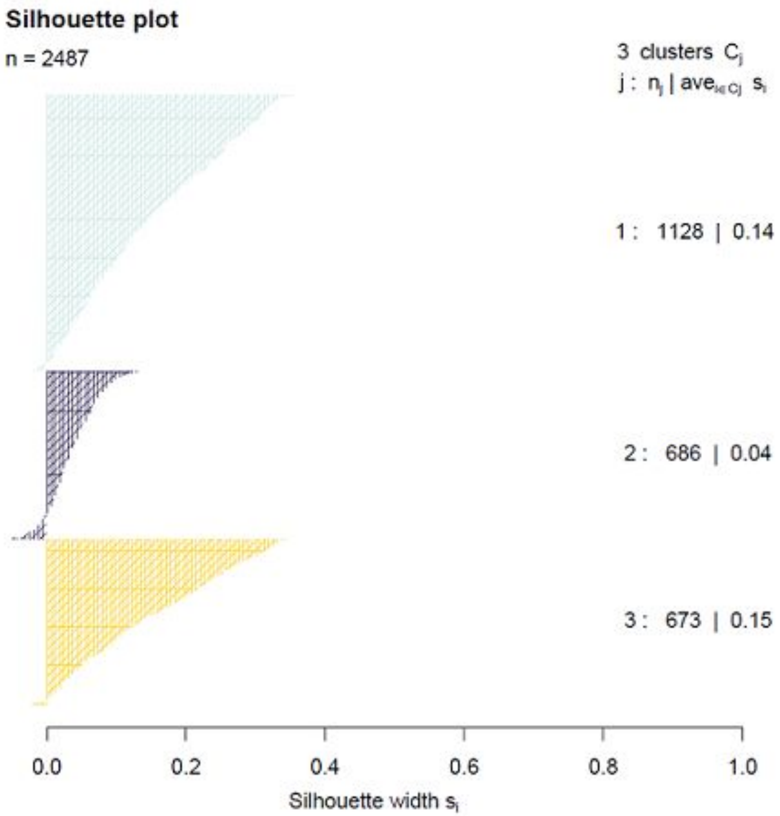
- For each record, calculate the **euclidean distance** from it's assigned cluster center/centroid
- This distance is totaled for the cluster
  - **inertia**
- We can evaluate cluster and item associations
  - **Silhouette scores (fit)**
  - **Silhouette samples (cluster/row)**

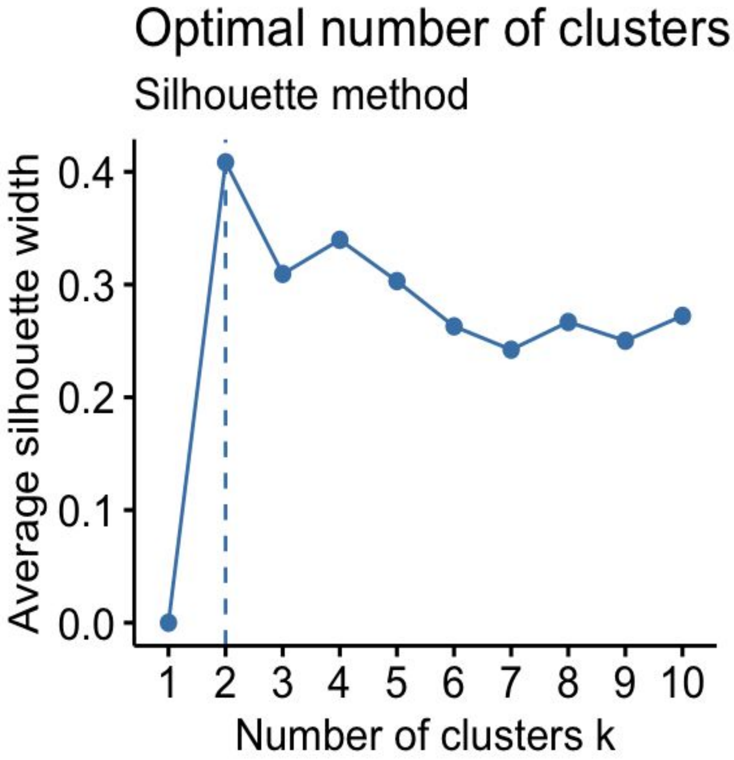
We want to **minimize** inertia, and maximize silhouette via selection of *k*

Silhouette Score

- *Silhouette refers to a method of interpretation and validation of consistency within clusters of data*[1]
- Ranges from -1 to 1, and is a measure of how similar a point is to the points in its assigned cluster
- Generalized process
  - Metric is done for every data point
  - Average distance from itself and every points in its cluster
  - Average distance from itself and the points in the closest neighboring cluster
- Want high values of 1
- Many negative values suggest an improper cluster solution

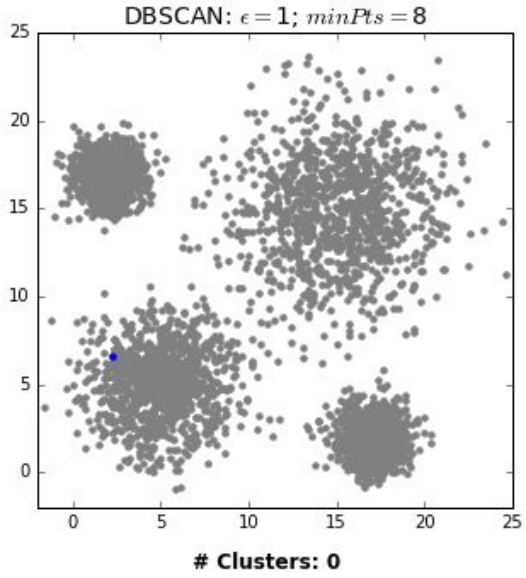
Silhouette Plots

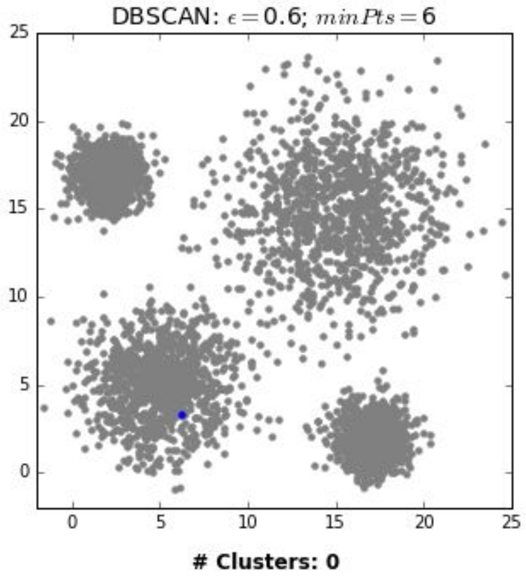


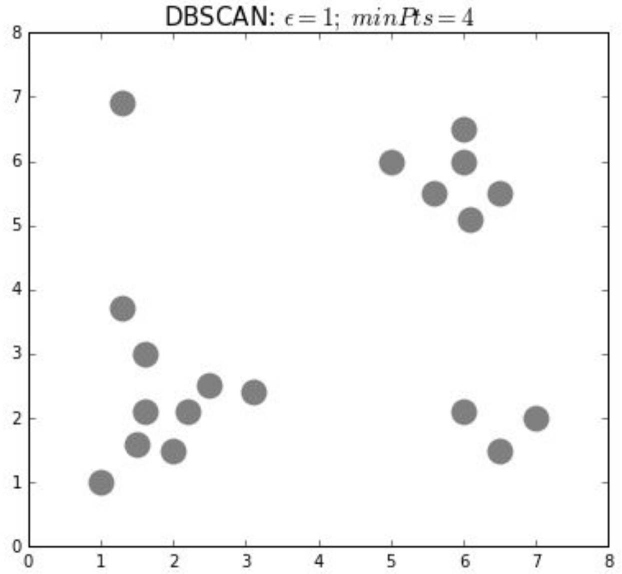


Some other approaches to consider

DBSCAN – A Visual Tutorial







Overview

- Density-based approach and does not require that we set k
- Intuition: find dense collections of points satisfy settings, everything else are outliers
- For a given point, evaluate neighbors to determine if a cluster can be formed relative its "neighborhood"
- If a point can't find any other neighbors, process moves to another point in hopes of finding sufficient neighboring points
- All points are visited and evaluated, or Scanned

Settings:

- Min # of points required to form a cluster
  - Rule of thumb is greater than, or equal to # of features considered + 1
- Size of the neighborhood (points need to fall inside this distance)

Hands on in python