Clustering - Kmeans

- **Clustering**: a grouping / assignment of data points such that data points in the same group / cluster are: • Similar to each other • dissimilar to objects in other groups

- Applications:
  - outlier detection / anomaly detection
    - data cleaning/processing, credit card fraud, spam filter, etc.
  - Feature extraction
  - Filling Gaps in data
    - infer probable vals for gaps in data
- Clusters can be ambiguous > could have very diff. clusterings depending on parameters
- Types of Clustering:
  - Partitional > each object belongs to exactly one cluster
  - Hierarchichal > a set of nested clusters organized in a tree
  - Density Based > defined based on local density of pts
  - Soft Clustering > each pt is assigned to every Cluster w/ a certain probability
- Partitional Clustering:
  - goal is to minimize inner cluster distance (how close pts are in a cluster), such that each data pt belongs to exactly one cluster & maximize the distance btwn other clusters. This is Hard Clustering
- Cost Function: $\sum_{i}^{k}\sum_{x\in_i} d(x,\mu_i)^2$
  - a way to evaluate & compare solutions
  - want to find an algorithm to reduce cost
- Kmeans: given $\mathbf{X}=\{x_1,\dots,x_n\}$ dataset $\mathbf{d}$, the euclidean dist., & k. Find k centers $\{\mu_1,\dots,\mu_k\}$ that minimize the cost function $\sum_{i}^{k}\sum_{x=C_i} d(x,\mu_i)^2$
  - when k=1 (1D) or k=2 (2D) its easier & when it goes past 2D it becomes NP-Hard
  - Kmeans (Lloyd's Algorithm)
    1. Randomly pick k centers $\{\mu_1,\dots,\mu_k\}$
    2. Assign each pt to its closest center
    3. Compute new centers as the means of each cluster
    4. Repeat 2&3 until convergence
    - minimizes w/in cluster sum of squares
  - Kmeans weakness:
    - must specify k (need predetermined # of clusters); Sensitive to outliers (they can distort centroids)
    - doesn't work well w/ non-spherical or overlapping shapes → assumes convex