Quick and Dirty Machine Learning

K-Means Algorithm

Konstantin Itskov

The Plan!

- Setup the development environment.
- Theory behind K-means
- K-means implementation

Theory

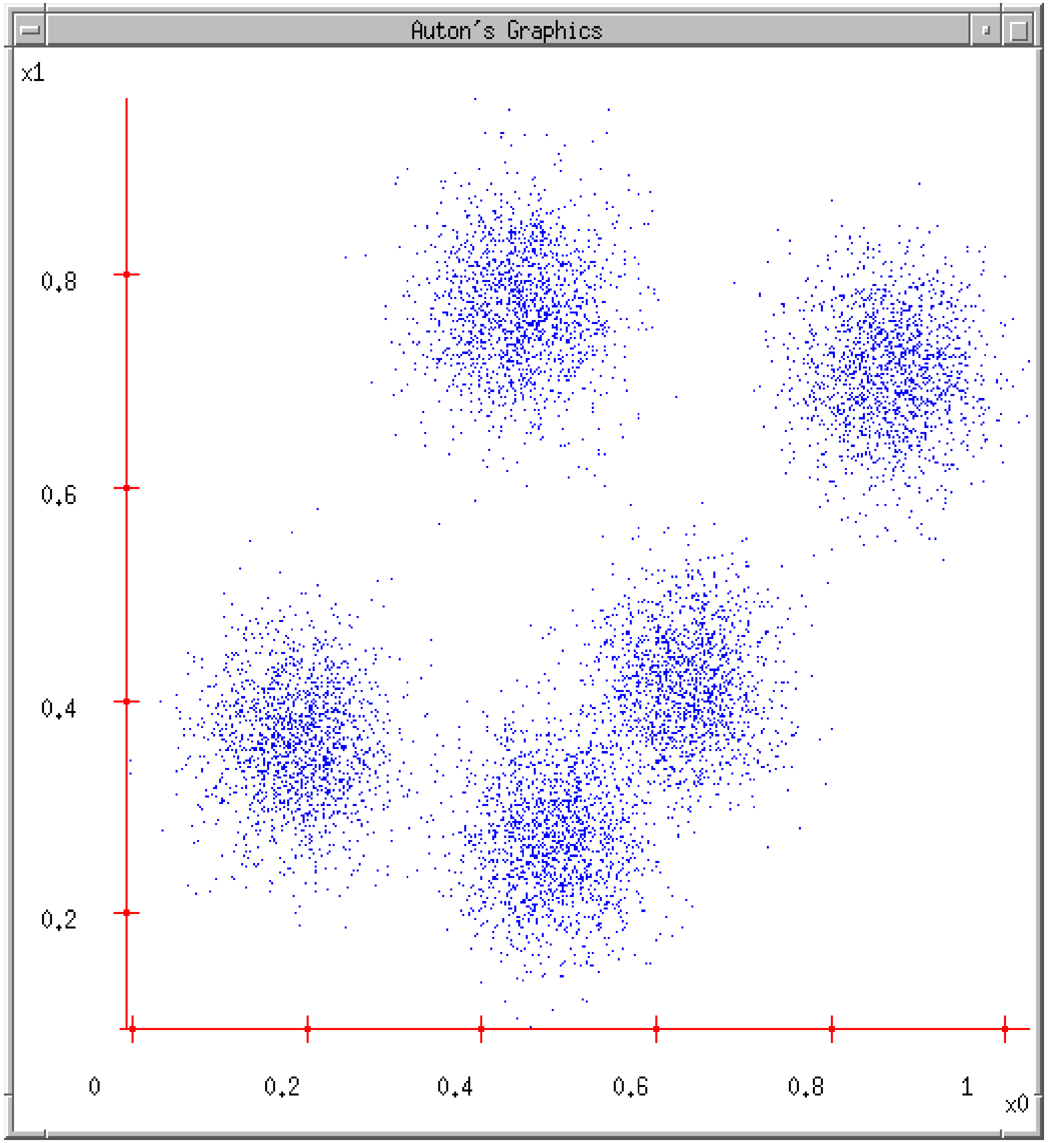
Definition
Given a set of observations $(x_1, x_2, \ldots, x_m)$ where each observation is a d-dimensional real vector $x_i \in \mathbb{R}^d$, k-means clustering aims to partition the m observations into k clusters by minimizing the sum of square distance within each cluster.

Algorithm
1. Initialize cluster centroids $\mu_1, \mu_2, \ldots, \mu_k$ for $\mu_i \in \mathbb{R}^d$ randomly.
2. Repeat until convergence:

$$label_i := \arg\min_j \|x_i - \mu_j\|^2$$

$$\mu_j := \frac{\sum_{i=1}^{m} 1\{label_i = j\} x_i}{\sum_{i=1}^{m} 1\{label_i = j\}}$$

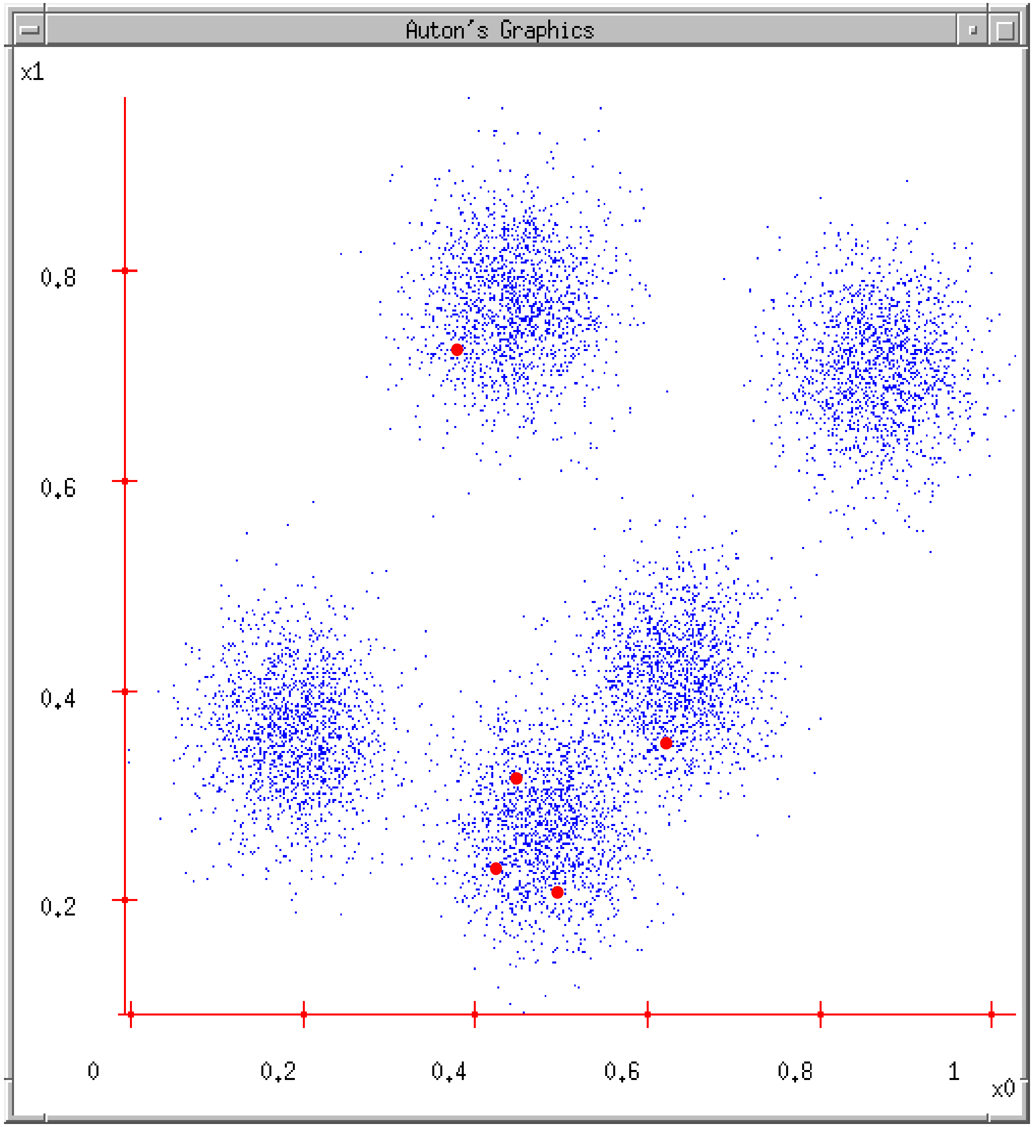
K-means Demo



1. User set up the number of clusters they'd like. *(e.g. k=5)*

https://courses.cs.washington.edu/courses/csep546/07sp/

K-means Demo

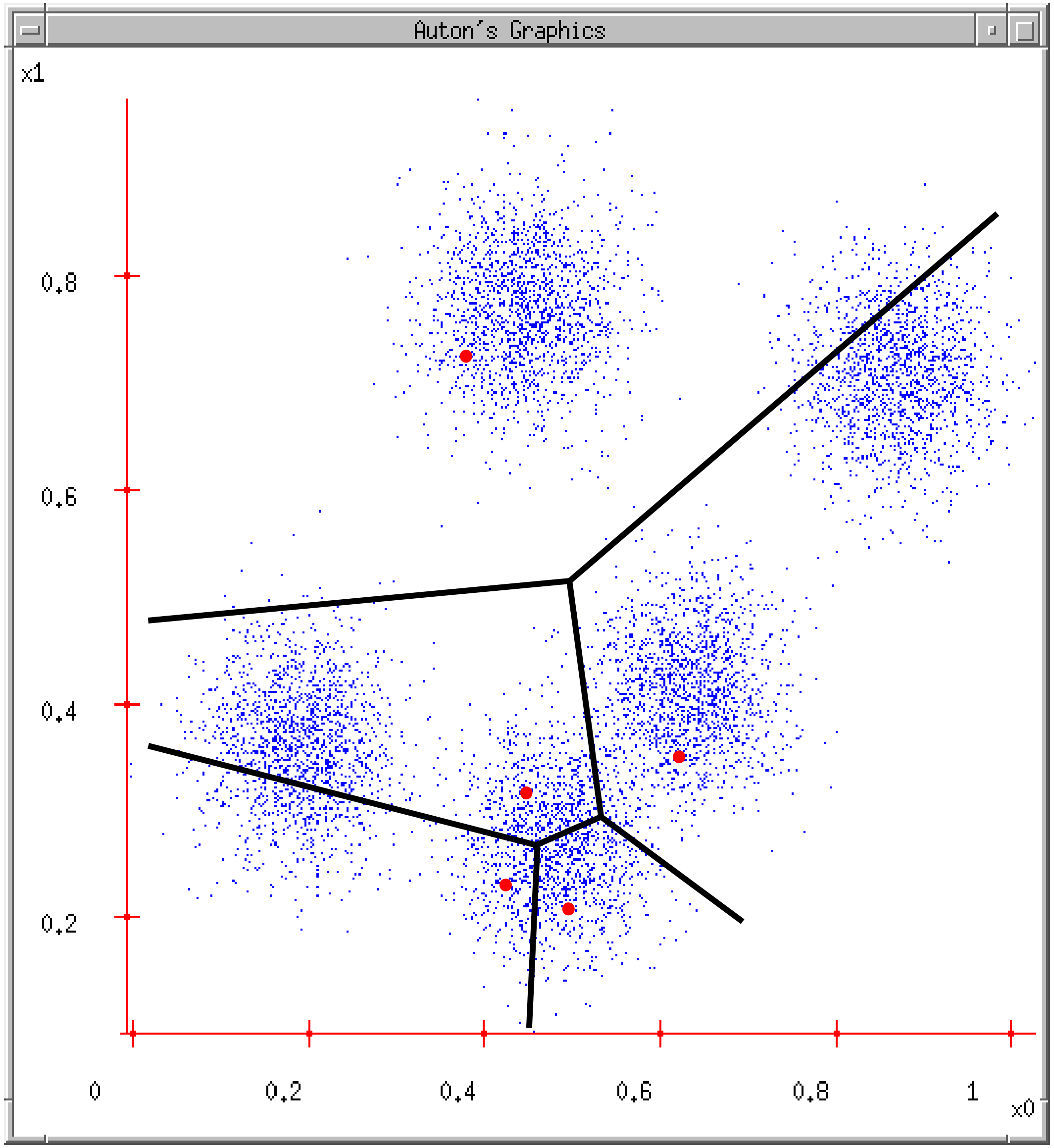


1. User set up the number of clusters they'd like. *(e.g. K=5)*

2. Randomly guess K cluster Center locations

https://courses.cs.washington.edu/courses/csep546/07sp/

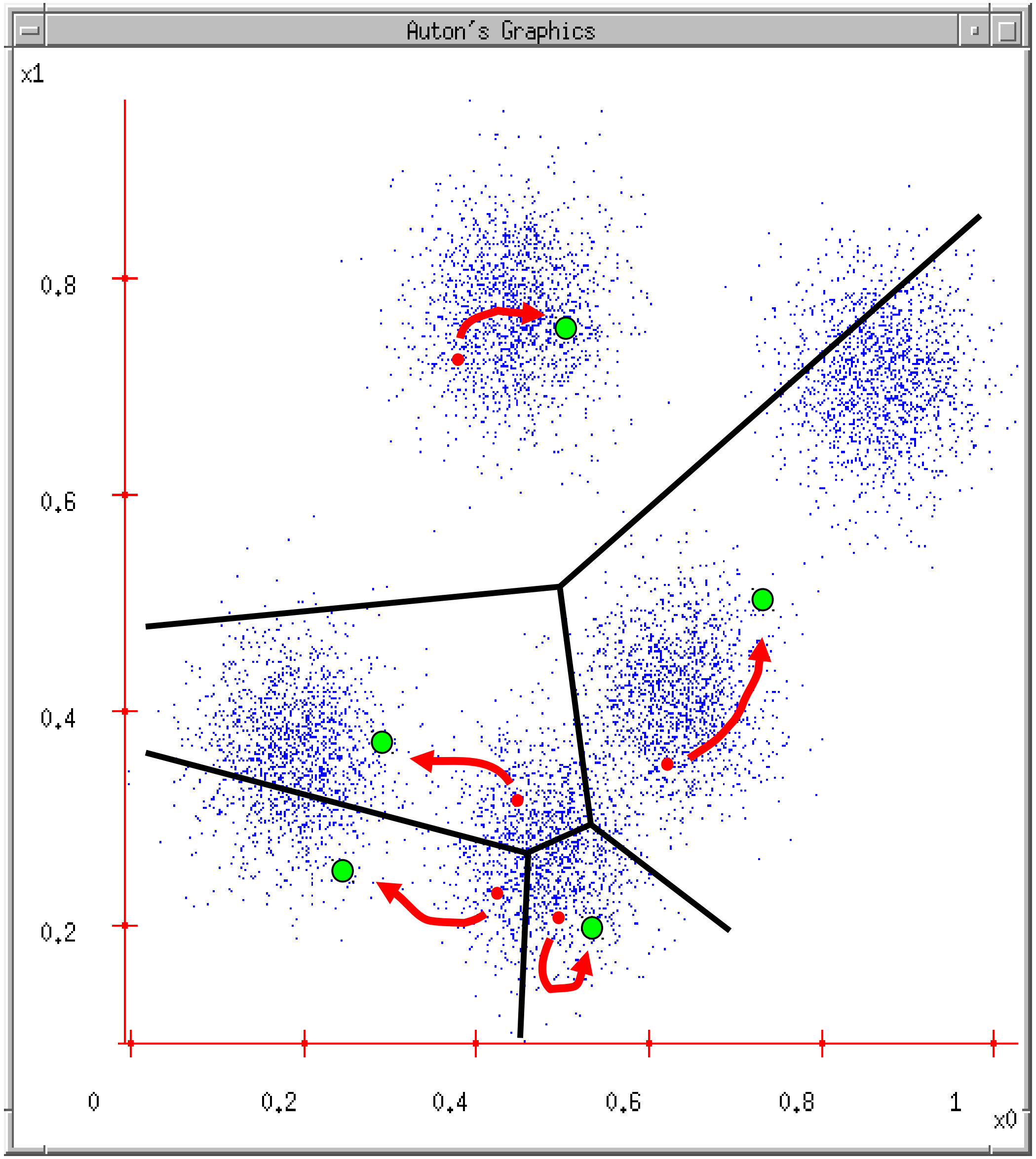
K-means Demo



1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess *K* cluster Center locations
3. Each data point finds out which Center it's closest to. (Thus each Center "owns" a set of data points)

https://courses.cs.washington.edu/courses/csep546/07sp/

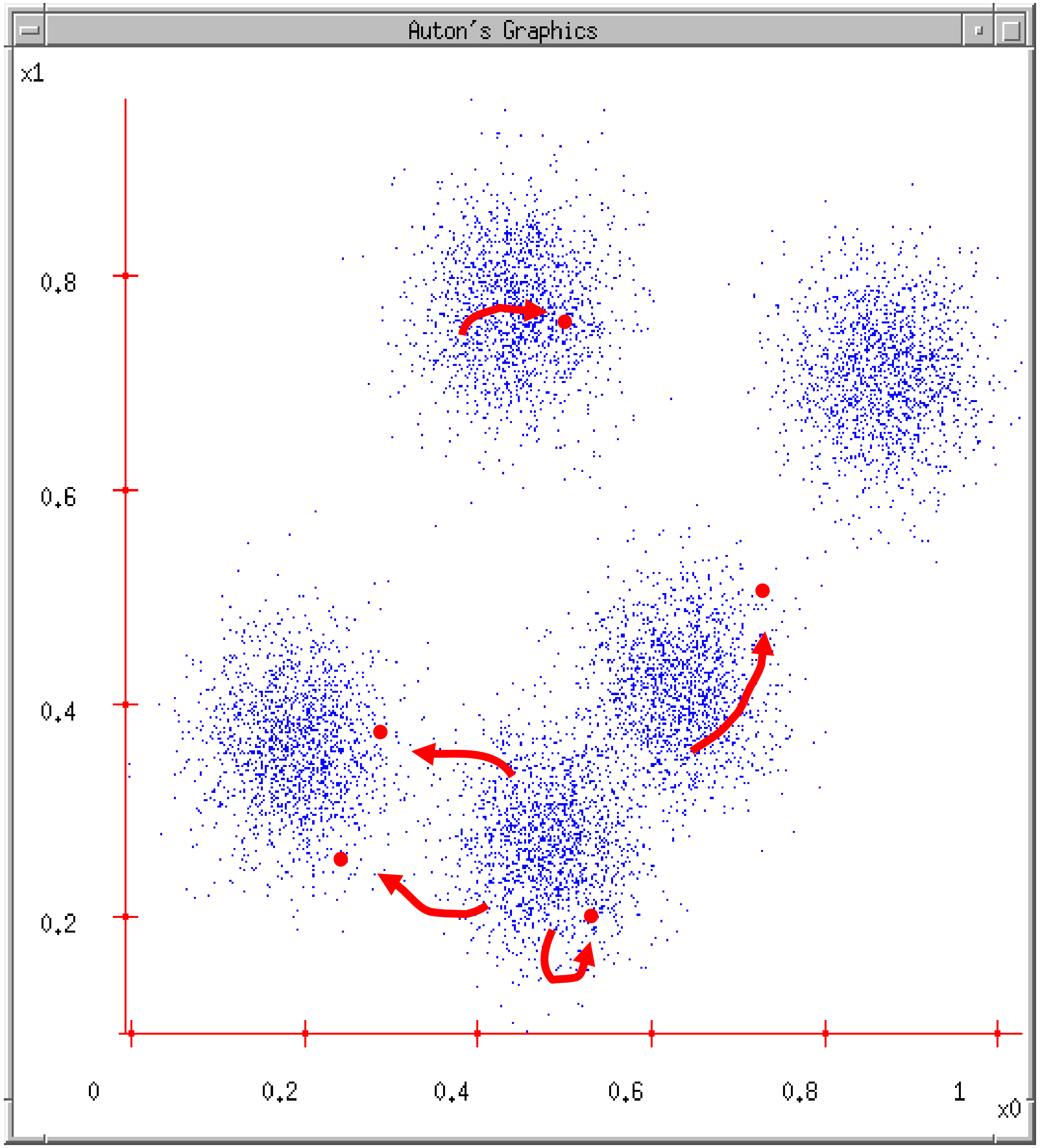
K-means Demo



1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess *K* cluster centre locations
3. Each data point finds out which centre it's closest to. (Thus each Center "owns" a set of data points)
4. Each centre finds the centroid of the points it owns

https://courses.cs.washington.edu/courses/csep546/07sp/

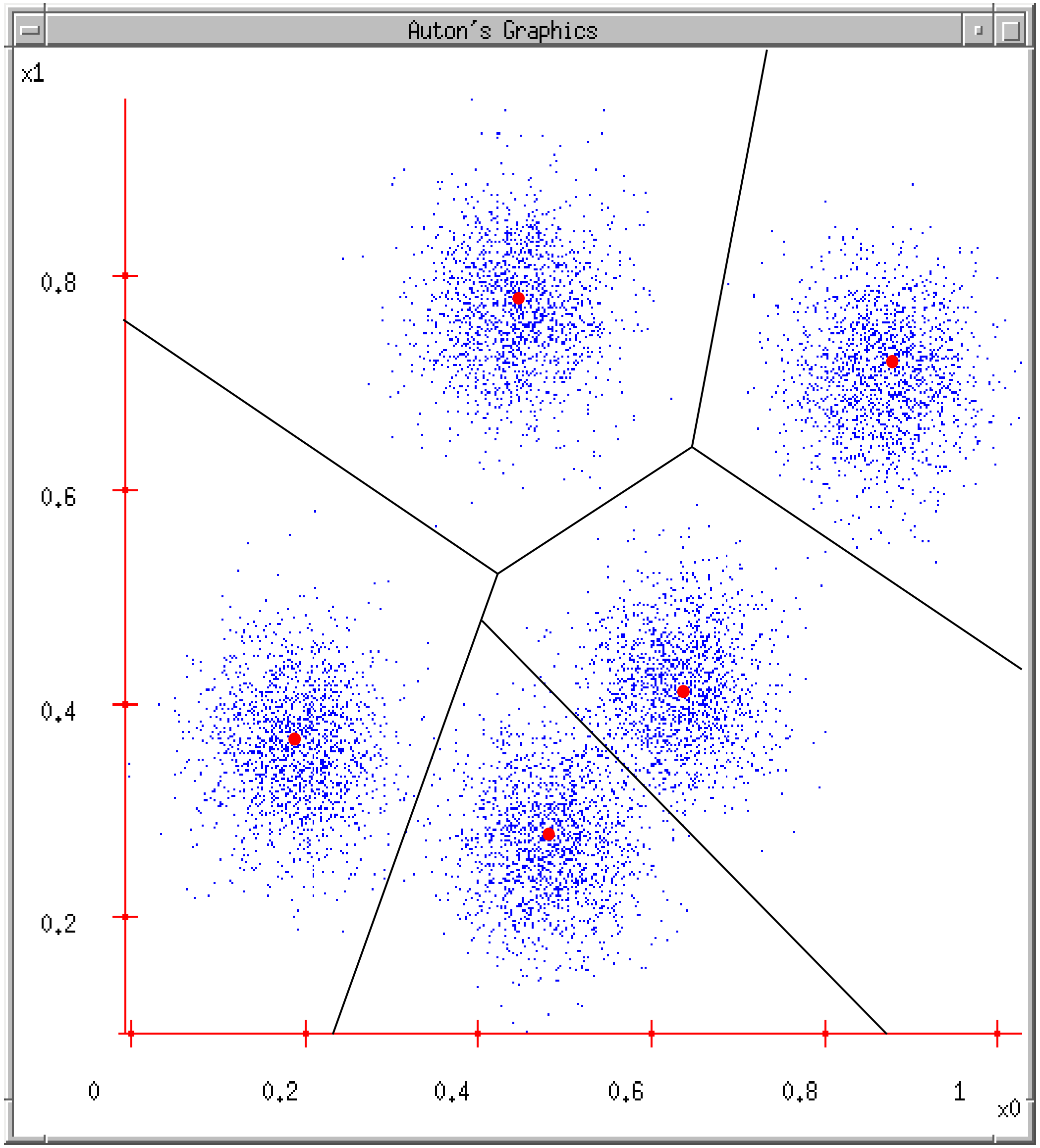
K-means Demo



1. User set up the number of clusters they'd like. (e.g. *K=5*)
2. Randomly guess *K* cluster centre locations
3. Each data point finds out which centre it's closest to. (Thus each centre "owns" a set of data points)
4. Each centre finds the centroid of the points it owns
5. …and jumps there

https://courses.cs.washington.edu/courses/csep546/07sp/

K-means Demo



1. User set up the number of clusters they'd like. (e.g. *K=5*)
2. Randomly guess *K* cluster centre locations
3. Each data point finds out which centre it's closest to. (Thus each centre "owns" a set of data points)
4. Each centre finds the centroid of the points it owns
5. …and jumps there
6. …Repeat until terminated!

https://courses.cs.washington.edu/courses/csep546/07sp/