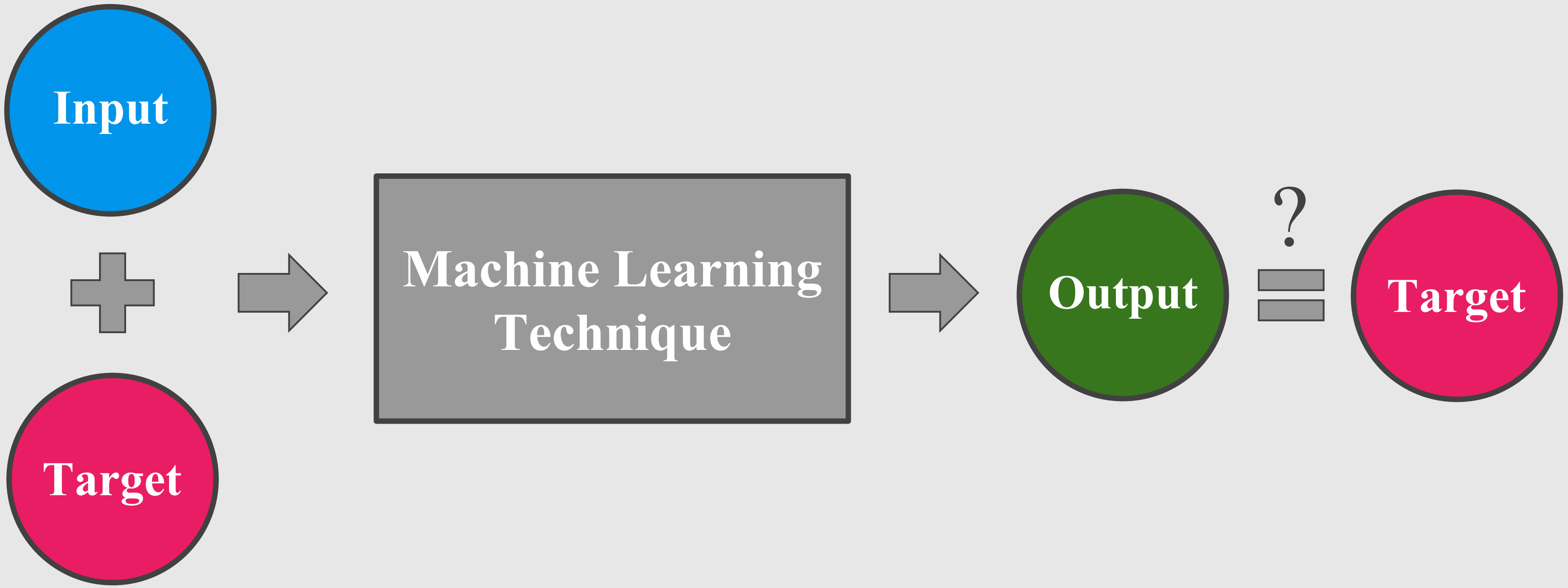
Machine Learning and Pattern Recognition
A High Level Overview

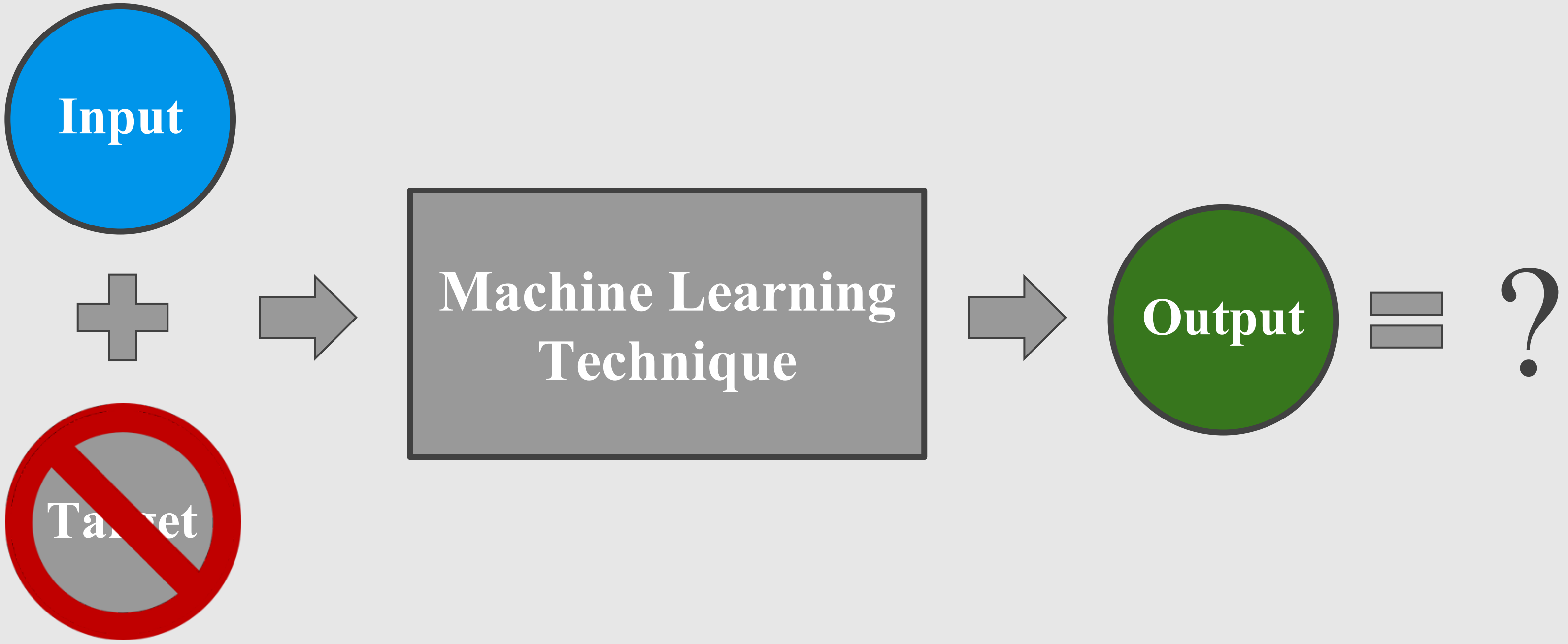
Prof. Anderson Rocha
(Main bulk of slides kindly provided by **Prof. Sandra Avila**)
Institute of Computing (IC/Unicamp)

MC886/MO444

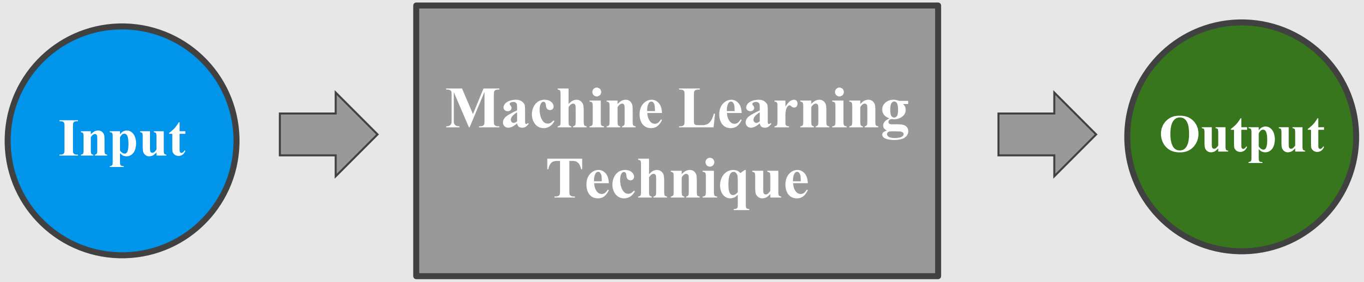
Supervised Learning



Unsupervised Learning



Unsupervised Learning



The goal of unsupervised learning is **to find patterns** in the data, and build new and useful representations of it.

Today’s Agenda

- Clustering
  - k-Means Algorithm
  - Optimization Objective
  - Random Initialization
  - Choosing the Number of Clusters
  - k-Means Variations
  - Evaluation Performance Clustering

Clustering

k-Means Algorithm

Did anyone say pizza?

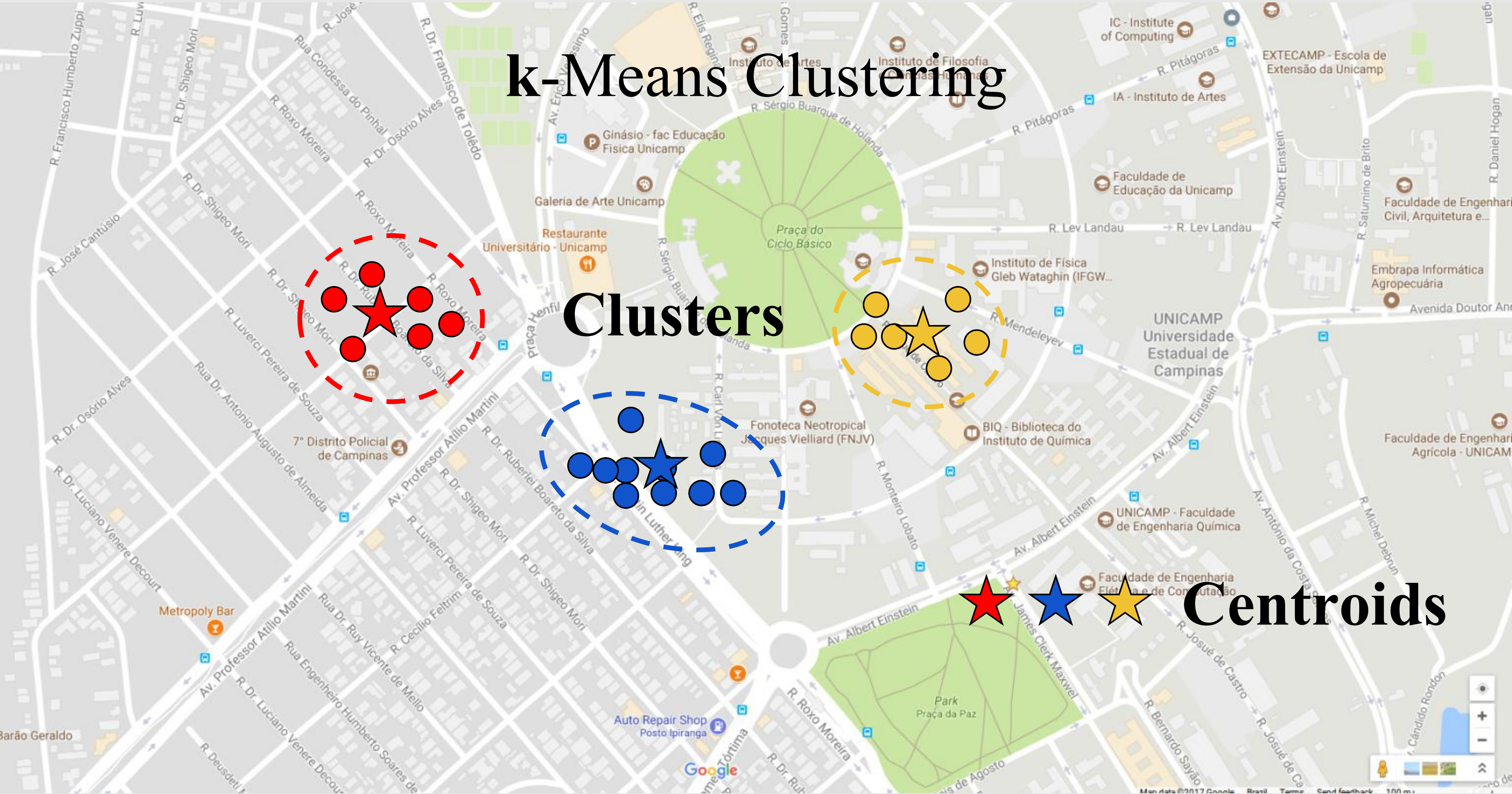
Praça do Ciclo Básico
Instituto de Artes
Instituto de Filosofia e Ciências Humanas
IC - Institute of Computing
EXTECAMP - Escola de Extensão da Unicamp
IA - Instituto de Artes
Ginásio - fac Educação Física Unicamp
Galeria de Arte Unicamp
Restaurante Universitário - Unicamp
Instituto de Física Gleb Wataghin (IFGW...
Fonoteca Neotropical Jacques Vielliard (FNJV)
BIQ - Biblioteca do Instituto de Química
7° Distrito Policial de Campinas
UNICAMP - Faculdade de Engenharia Química
Faculdade de Engenharia Elétrica e de Computação
Metropoly Bar
Auto Repair Shop Posto Ipiranga
Park Praça da Paz
Av. Albert Einstein
Av. Martin Luther King
Av. Professor Atílio Martini
R. Pitágoras
R. Lev Landau
R. Mendeleyev
Barão Geraldo
Google

https://www.youtube.com/watch?v=IpGxLWOIZy4

Did anyone say pizza?

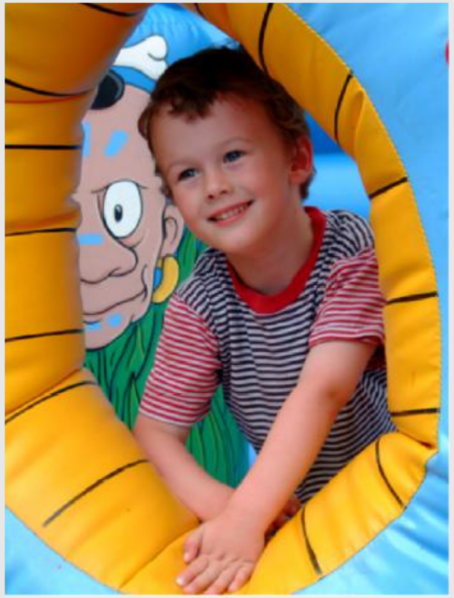
Instituto de Artes
Instituto de Filosofia e Ciências Humanas
IC - Institute of Computing
EXTECAMP - Escola de Extensão da Unicamp
IA - Instituto de Artes
R. Pitágoras
Ginásio - fac Educação Física Unicamp
Galeria de Arte Unicamp
Restaurante Universitário - Unicamp
Praça do Ciclo Básico
R. Sérgio Buarque de Holanda
R. Lev Landau
Instituto de Física Gleb Wataghin (IFGW...
R. Mendeleyev
Fonoteca Neotropical Jacques Vielliard (FNJV)
BIQ - Biblioteca do Instituto de Química
7° Distrito Policial de Campinas
Av. Professor Atílio Martini
UNICAMP - Faculdade de Engenharia Química
Faculdade de Engenharia Elétrica e de Computação
Av. Albert Einstein
Metropoly Bar
Park Praça da Paz
Auto Repair Shop Posto Ipiranga
R. Roxo Moreira
Google

k-Means Clustering

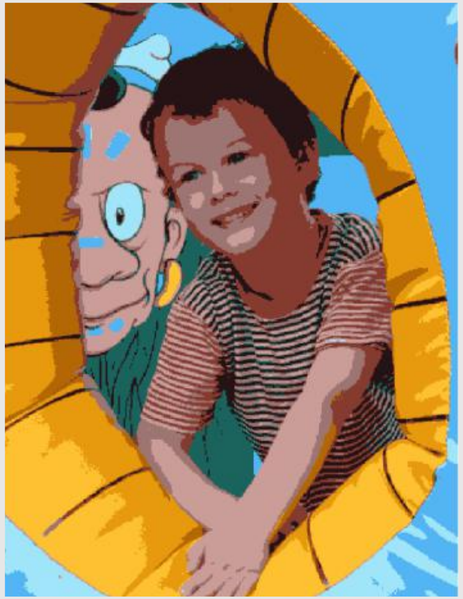


k-Means: Image Segmentation

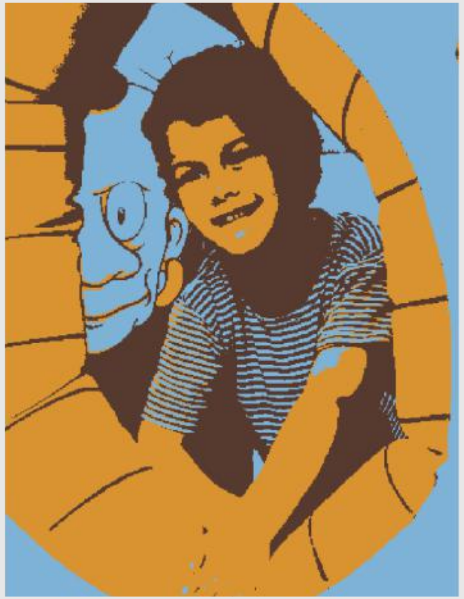
Original



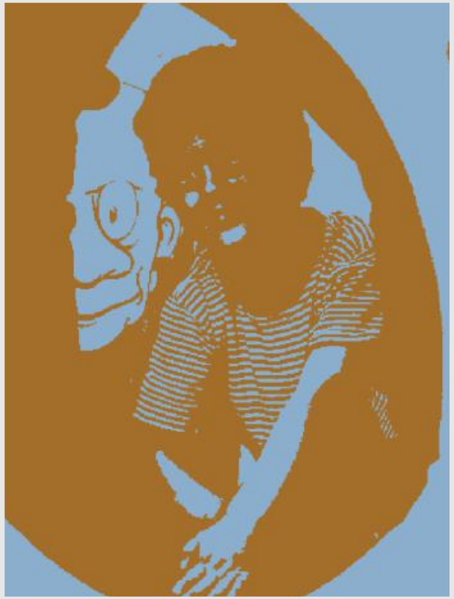
K = 10



K = 3



K = 2

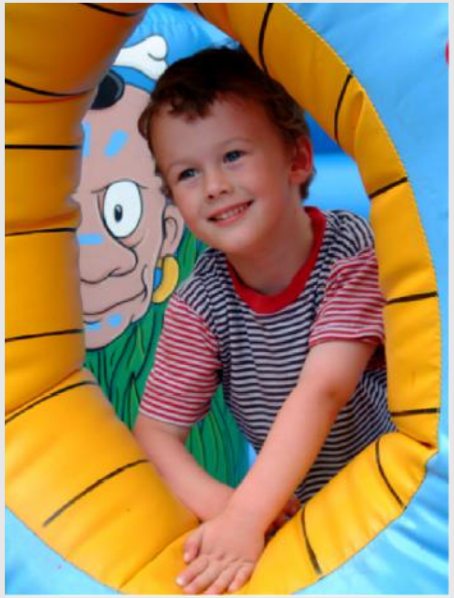


Credit: Christopher Bishop

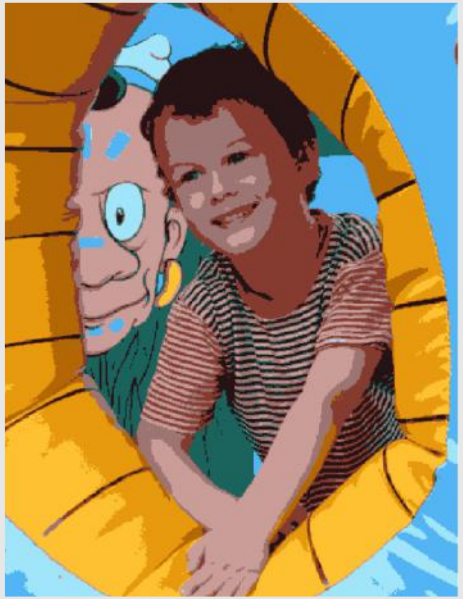
k-Means: Image Segmentation

Color Quantization

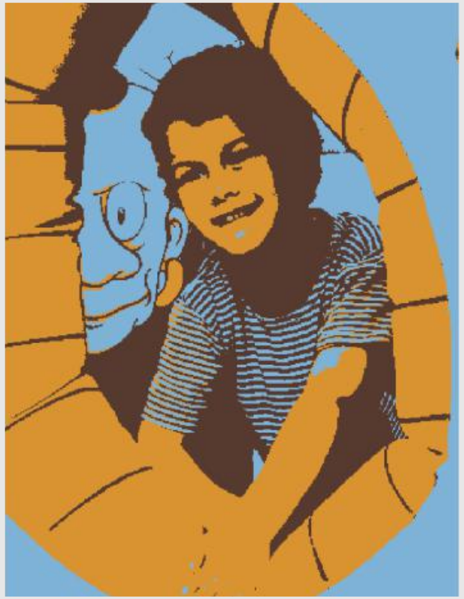
Original



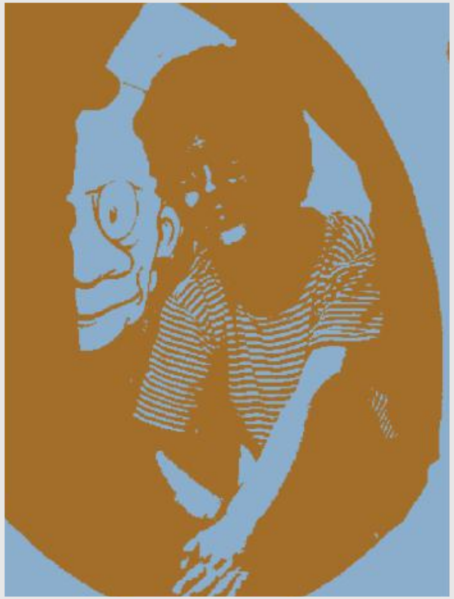
K = 10



K = 3



K = 2



Credit: Christopher Bishop

k-Means Algorithm

1. **Define the *k* centroids**.
2. **Find the closest centroid & update cluster assignments.**
3. **Move the centroids to the center of their clusters.**
4. **Repeat steps 2 and 3** until the centroid stop moving a lot at each iteration.

k-Means Algorithm (Lloyd's Algorithm)

Input:

- ➔ K (number of clusters)
- ➔ Training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

k-Means Algorithm (Lloyd's Algorithm)

```
Randomly initialize $K$ cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$
repeat {
    for $i$ = 1 to $m$
        $c^{(i)}$ := index (from 1 to $K$) of cluster centroid **closest** to $x^{(i)}$
    for $k$ = 1 to $K$
        $\mu_k$ := mean of points assigned to cluster $k$
}
```

$$\min_k ||x^{(i)} - \mu_k||$$

k-Means Algorithm (Lloyd's Algorithm)

```
Randomly initialize $K$ cluster centroids    $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$
repeat {                Cluster assignment step
    for $i = 1$ to $m$
        $c^{(i)}$ := index (from 1 to $K$) of cluster centroid **closest** to    $x^{(i)}$
    for $k = 1$ to $K$
        $\mu_k$ := mean of points assigned to cluster $k$
}  Move (Update) centroid step
```

k-Means Algorithm (Complexity)

- Relatively efficient: **O(*Kmnt*)**
  - K = #clusters
  - m = #vectors (samples)
  - n = #dimension of vectors
  - t = #iterations
  - $n \ll m$

Clustering
Optimization Objective

k-Means Optimization Objective

$c^{(i)}$ = index of cluster (from 1 to K) to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid k

k-Means Optimization Objective

$c^{(i)}$ = index of cluster (from 1 to K) to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid k

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned

Optimization objective:

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} \| x^{(i)} - \mu_{c^{(i)}} \|$$

$$\min_{\substack{c^{(1)}, \ldots, c^{(m)} \\ \mu_1, \ldots, \mu_K}} J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$$

k-Means Optimization Objective

```
Randomly initialize $K$ cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$
repeat {
    for $i = 1$ to $m$
        $c^{(i)}$ := index (from 1 to $K$) of cluster centroid **closest** to $x^{(i)}$
    for $k = 1$ to $K$
        $\mu_k$ := mean of points assigned to cluster $k$
}
```

Clustering

Initialization

Random Initialization

Assign each object to a random cluster &

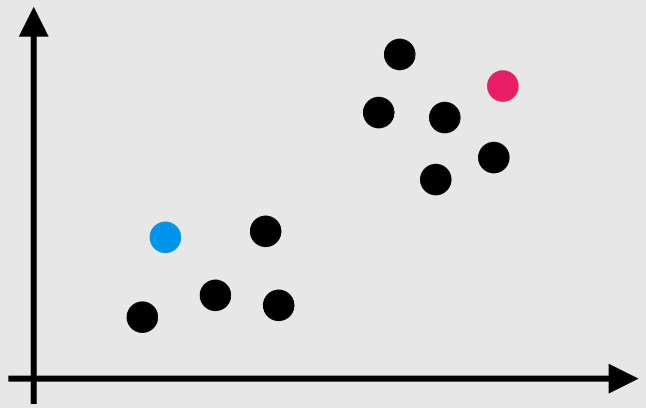
Computes the initial centroid of each cluster.

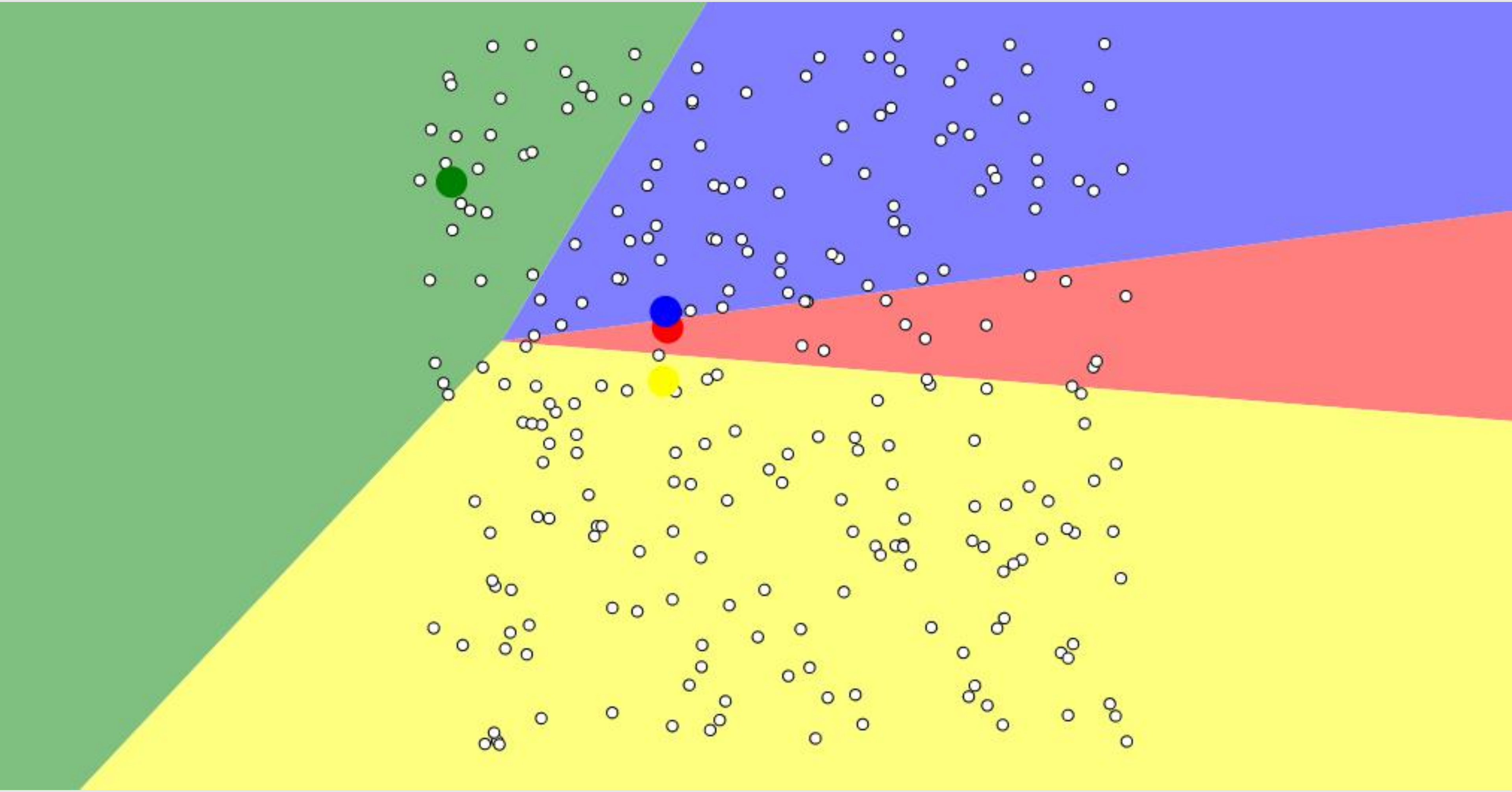
Random Initialization (Forgy, 1965)

Should have $K < m$.

Randomly pick K training examples.

Set $\mu_1, \ldots, \mu_K$ equal to these K examples.





Random Initialization

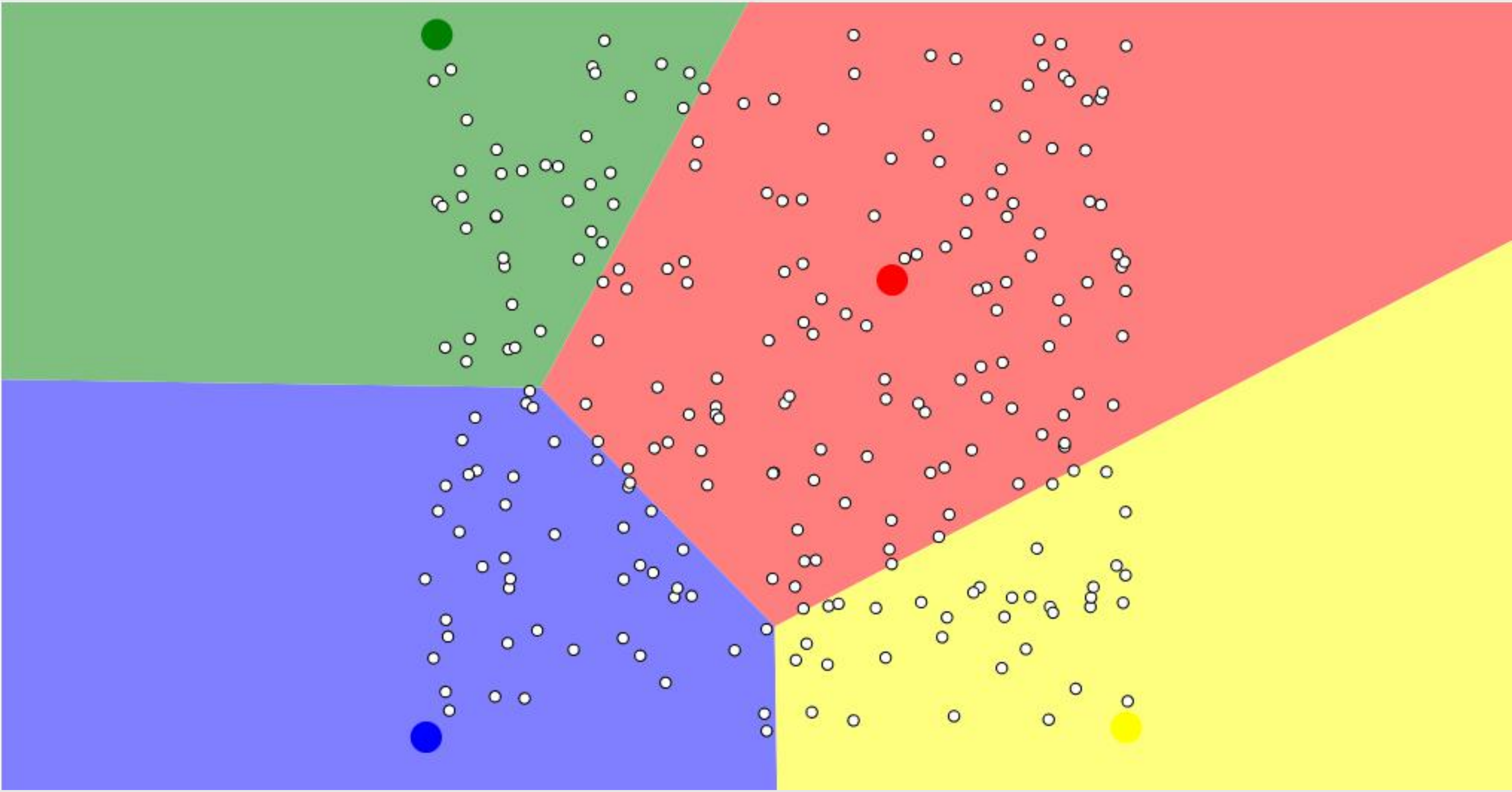
```
for $i = 1$ to 100 {
        Randomly initialize k-Means.
    Run k-means. Get $c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K$
    Compute cost function $J$.
}
```

Pick clustering that gave lowest cost $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$.

Can we do better?

Can we do better?

- One idea for initializing k-Means is to use a farthest-first traversal on the data set, **to pick K points that are far away from each other**.



Can we do better?

- One idea for initializing k-Means is to use a farthest-first traversal on the data set, to pick K points that are far away from each other.

- However, this is **too sensitive to outliers**.

k-Means++ (Arthur & Vassilvitski, 2007)

- It works similarly to the “farthest” heuristic.
- Choose each point at random, with probability proportional to its squared distance from the centers chosen already.

k-Means++ (Arthur & Vassilvitski, 2007)

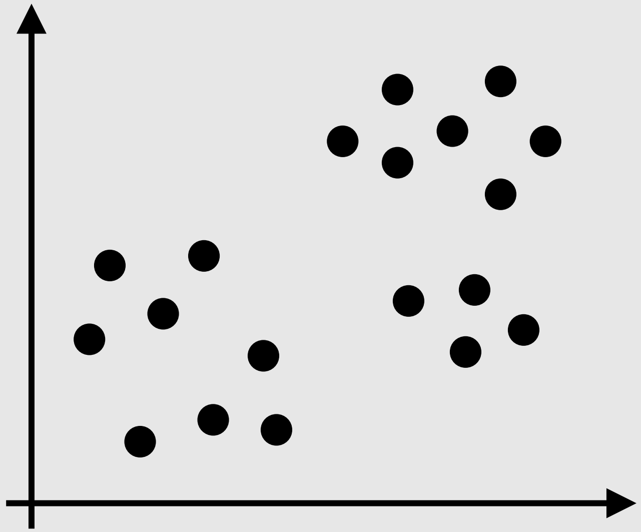
- It works similarly to the “farthest” heuristic.
- Choose each point at random, with probability proportional to its squared distance from the centers chosen already.

scikit-learn (default)

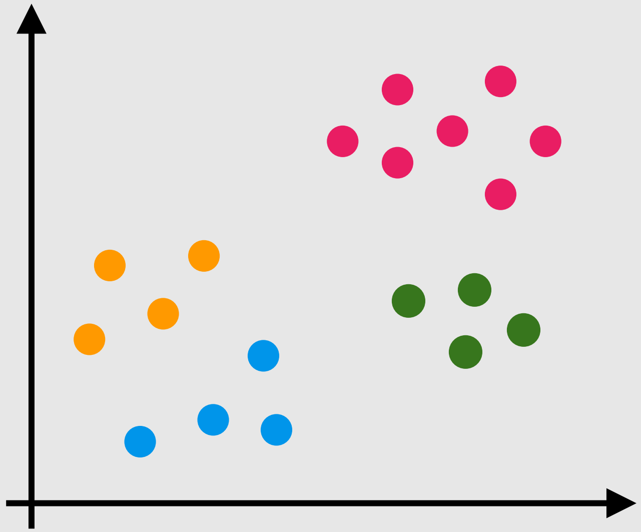
Clustering

Choosing the number of clusters

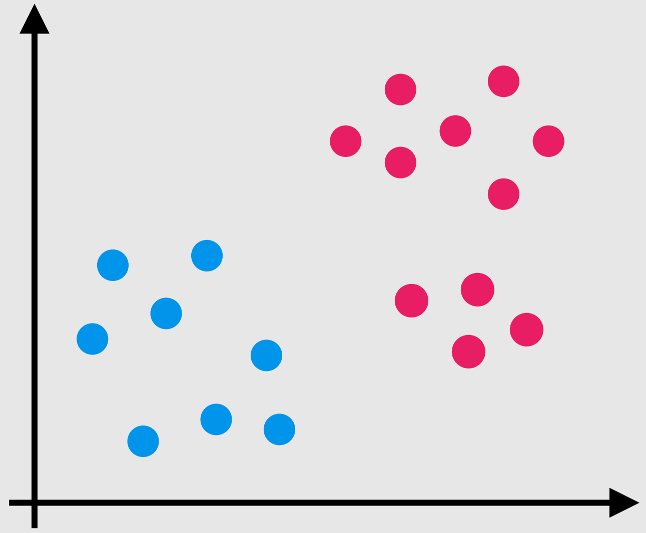
What is the right value of **K**?



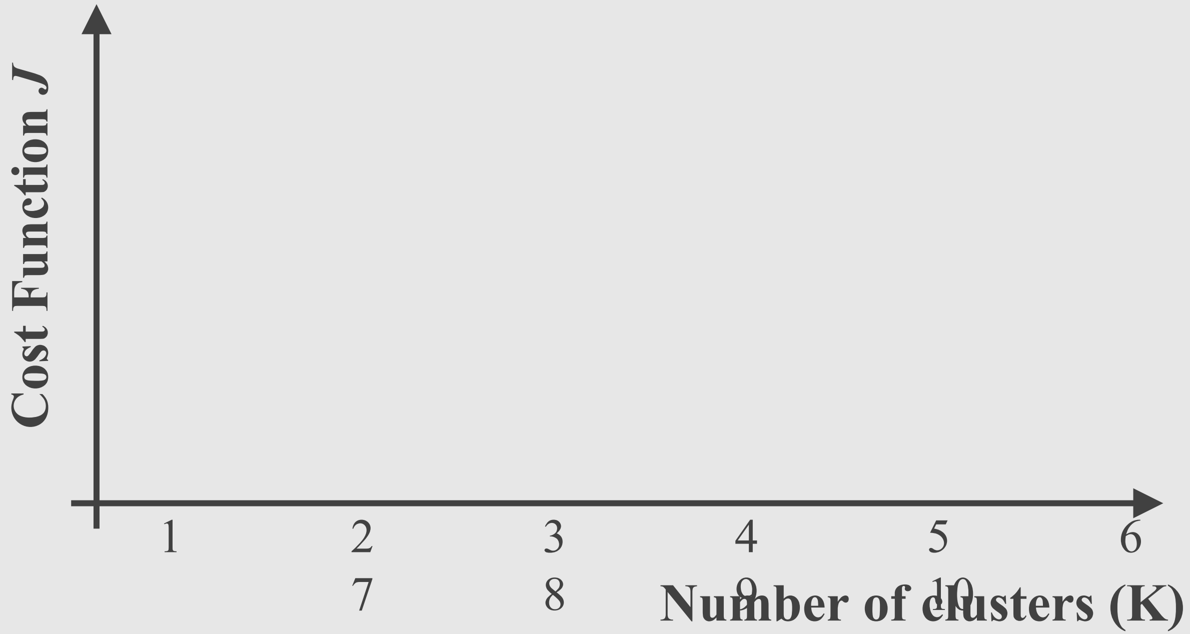
What is the right value of **K**?



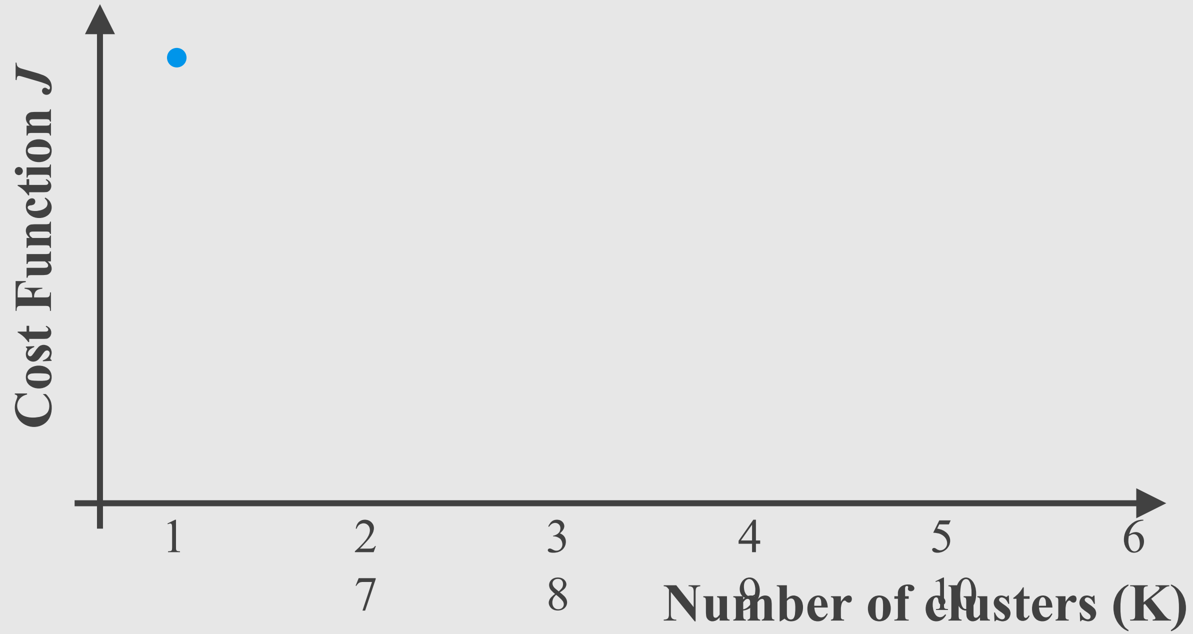
What is the right value of **K**?



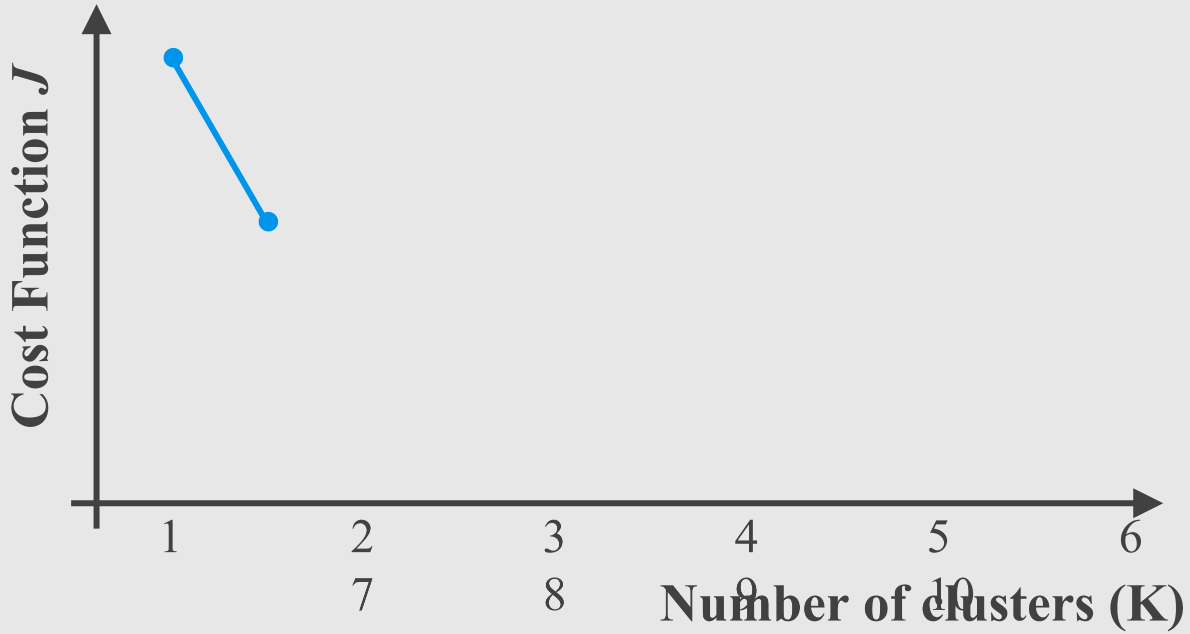
Elbow Method



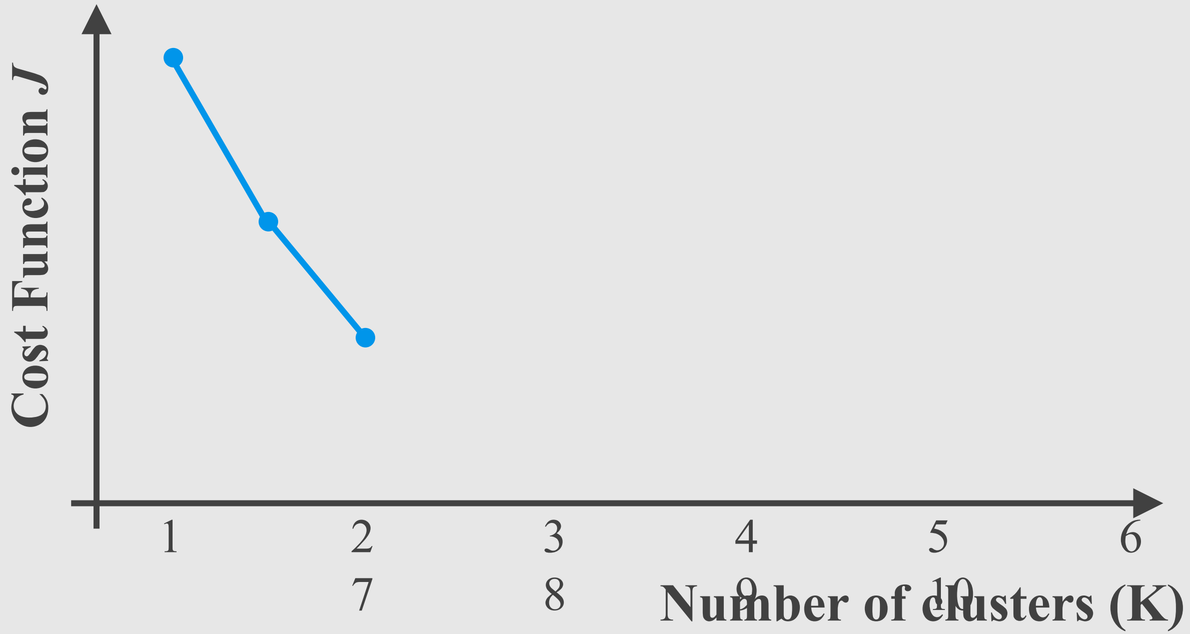
Elbow Method



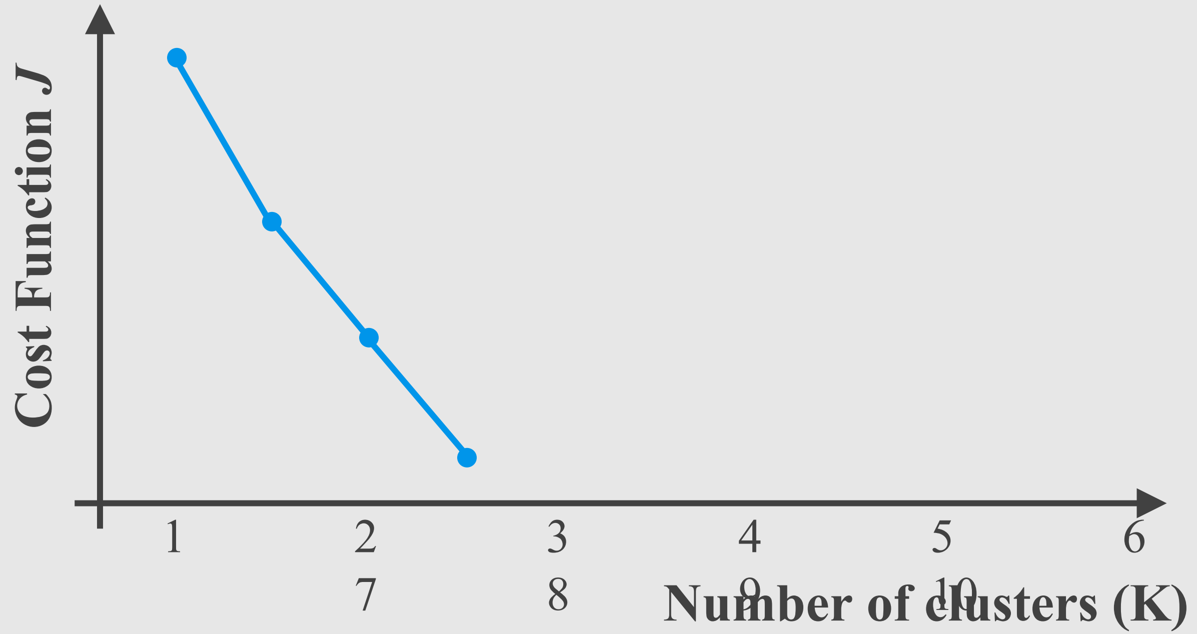
Elbow Method



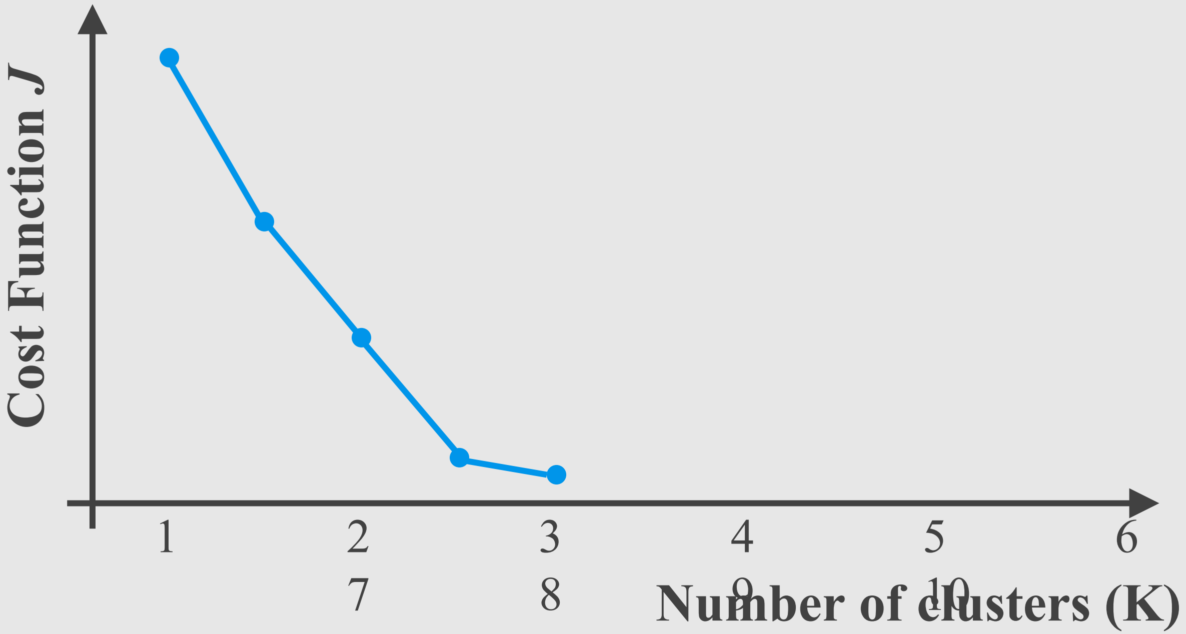
Elbow Method



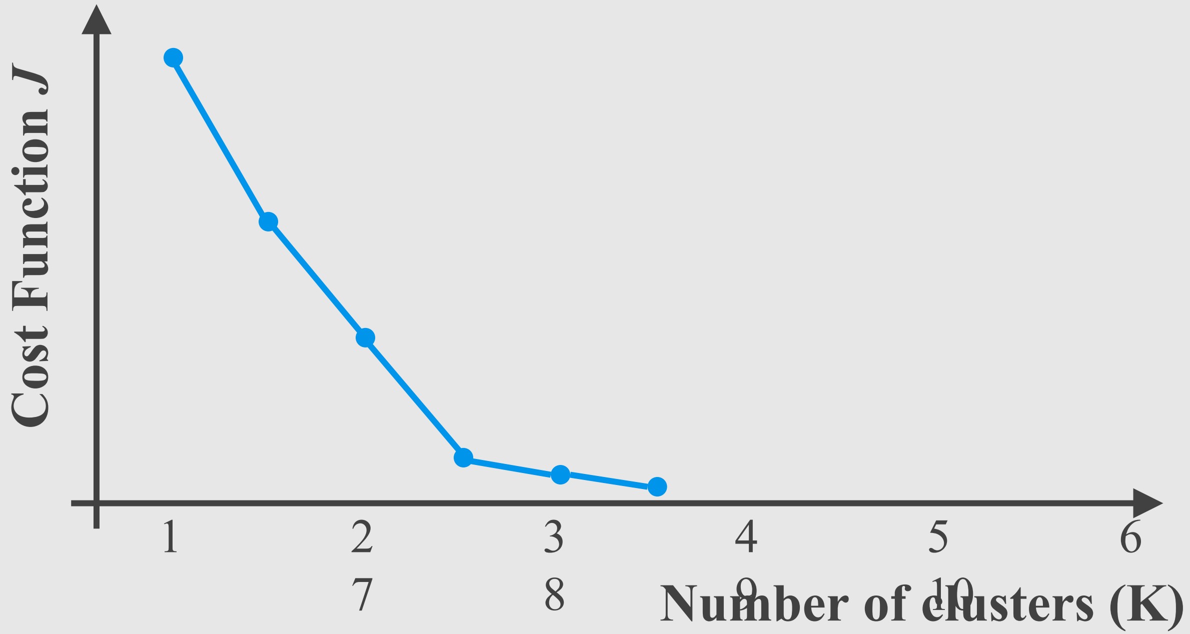
Elbow Method



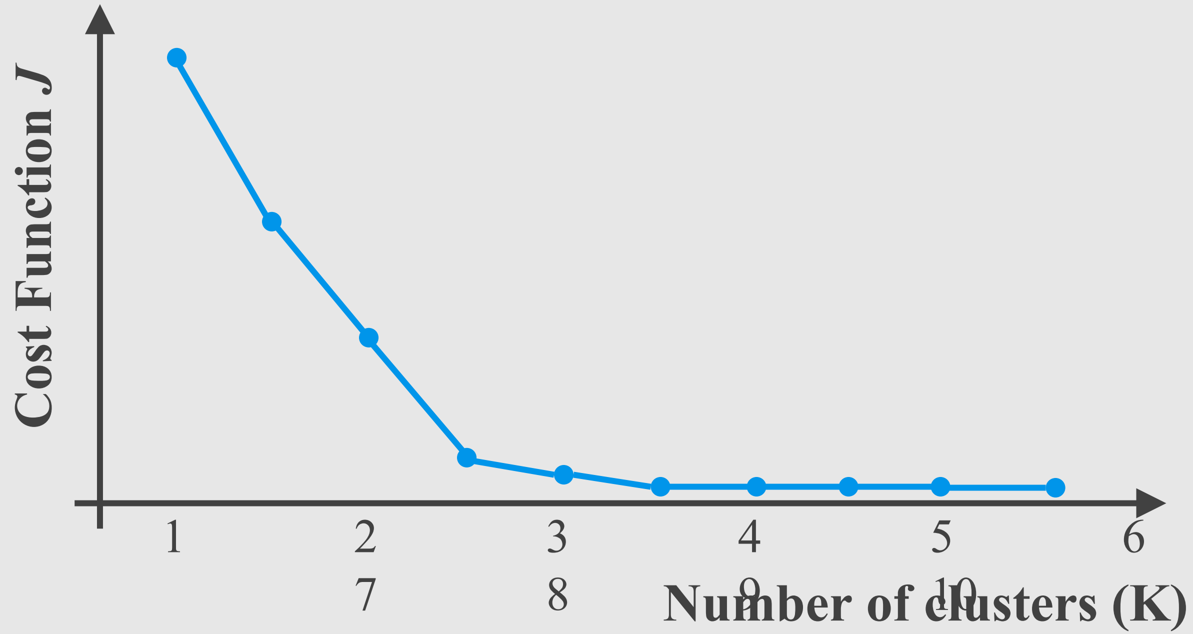
Elbow Method



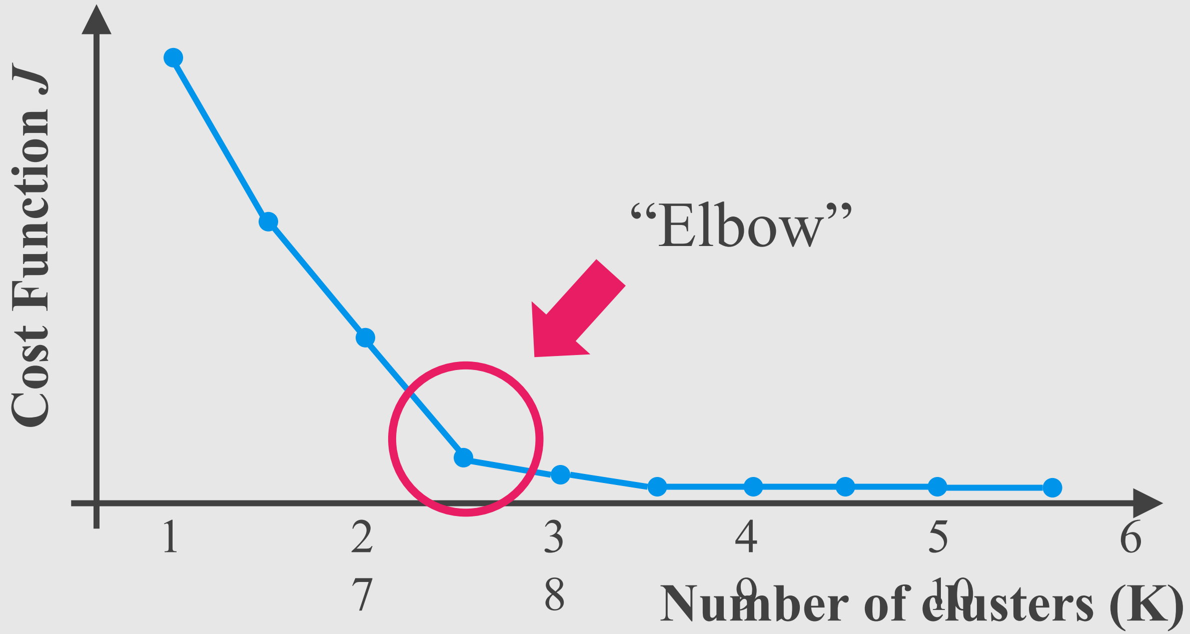
Elbow Method



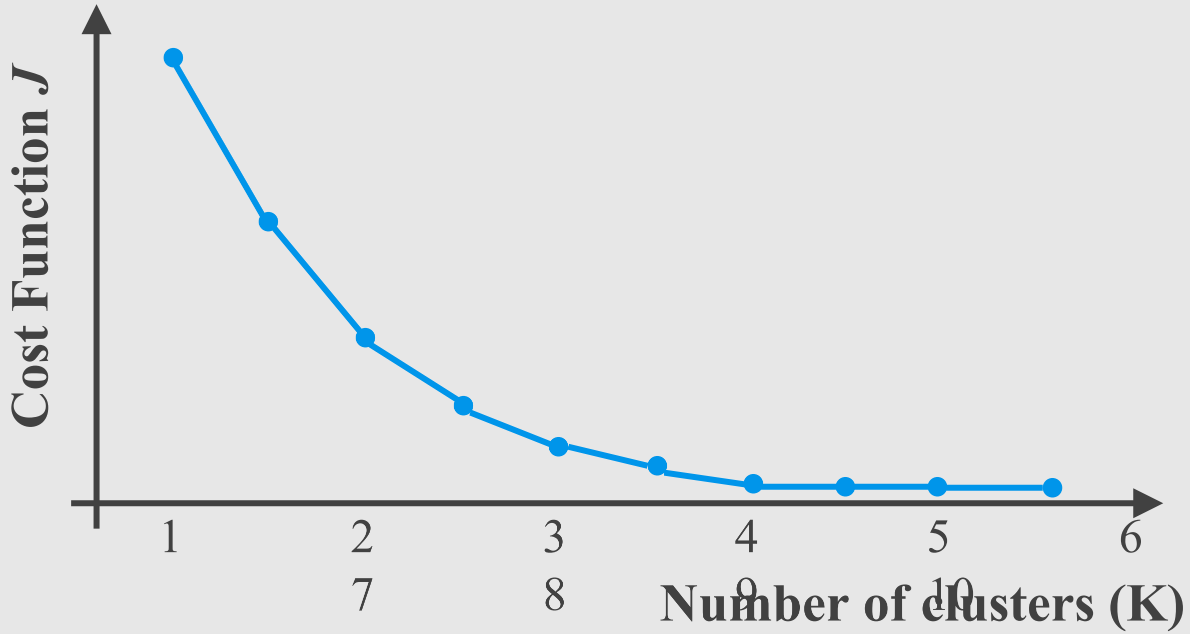
Elbow Method



Elbow Method

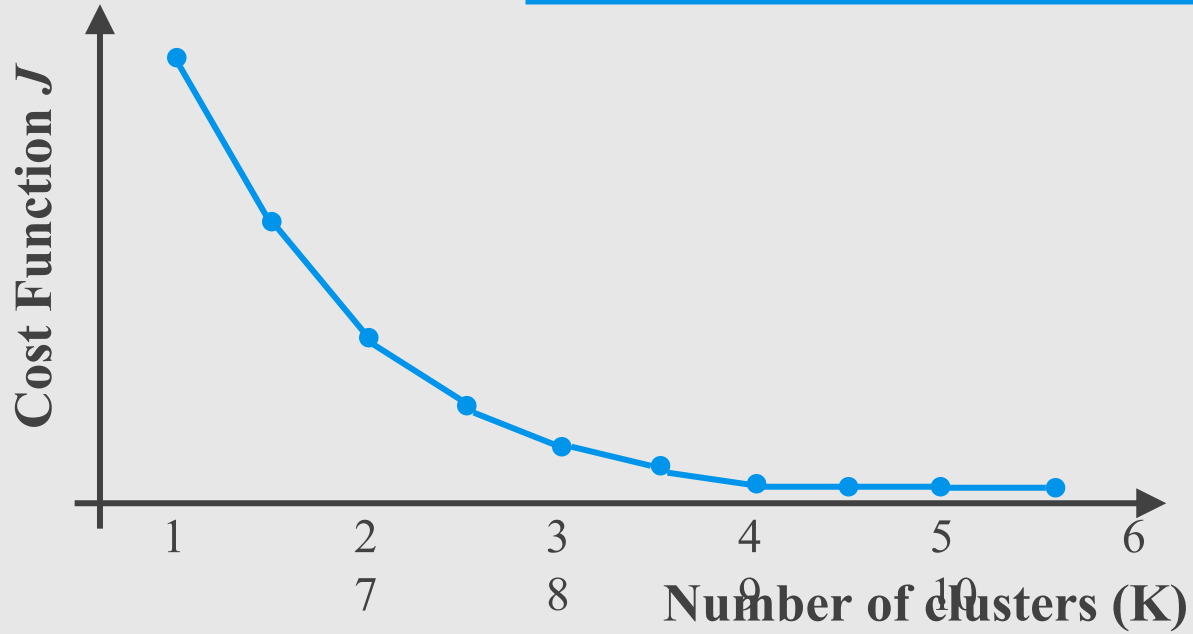


~~Elbow~~ Method



~~Elbow~~ Method

Q: You find that cost function *J* is much higher for k = 5 than for k =3. What can you conclude?



k-Means Variations

Mini-batch **k**-Means

- Uses mini-batches to reduce the computation time, while still attempting to optimize the same objective function.

- Converges faster than k-Means, but the quality of the results is reduced.

k-Medians Clustering

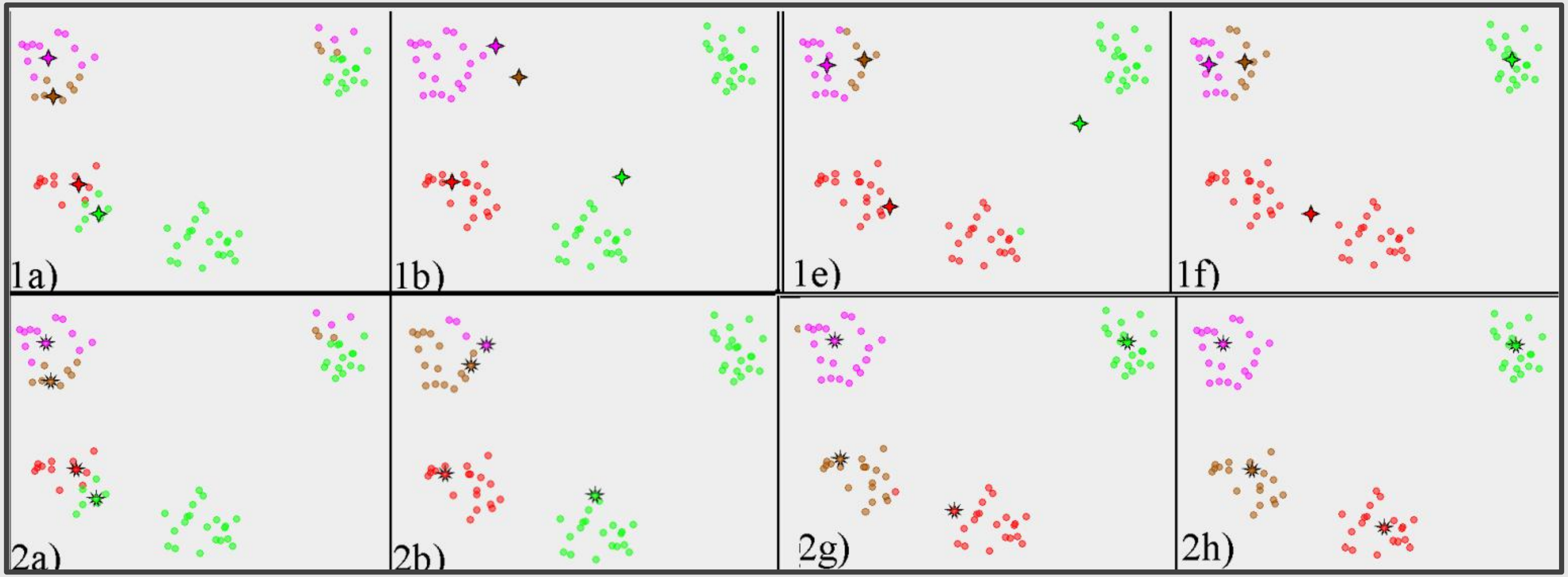
- Instead of calculating the mean for each cluster to determine its centroid, one instead **calculates the median**.

- Minimizing error over all clusters with respect to the **1-norm distance metric**, as opposed to the square of the 2-norm distance metric (which k-Means does).

k-Medoids Clustering

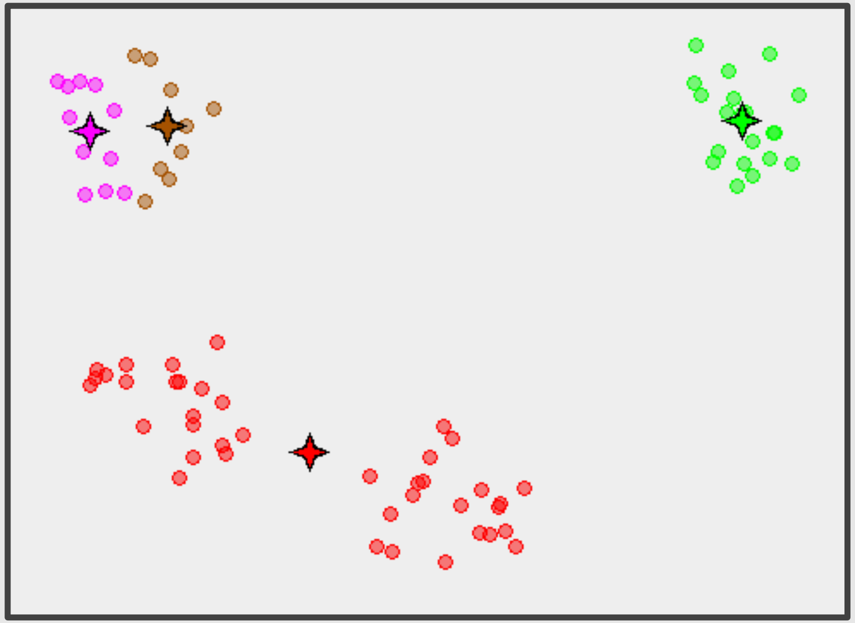
- Instead of calculating the mean for each cluster to determine its centroid, one instead **calculates the medoid**.
- Minimizing error over all clusters with respect to the **1-norm distance metric**.
- In contrast to the k-Means, k-Medoids **chooses data points as centroids**.

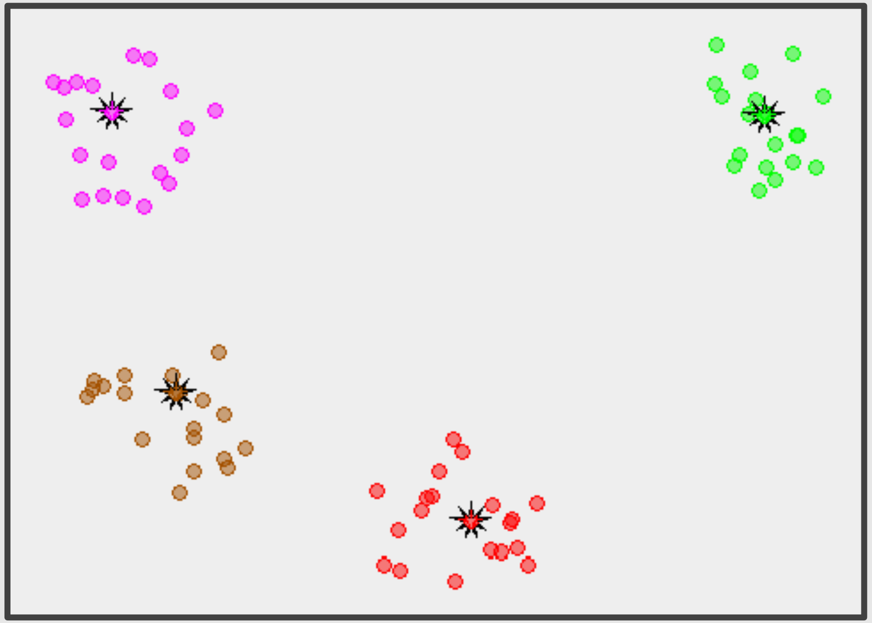
k-Means (top) vs **k**-Medoids (bottom)



Credit: https://commons.wikimedia.org/wiki/File:K-means_versus_k-medoids.png

k-Means (left) vs **k**-Medoids (right)





Credit: https://commons.wikimedia.org/wiki/File:K-means_versus_k-medoids.png

Fuzzy Clustering (Soft Clustering)

- Each data point can belong to more than one cluster.

Hierarchical Clustering

- **Agglomerative** (“bottom up”): each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

- **Divisive** (“top down”) :all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

DBSCAN Clustering

- **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise
- Given a set of points in some space, **it groups together points that are closely packed together** (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions.

Clustering Performance Evaluation

Clustering Evaluation

- Evaluating the performance of a clustering algorithm **is not as trivial** as counting the number of errors or the precision and recall of a supervised classification algorithm.

Clustering Evaluation

- Evaluating the performance of a clustering algorithm **is not as trivial** as counting the number of errors or the precision and recall of a supervised classification algorithm.

- Adjusted Rand index
- Mutual Information based scores
- Homogeneity, completeness and V-measure
- **Silhouette Coefficient**

Silhouette Coefficient

- The silhouette value is a measure of how similar a sample is to its own cluster (**cohesion**) compared to other clusters (**separation**).

- The silhouette ranges from −1 to +1.
  - High value = the clustering configuration is appropriate.
  - Low value = the clustering configuration may have too many or too few clusters.

Silhouette Coefficient

- The Silhouette Coefficient is defined **for each sample** and is composed of two scores:
  - ***a***: The mean distance between a sample and all other points in the same cluster.
  - ***b***: The mean distance between a sample and all other points in the next nearest cluster.

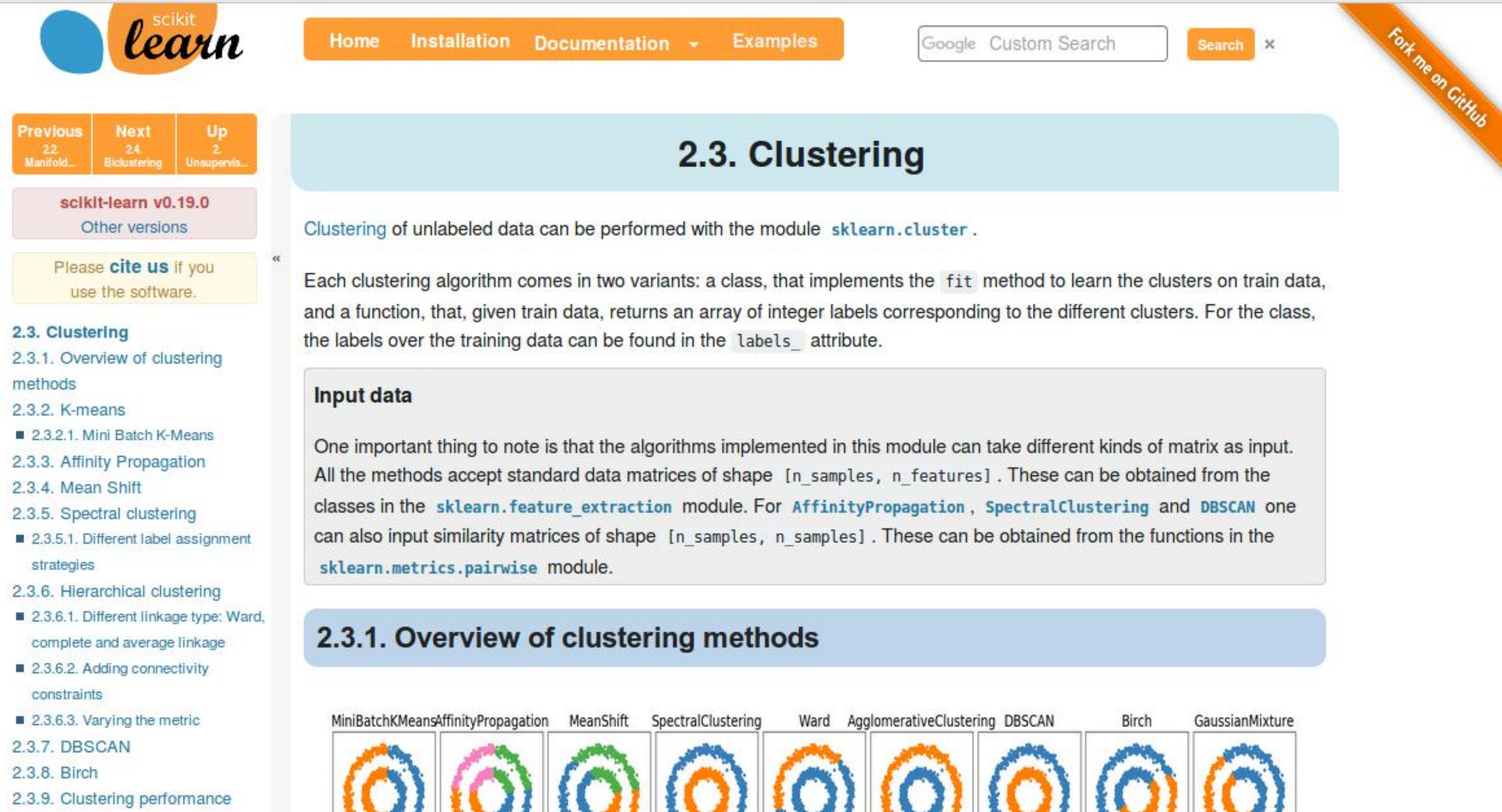
Silhouette Coefficient

- The Silhouette Coefficient s for a single sample is given as:

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

- The score is bounded between -1 for incorrect clustering and +1 for highly dense clustering ($a \ll b$). Scores around zero indicate overlapping clusters.

http://scikit-learn.org/stable/modules/clustering.html#clustering



scikit learn

Home Installation Documentation ▾ Examples

Google Custom Search Search ×

Fork me on GitHub

Previous 2.2. Manifold... Next 2.4. Biclustering Up 2. Unsupervis...

scikit-learn v0.19.0
Other versions

Please **cite us** if you use the software.

2.3. Clustering

Clustering of unlabeled data can be performed with the module `sklearn.cluster`.

Each clustering algorithm comes in two variants: a class, that implements the `fit` method to learn the clusters on train data, and a function, that, given train data, returns an array of integer labels corresponding to the different clusters. For the class, the labels over the training data can be found in the `labels_` attribute.

Input data

One important thing to note is that the algorithms implemented in this module can take different kinds of matrix as input. All the methods accept standard data matrices of shape `[n_samples, n_features]`. These can be obtained from the classes in the `sklearn.feature_extraction` module. For `AffinityPropagation`, `SpectralClustering` and `DBSCAN` one can also input similarity matrices of shape `[n_samples, n_samples]`. These can be obtained from the functions in the `sklearn.metrics.pairwise` module.

2.3.1. Overview of clustering methods

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

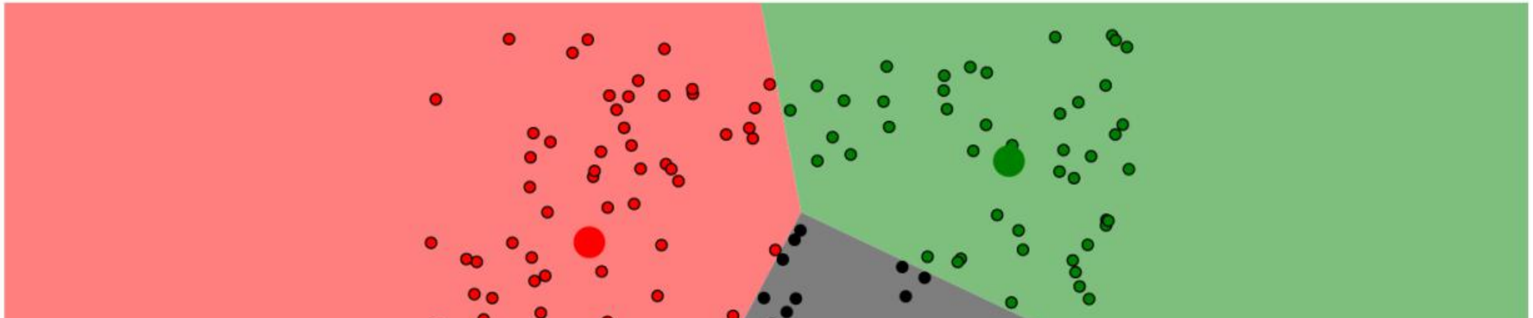
naftali harris

Blog About Contact I'm Feeling Lucky

Visualizing K-Means Clustering

January 19, 2014

Suppose you plotted the screen width and height of all the devices accessing this website. You'd probably find that the points form three clumps: one clump with small dimensions, (smartphones), one with moderate dimensions, (tablets), and one with large dimensions, (laptops and desktops). Getting an algorithm to recognize these clumps of points without help is called *clustering*. To gain insight into how common clustering techniques work (and don't work), I've been making some visualizations that illustrate three fundamentally different approaches. This post, the first in this series of three, covers the k-means algorithm. To begin, click an initialization strategy below:



References

Machine Learning Books

- Pattern Recognition and Machine Learning, Chap. 9 “Mixture Models and EM”
- Pattern Classification, Chap. 10 “Unsupervised Learning and Clustering”

Machine Learning Courses

- https://www.coursera.org/learn/machine-learning, Week 8