emmading.com

info@datainterviewpro.com

K-means

Lesson Structure

📌 Interview Questions

- Explain k-means clustering.
- How to choose k in k-means?
- Pros and cons of k-means.
- Implement k-means from scratch.

▼ K-means Algorithm

k-means is a centroid-based clustering algorithm. It's very popular and used in a variety of applications such as market segmentation, document clustering, fraud detection, and image segmentation, etc.

▼ 4 step algorithm

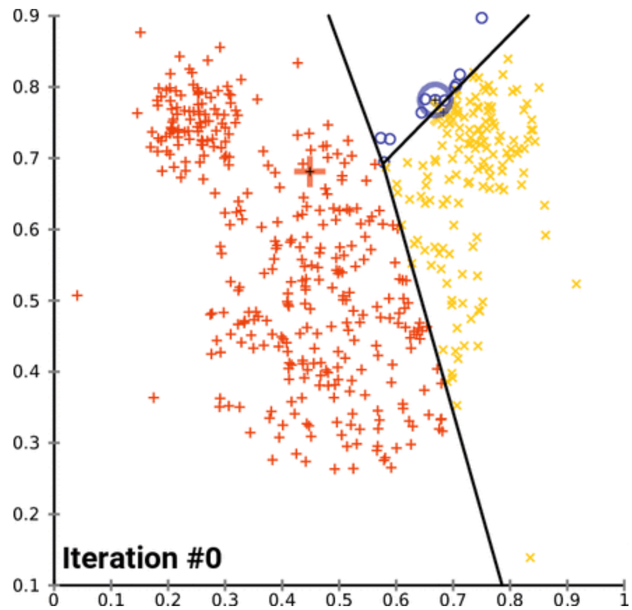
1. Randomly pick k centroids from the training examples as initial cluster centers.
    - These centroids should be chosen in a smart way because different positions lead to different results. A good choice is to place the initial centroids far away from each other (instead of random initialization).
2. Assign each example to the nearest centroid.

    ▼ Distance Metric: Euclidean distance

$$d(x,y) = \sqrt{\sum_{j=1}^{m}(x_j - y_j)^2} = ||x-y||_2^2$$

> 💡 **Feature scaling** is important for k-means.
> We want to make sure that the features are measured on the same scale, so we need to apply normalization or standardization if necessary.

3. Move the centroids to the center (average) of the examples that were assigned to it.
    - Compute the average for all the points inside each cluster, then move the cluster centroid to the average.
4. Repeat step 2 and 3 until some stopping criteria are met.
    - Convergence (cluster assignments do not change)
    - Maximum number of iterations is reached
    - A user-defined tolerance is reached (e.g. variance does not improve by at least X)



Convergence of k-means https://en.wikipedia.org/wiki/K-means_clustering

▼ **Objective function**

k-means algorithm chooses centroids that minimize the **within-cluster sum-of-squared errors (SSE)**, or **cluster inertia**:

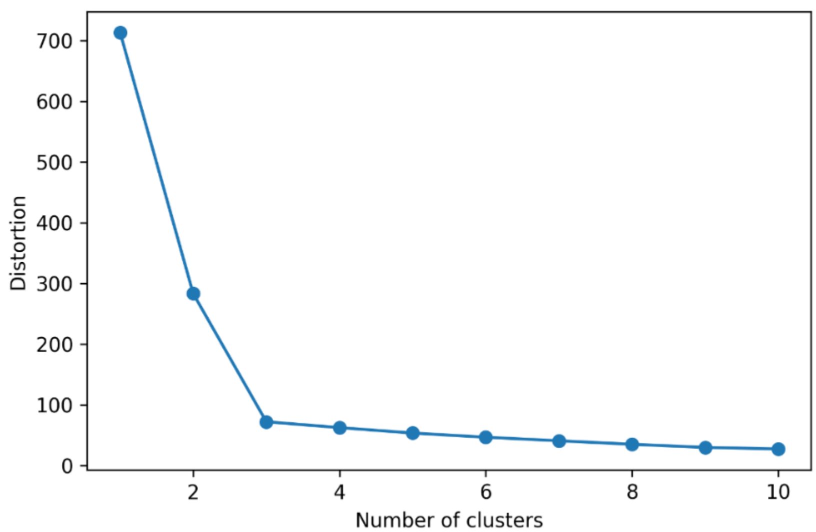
$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||_2^2)$$

- x_i: examples in cluster j.
- μ_j: The centroid for cluster j.

How to choose k?

Elbow method

- The intuition behind this technique is that the first few clusters will explain a lot of the variation in the data, but past a certain number of clusters, the amount of information added is diminishing.
- Use SSE to quantify the quality (homogeneity) of clustering. If k increases, SSE will decrease because examples will be closer to the centroids they are assigned to.
- Elbow Curve: identity the value of k ("elbow") where SSE begins to increase most rapidly. This is a point after which we don't see much decrement in SSE.



Use the elbow method to find optimal value of k

Looking at the above graph of explained variation on the y-axis versus the number of clusters (k), there's be a sharp change in the y-axis when k = 3.

Silhouette Coefficient

Measures how similar points are in its cluster compared to other clusters.

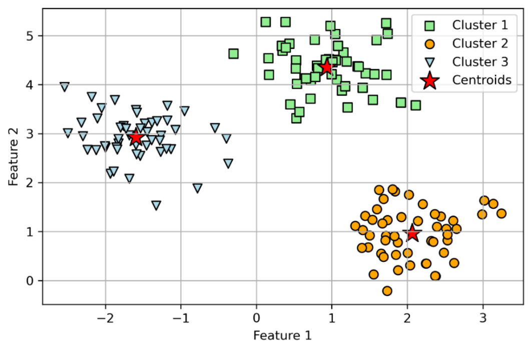
$$s = \frac{b - a}{max(a, b)}$$

- **a**: The average distance between an example and all other points in the **same** cluster. (similarity)
- **b**: The average distance between an example and all other points in the **next closest** cluster. (dissimilarity)

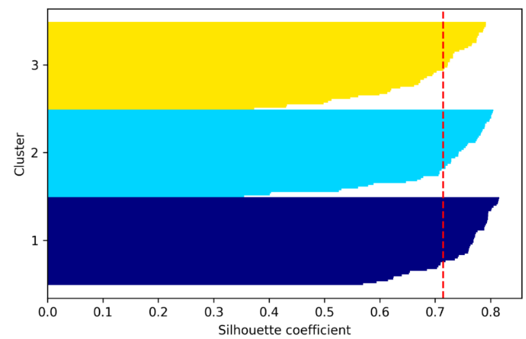
Silhouette coefficient varies between - 1 and 1 for any given example.

- 1: the example is in the right cluster as $b >> a$.
- 0: cluster separation and cohesion are equal.
- -1: the example is in a wrong cluster as $a >> b$.

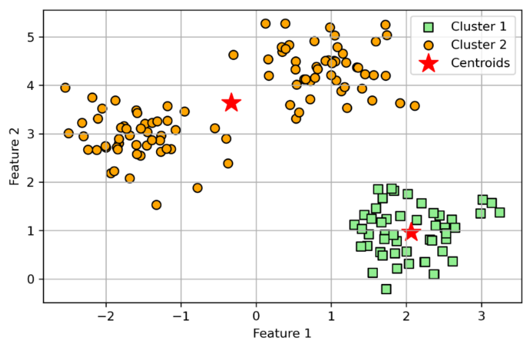
By plotting the coefficient versus k, we can get an idea of the optimal value of k.



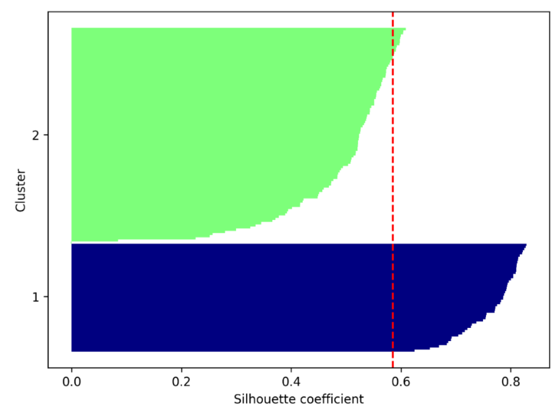
k-means clustering with 3 centroids



Silhouette coefficients when k = 3



k-means clustering with 2 centroids



Silhouette coefficients when k = 2

▼ Pros and Cons

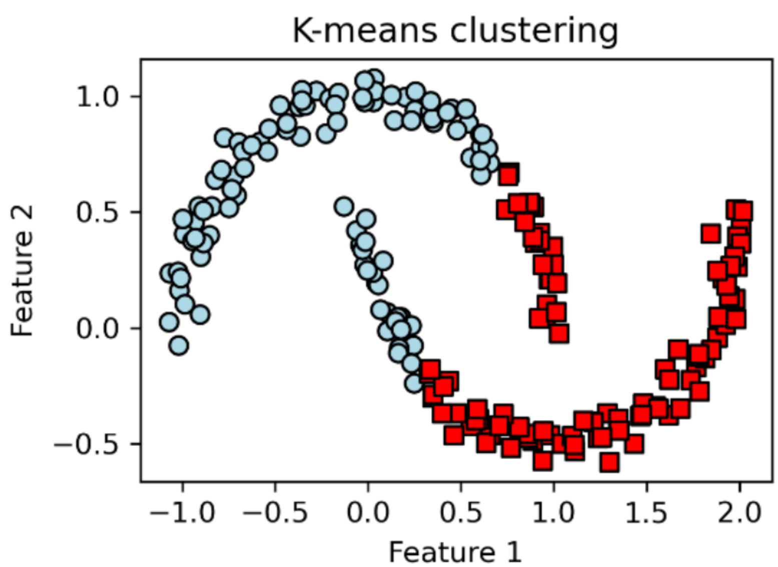
▼ Pros

- Easy to implement.
- Computationally efficient.
- Speed is K-means' big win.

💡 K-means scales well to large numbers of samples and has been used across a large range of applications in many different fields.

▼ **Cons**

- The number of clusters, k, has to be determined. An inappropriate choice of k can result in poor clustering performance.
- Stability: Initial positions of centroids influence the final position, so two runs can result in two different clusters.
- The shapes of clusters can only be circular (because Euclidean distance doesn't prefer one direction over another). It does not work well for datasets requiring flexible cluster shapes.
    - e.g. K-means is unable to separate this half-moon-shaped dataset.



📌 K-means is susceptible to **curse of dimensionality**. In very high-dimensional spaces, Euclidean distances tend to become inflated.
Running a dimensionality reduction algorithm such as Principal component analysis (PCA) prior to k-means can alleviate this problem and speed up the computations.