K means intro

When to use

K-Means is an unsupervised learning algorithm used when the data is unlabeled. It helps group similar data points into clusters without prior knowledge of labels.

It is ideal when the goal is to segment data into distinct groups (clusters) based on similarities. This is useful in customer segmentation, document classification, image compression, etc.

K-Means works efficiently with large datasets because of its relatively simple structure and fast convergence.

Continuous Variables: K-Means is most effective when the features (variables) in the dataset are continuous or numeric.

Spherical Clusters: It performs well when clusters are approximately spherical (circular) and have similar sizes. This fits the assumptions of the algorithm.

What

It is an unsupervised learning. K-Means is used to cluster the data into groups based on similarity. It does this by finding centroids (central points) for each cluster and assigning each data point to the nearest centroid.

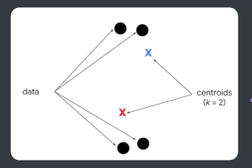
https://scikit-learn.org/stable/modules/clustering.html

Centroid

The center of a cluster determined by the mathematical mean of all points in that cluster

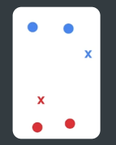
Steps

1. Choose the number of centroids (K)



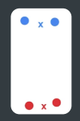
Initiate k centroids

2. Assign all points to their nearest centroid



Assign points

3. Recalculate the centroid of each cluster based on the points assigned to it



Calculate k means

4. Steps 2 and 3 are repeated iteratively until the centroids no longer move significantly (i.e., convergence is reached). At this point, the clusters are stable.

Assumptions

Clusters are Spherical and Well-Separated: K-Means assumes that the clusters are spherical (circular in 2D space) and have similar sizes. This means that each point in a cluster is closer to its cluster's centroid than to any other centroid.

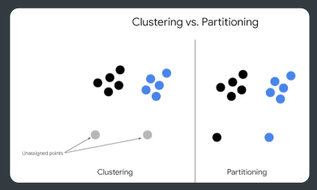
Clusters K is known: The number of clusters (K) must be specified beforehand, and the algorithm assumes that this number is correct.

Equal Importance of Features:
All features in the dataset contribute equally to the distance calculation. There is no inherent weighting for features unless explicitly added.

Points are Assigned to the Nearest Centroid:
The algorithm assigns each point to the cluster whose centroid is closest, based on Euclidean distance (unless a different distance metric is specified).

Local minima

Not having enough distance between clusters this will result to poor clustering. K-Means can sometimes end up in suboptimal clustering due to local minima. This happens when the centroids do not have enough separation between them, leading to poor clustering. Different initializations of centroids can yield different results, and techniques like the Elbow Method or Silhouette Score are used to find the optimal K value and avoid local minima.



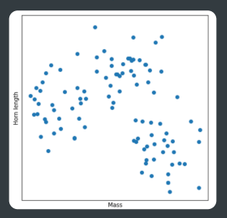
K means does not allow unassigned outliers

randomize centroids

In scikit-learn, this implementation is called K-means++. K-means++ still randomly initializes centroids in the data, but it does so based on a probability calibration. Basically, it randomly chooses one point within the data to be the first centroid, then it uses other data points as centroids, selecting them pseudo-randomly. The probability that a point will be selected as a centroid increases the farther it is from other centroids. This helps to ensure that centroids aren't initially placed very close together, which is when convergence in local minima is most likely to occur.

Example

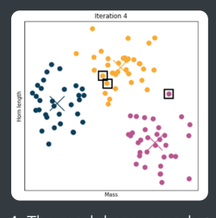
Suppose you have 100 sample data of 3 species beetles



1. The data compares total body mass to horn length

2. Separate by color. You can use K-means to cluster this data.

3. Zero iteration (centroids are marked as X)



4. The model converged at iteration 4

*Remember, you will not always know how many clusters you should have, and you probably won't be able to visualize your data so easily because it will likely have more than three features (i.e., dimensions)

K means intro

1. What

1.1. It is an unsupervised learning. K-Means is used to cluster the data into groups based on similarity. It does this by finding centroids (central points) for each cluster and assigning each data point to the nearest centroid. https://scikit-learn.org/stable/modules/clustering.html

1.2. Centroid

1.2.1. The center of a cluster determined by the mathematical mean of all points in that cluster

2. Steps

2.1. 1. Choose the number of centroids (K)

2.1.1. Initiate k centroids

2.1.1.1. 2. Assign all points to their nearest centroid

2.1.1.1.1. Assign points

2.2. 3. Recalculate the centroid of each cluster based on the points assigned to it

2.2.1. Calculate k means

2.2.1.1. 4. Steps 2 and 3 are repeated iteratively until the centroids no longer move significantly (i.e., convergence is reached). At this point, the clusters are stable.

3. Local minima

3.1. Not having enough distance between clusters this will result to poor clustering. K-Means can sometimes end up in suboptimal clustering due to local minima. This happens when the centroids do not have enough separation between them, leading to poor clustering. Different initializations of centroids can yield different results, and techniques like the Elbow Method or Silhouette Score are used to find the optimal K value and avoid local minima.

3.2. K means does not allow unassigned outliers

3.3. randomize centroids

3.3.1. In scikit-learn, this implementation is called K-means++. K-means++ still randomly initializes centroids in the data, but it does so based on a probability calibration. Basically, it randomly chooses one point within the data to be the first centroid, then it uses other data points as centroids, selecting them pseudo-randomly. The probability that a point will be selected as a centroid increases the farther it is from other centroids. This helps to ensure that centroids aren't initially placed very close together, which is when convergence in local minima is most likely to occur.

4. Example Suppose you have 100 sample data of 3 species beetles

4.1. 1. The data compares total body mass to horn length

4.2. 2. Separate by color. You can use K-means to cluster this data.

4.3. 3. Zero iteration (centroids are marked as X)

4.4. 4. The model converged at iteration 4

5. *Remember, you will not always know how many clusters you should have, and you probably won't be able to visualize your data so easily because it will likely have more than three features (i.e., dimensions)

6. Assumptions

6.1. **Clusters are Spherical and Well-Separated:** K-Means assumes that the clusters are spherical (circular in 2D space) and have similar sizes. This means that each point in a cluster is closer to its cluster's centroid than to any other centroid.

6.2. **Clusters K is known:** The number of clusters (K) must be specified beforehand, and the algorithm assumes that this number is correct.

6.3. **Equal Importance of Features:** All features in the dataset contribute equally to the distance calculation. There is no inherent weighting for features unless explicitly added.

6.4. **Points are Assigned to the Nearest Centroid:** The algorithm assigns each point to the cluster whose centroid is closest, based on Euclidean distance (unless a different distance metric is specified).

7. When to use

7.1. K-Means is an unsupervised learning algorithm used when the data is unlabeled. It helps group similar data points into clusters without prior knowledge of labels.

7.2. It is ideal when the goal is to segment data into distinct groups (clusters) based on similarities. This is useful in customer segmentation, document classification, image compression, etc.

7.3. K-Means works efficiently with large datasets because of its relatively simple structure and fast convergence.

7.4. Continuous Variables: K-Means is most effective when the features (variables) in the dataset are continuous or numeric.

7.5. Spherical Clusters: It performs well when clusters are approximately spherical (circular) and have similar sizes. This fits the assumptions of the algorithm.