Member-only story

UNSUPERVISED LEARNING

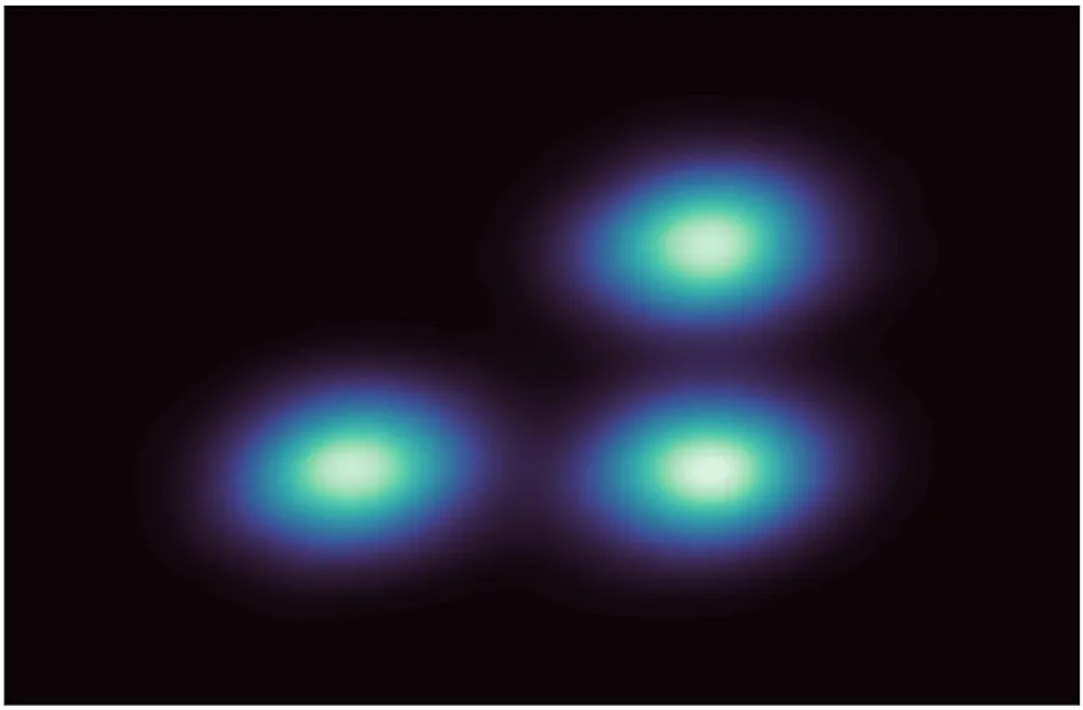
Understanding Mean Shift Clustering and Implementation with Python

In this post, I briefly go over the concept of an unsupervised learning method, mean shift clustering, and its implementation in Python

Yufeng · Follow

Published in Towards Data Science

7 min read · Feb 22, 2022



Mean Shift Clustering (image by author)

Mean shift is an unsupervised learning algorithm that is mostly used for clustering. It is widely used in real-world data analysis (e.g., image segmentation)because it's non-parametric and doesn't require any predefined shape of the clusters in the feature space.

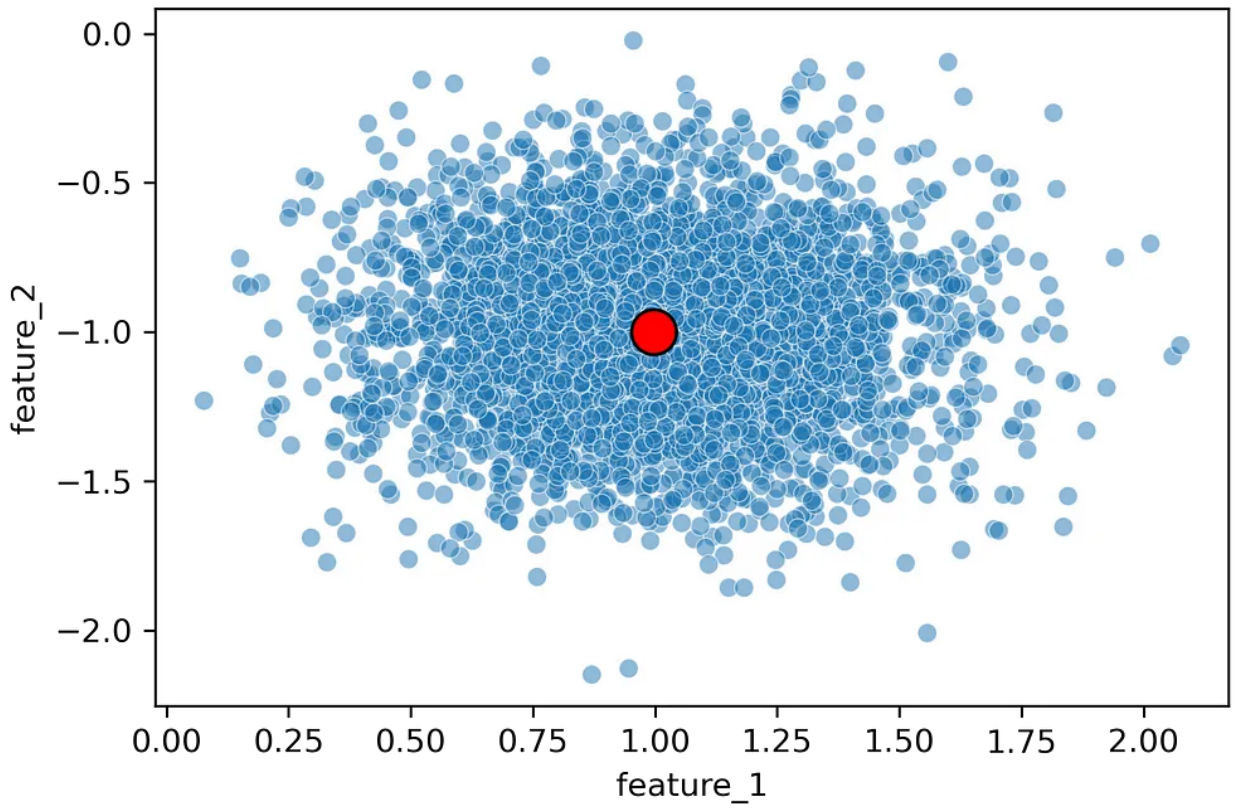
Simply speaking, "mean shift" is equal to "shifting to the mean" in an iterative way. In the algorithm, every data point is shifting to the "regional mean" step by step and the location of the final destination of each point represents the cluster it belongs to.

In this post, I briefly go over how to understand this algorithm as well as the implementation in Python. Hope this article is helpful!

What Mean?

So, based on the brief description of the mean shift algorithm above, you may have already noticed that several terms are still confusing without a further explanation. The very first one is the "mean" we are referring to.

If we look at a sample dataset with two features below,



Mean Point (red) of a sample dataset. (image by author)

We can easily locate the “mean point” of the entire dataset (the red one in the plot above) by calculating the mean of feature_1 and the mean of feature_2. To note, the “mean point” here is defined by the arithmetic mean of feature_1 and that of feature_2, respectively, because it is calculated based on the equal weights to all points,

$$M_A = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Arithmetic mean (image by author)

where M represents the mean, n is the sample size, and x_i is one feature (feature_1 or feature_2)of the data points.

Sometimes it's not reasonable to give equal weights to everything, we can also calculate the weighted mean,

$$M_W = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

Weighted Mean (Image by author)

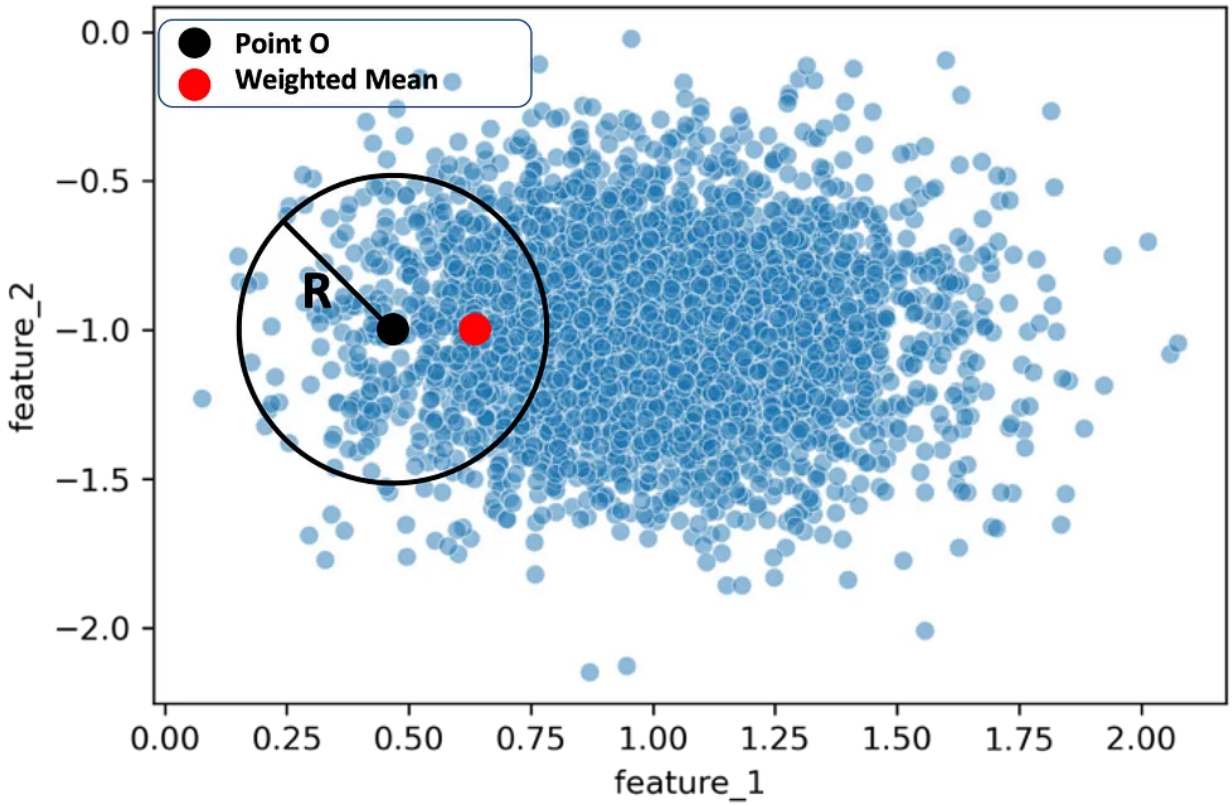
where w_i is the weight for x_i, and it's related to the euclidean distance between points. The most widely used weight function in mean shift algorithm is a flat one,

$$w(d) = \begin{cases} 1, & \text{if } d \leq R \\ 0, & \text{if } d > R \end{cases}$$

Flat weight function (image by author)

where d is the distance between any data point to the currently investigated one, and R is the radius of the circle centered at the investigated point.

For example, if we want to calculate the weighted mean for the area around point O, a flat weighting function can be understood as a hard boundary circle centered at O with radius R. Everything inside the circle will be counted and any data point outside the circle will be ignored.



Weighted Mean by Flat weighting function (image by author)

It's kind of like that we are standing on one local point (center point O) and cannot see the entire picture but are restricted to a local area to calculate the mean. Here, we see the red point is the weighted mean of the investigated area and we found that it tends to locate at a region with high density of points.

We can also replace the flat weight function by a Gaussian one,

$$w(d) = e^{-\frac{d}{2\sigma^2}}$$

Gaussian weight function (image by author)

where d is still the distance between the center point to any other point, and sigma here is a parameter to adjust how fast the weight decreases with the increasing of the distance.

The idea is similar to that of the flat weight function, the closer a point is to the circle center, the higher the weight is in the mean calculation. For simplicity of implementation and clear physical meaning, mean shift algorithm typically uses **the flat weight function** to calculate the mean.

The local mean point we calculated above kind of represents the position with the **largest density of points** in that specific local area. You may have already noticed that the radius of the black circle, **R**, in the plot above is very important to define the local region. Actually, the radius is the only parameter in the mean shift algorithm, which is called "**bandwidth**".

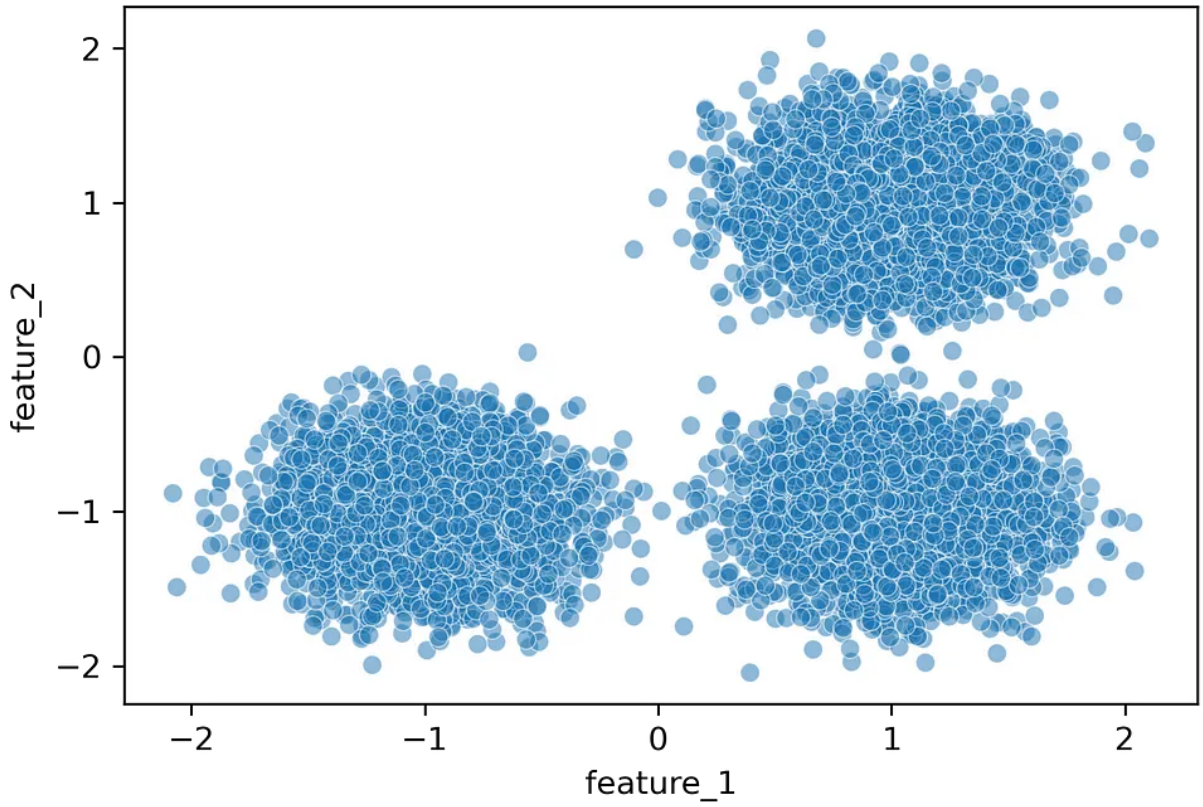
Up till now, I think you have understood what the mean is when we investigate one data point given the bandwidth parameter.

Why Shift?

One of the biggest problems I had when understanding the mean shift algorithm is "*why should the point shift to the mean??*" Are we changing the raw dataset when applying this method?

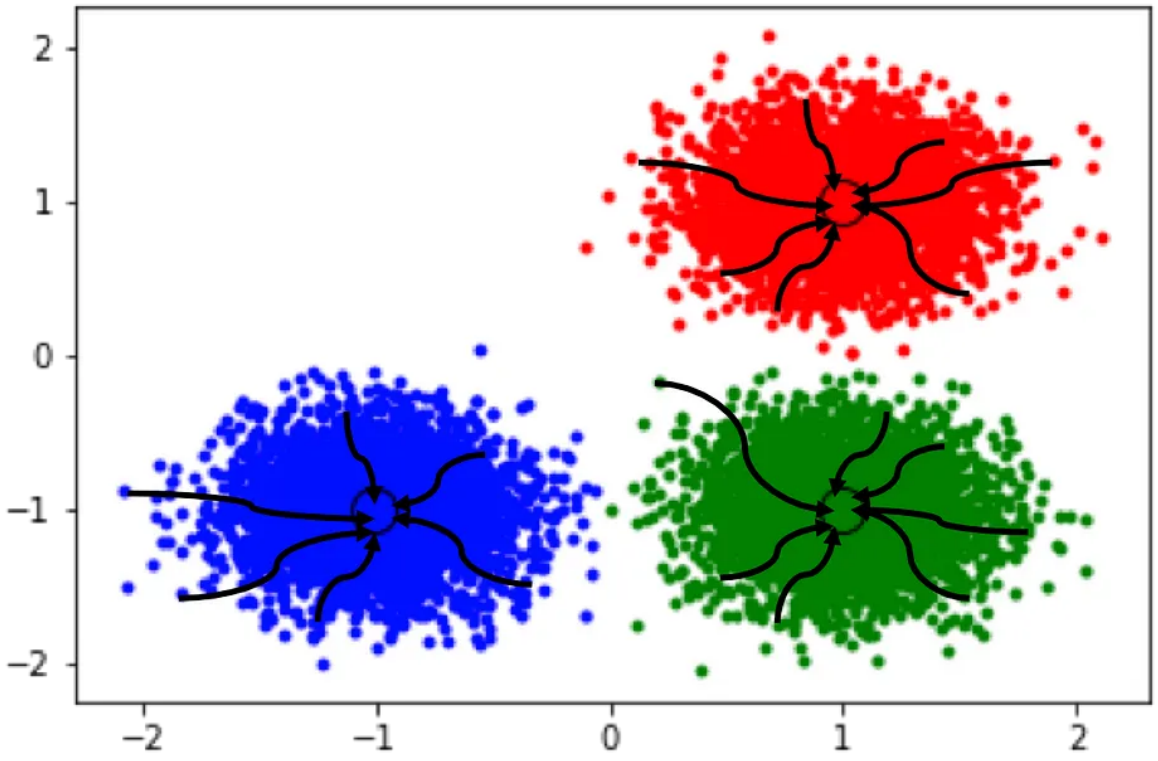
Actually, no, we are not changing anything to the raw data. The "shifting" just describes the process of how we label each data point. Similar points gradually "shift" to the centroid of the cluster they belong to, in order to be labeled as such.

Let's look at a 2-D toy dataset, which is designed to have three clusters.



Toy dataset with three clusters (image by author)

In the mean shift algorithm, each point try to find its group by moving towards the weighted mean of its local area in each step. The destination of each point will be the centroid of the data cluster that the point belongs to. Then, all the data points with the same destination point can be labeled with the same cluster.



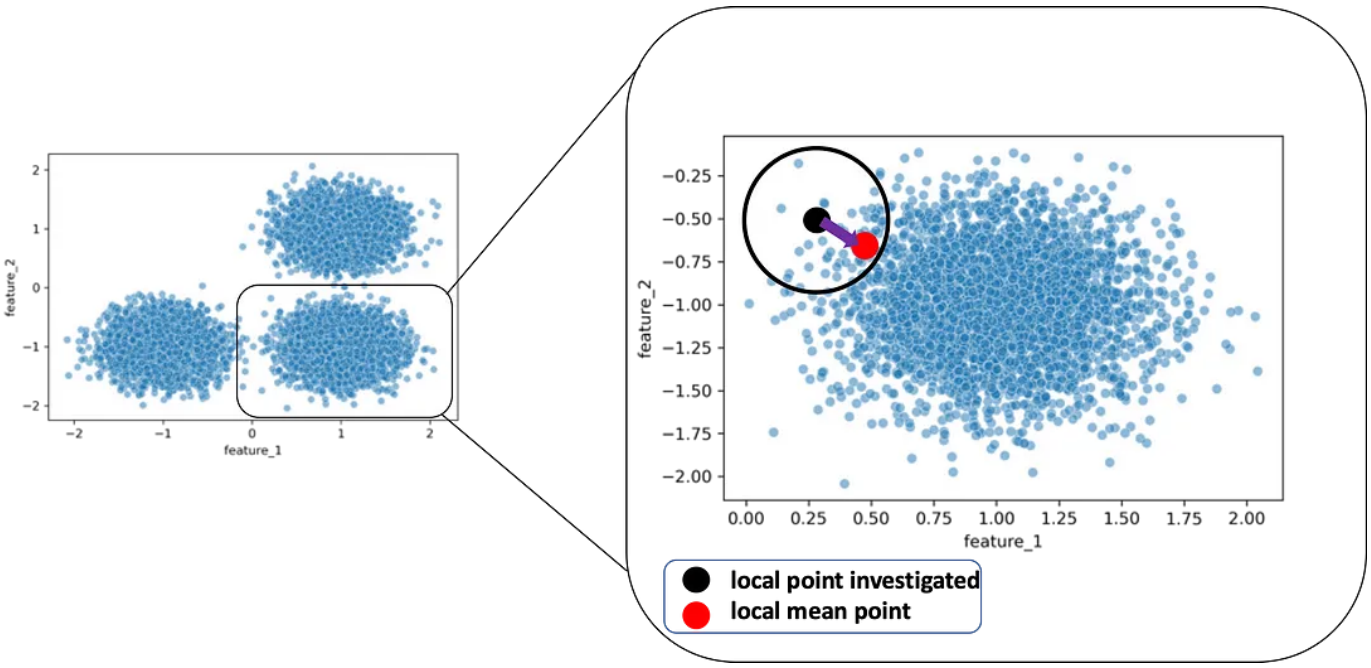
Toy dataset with three clusters with moving trajectory (image by author)

The plot above shows the result of labeling, where the black big circles indicate the final destinations of each group of data points and the arrows approximately shows the trajectory of some of the moving paths.

So, you see the data is not changed, but only labeled with its cluster id.

How to Shift?

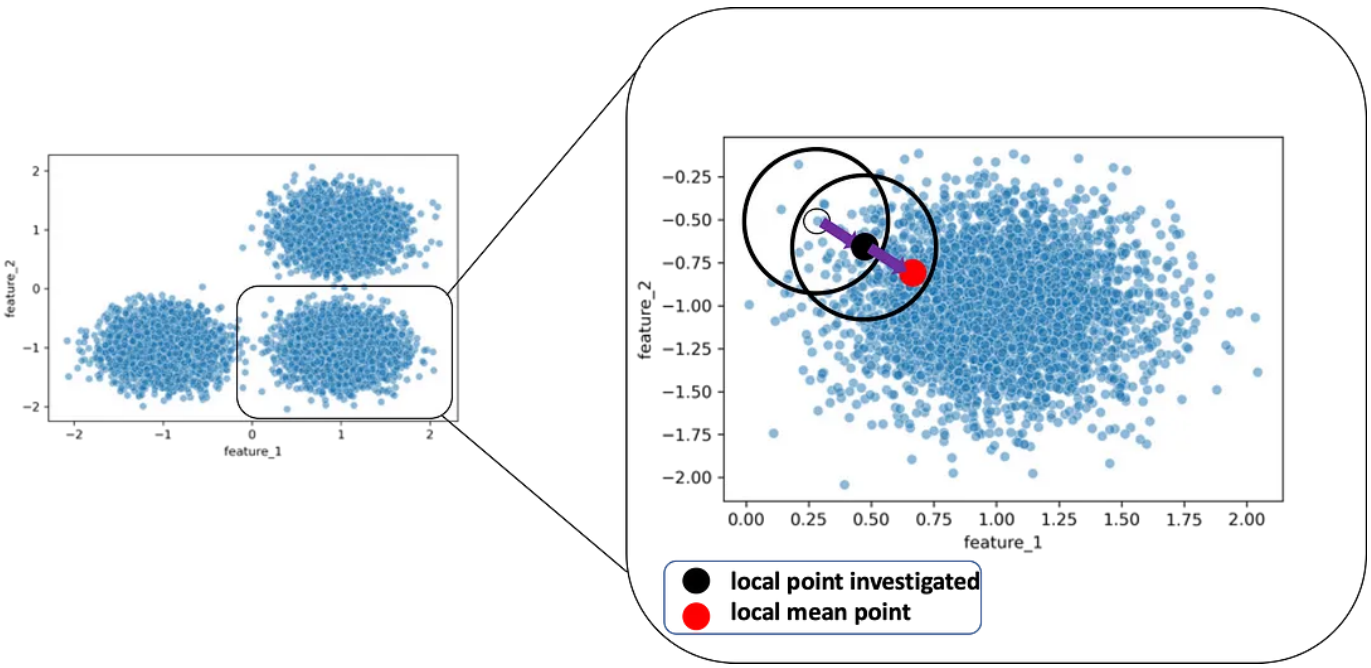
Let's use a single data point as an example to show how the shifting processes. Suppose we have a point located at the edge of the bottom right cluster (black dot shown in the plot below).



step 1 for a point in Mean Shift (image by author)

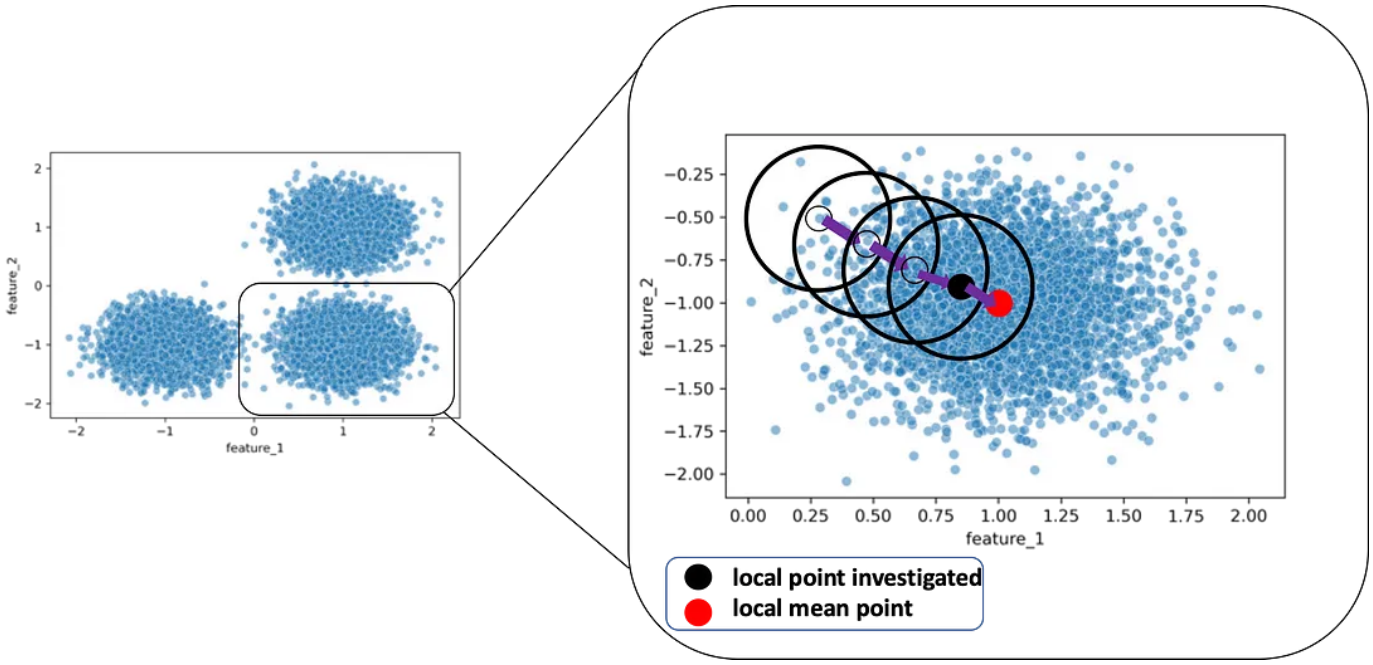
After calculating the mean value of *feature_1* and that of *feature_2* within the local area defined by the bandwidth parameter, we get the location of the mean point (shown as the red dot above). Then the center of circle moves from the black dot to the red one.

Next, the aforementioned procedure is repeated at the new point position as shown below.



step 2 for a point in Mean Shift (image by author)

After sufficient time of iterations or the number of points within the black circle no more increase (converge), the original data point arrives at its destination, the centroid of the cluster it belongs to.

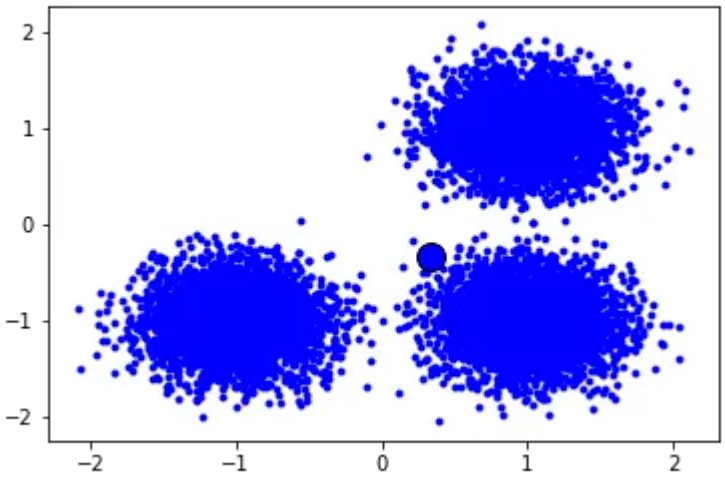


step N for a point in Mean Shift (image by author)

You may have noticed that the calculation will be super redundant if this procedure is performed for every data point. So, in practice, when these moving circles overlap, only the one containing the most points is preserved.

Since mean shift clustering only has one parameter, the bandwidth (or the window size in some contexts), it needs super caution when choosing a appropriate value for it.

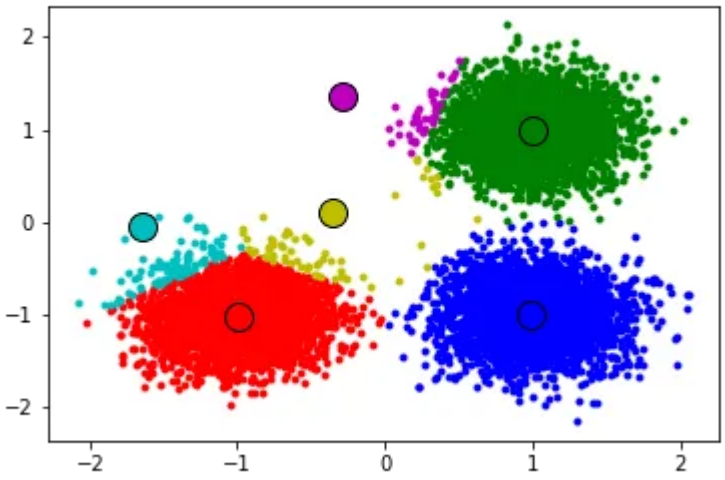
You may have already noticed that the larger the local region is, the closer the local mean point is to the global one. If the local area is **super large** for every data point investigated, then all of the "local mean points" are actually located at the position of the "global mean point".



Mean shift with super large bandwidth (image by author)

In other words, the super large local region setting can disable us from seeing the local structure of the dataset.

If we have a super small bandwidth, there will be a lot more non-sense clusters because we restricted the mean value calculation to a much smaller local area.



Mean shift with super small bandwidth (image by author)

So, it's better to have an idea about the bandwidth when doing the modeling. However, if we have no prior knowledge about the data, we can still estimate the bandwidth from the pairwise distances randomly sampled from the entire dataset.

Implementation in Python

The implementation of mean shift clustering is relatively easy thanks to the sklearn package. The following codes show how to estimate the bandwidth and use the estimated parameter to do the clustering.

```
bandwidth = estimate_bandwidth(X, quantile=0.3, n_samples=300)

ms = MeanShift(bandwidth=bandwidth)
ms.fit(X)
```

To extract the labels of the data points from the clsutering result, we can do,

```
labels = ms.labels_
```

That's it! Hope the article is helpful!

If you enjoy reading my articles, please subscribe to my Medium account.

References:

A demo of the mean-shift clustering algorithm

Reference: Dorin Comaniciu and Peter Meer, "Mean Shift: A robust approach toward feature space analysis". IEEE...

scikit-learn.org

Mean shift - Wikipedia

Mean shift is a non-parametric feature-space mathematical analysis technique for locating the maxima of a density...

en.wikipedia.org

Unsupervised Learning Data Science Clustering Python Mean Shift

Follow

Written by Yufeng

736 Followers · Writer for Towards Data Science

Ph.D., Data Scientist and Bioinformatician. Support my writing by becoming one of my referred members: https://jianan-lin.medium.com/membership

More from Yufeng and Towards Data Science

Yufeng in Towards Data Science

Extract Rows/Columns from A Dataframe in Python & R

Here is a simple cheat sheet of data frame manipulation in Python and R, in case you get upset about mixing the commands of the two...

· 6 min read · May 14, 2020

189

Jacob Marks, Ph.D. in Towards Data Science

How I Turned My Company's Docs into a Searchable Database with OpenAI

And how you can do the same with your docs

15 min read · Apr 25

3K 39

Leonie Monigatti in Towards Data Science

Getting Started with LangChain: A Beginner's Guide to Building LLM-Powered Applications

A LangChain tutorial to build anything with large language models in Python

· 12 min read · Apr 25

 Yufeng in Towards Data Science

Pheatmap Draws Pretty Heatmaps

A tutorial of how to generate pretty heatmaps with pheatmap in R.

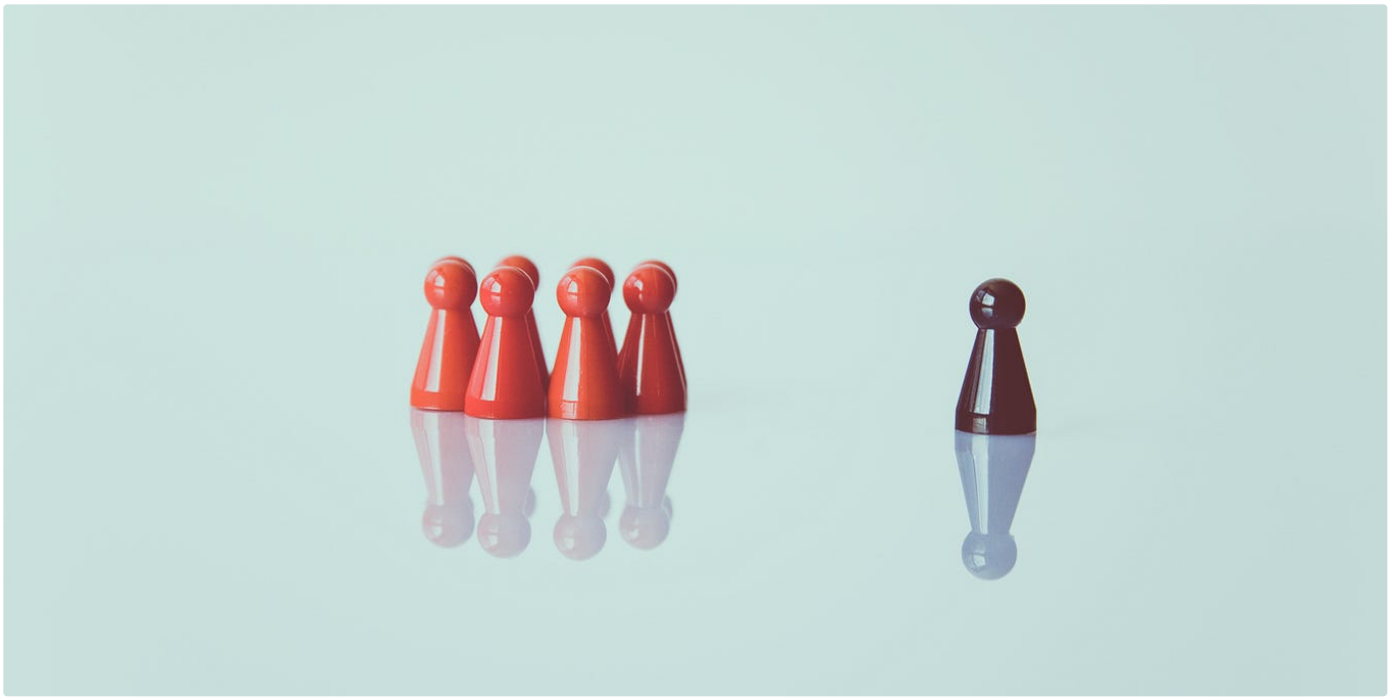
· 6 min read · Apr 9, 2020

42 2

See all from Yufeng

See all from Towards Data Science

Recommended from Medium

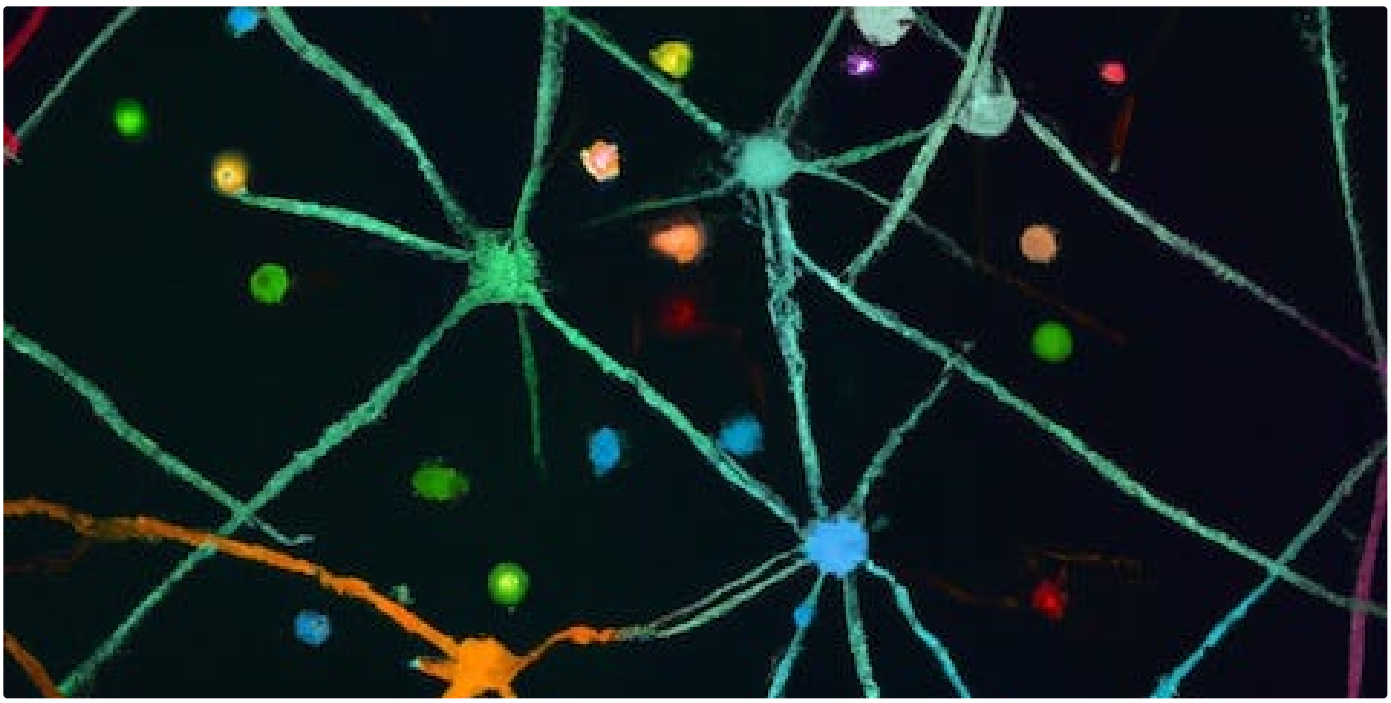


Kay Jan Wong in Towards Data Science

7 Evaluation Metrics for Clustering Algorithms

In-depth explanation with Python examples of unsupervised learning evaluation metrics

· 10 min read · Dec 9, 2022



Aydin Schwartz in Level Up Coding

Visualizing Clustering Algorithms: K-Means and DBSCAN

Data scientists and engineers are not always lucky enough to be working with nicely-labeled datasets. If we're given a bunch of unlabeled...

8 min read · Mar 24

58

Lists

What is ChatGPT?

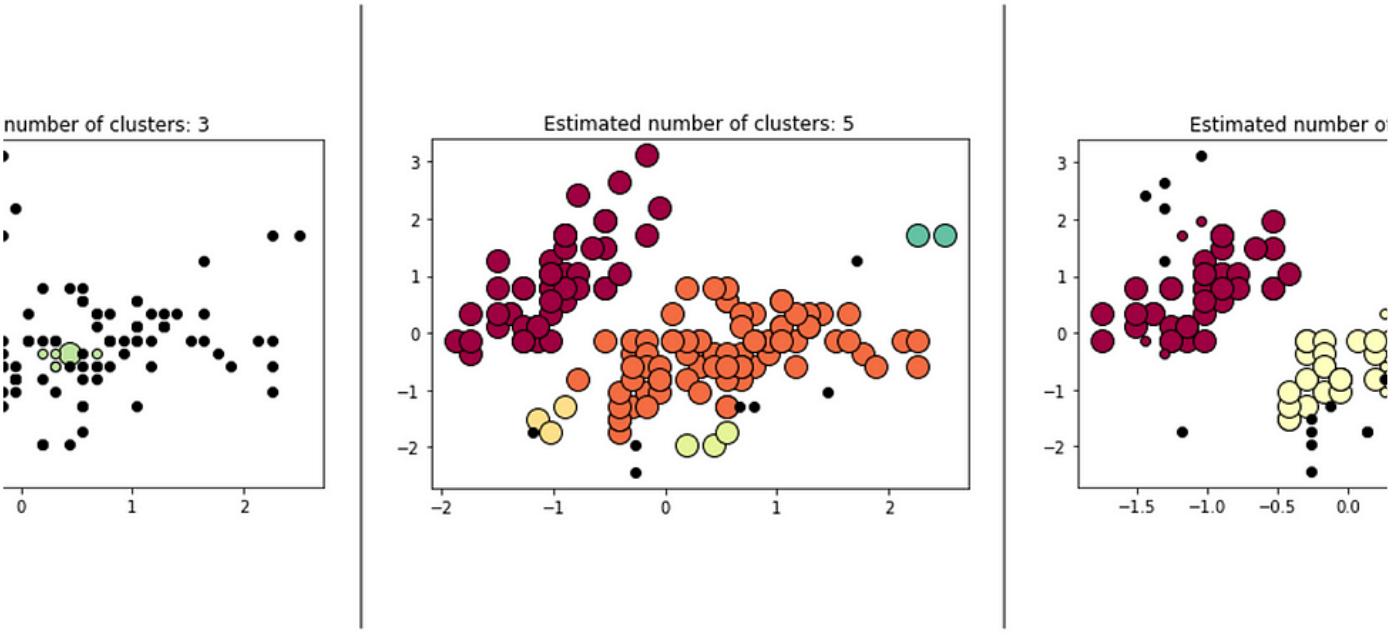
9 stories · 57 saves

Stories to Help You Level-Up at Work

19 stories · 48 saves

Staff Picks

323 stories · 82 saves



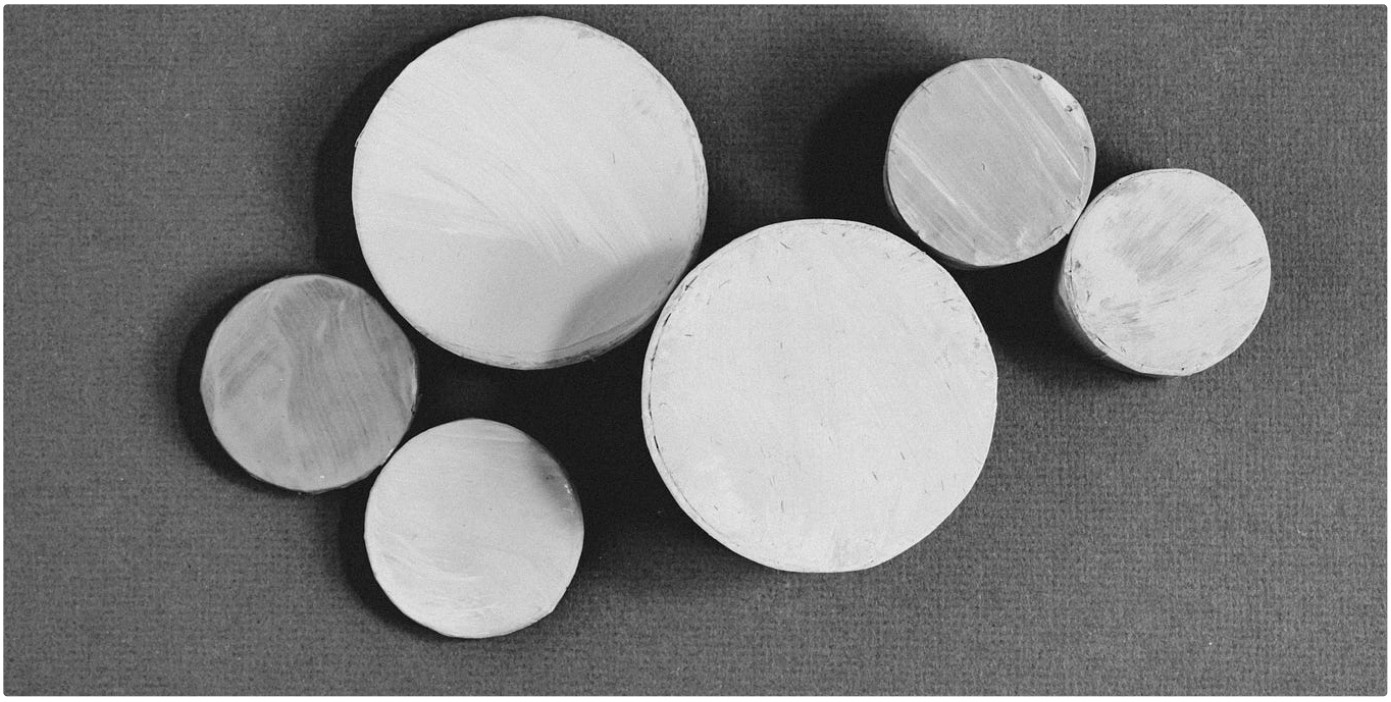
Carla Martins in CodeX

Understanding DBSCAN Clustering: Hands-On With Scikit-Learn

Unsupervised Learning—Clustering

✨ · 5 min read · Dec 21, 2022

274 1



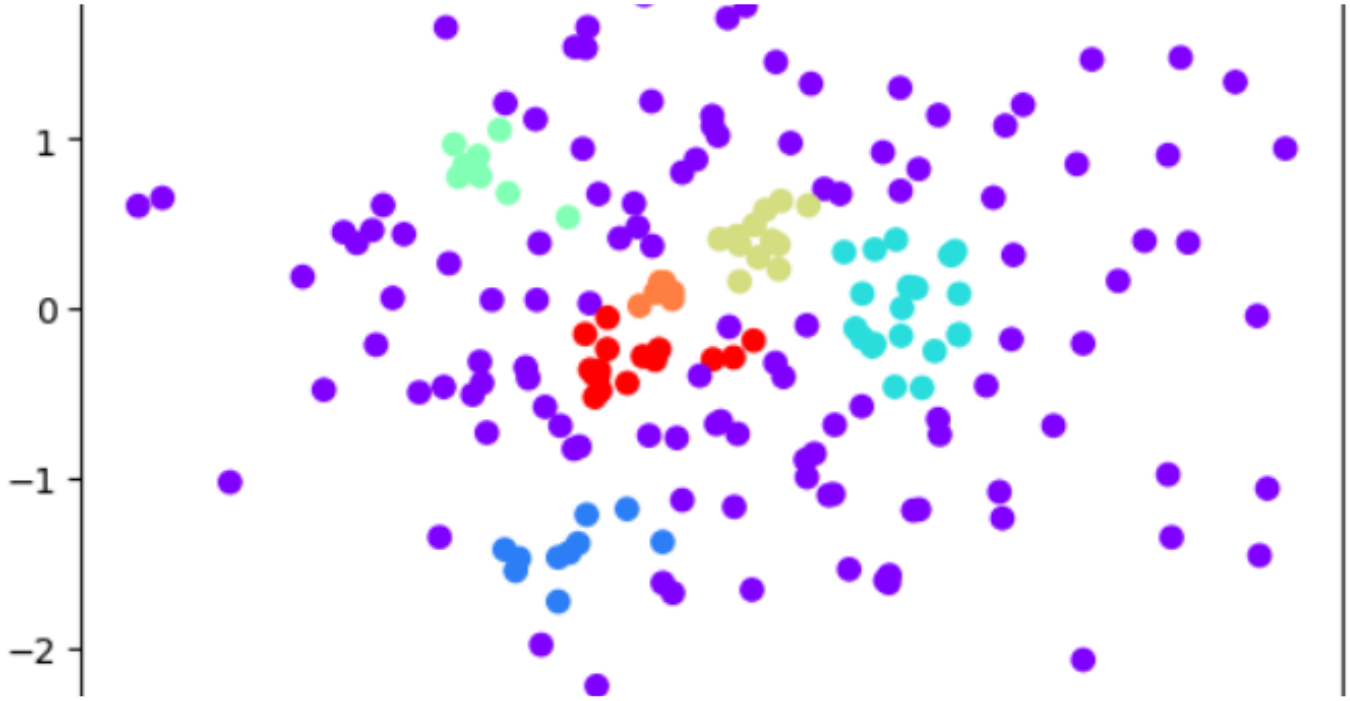
Okan Yenigün in Dev Genius

K-Means Clustering: An Introduction

Exploration and Implementation of K-Means

6 min read · Feb 8

43

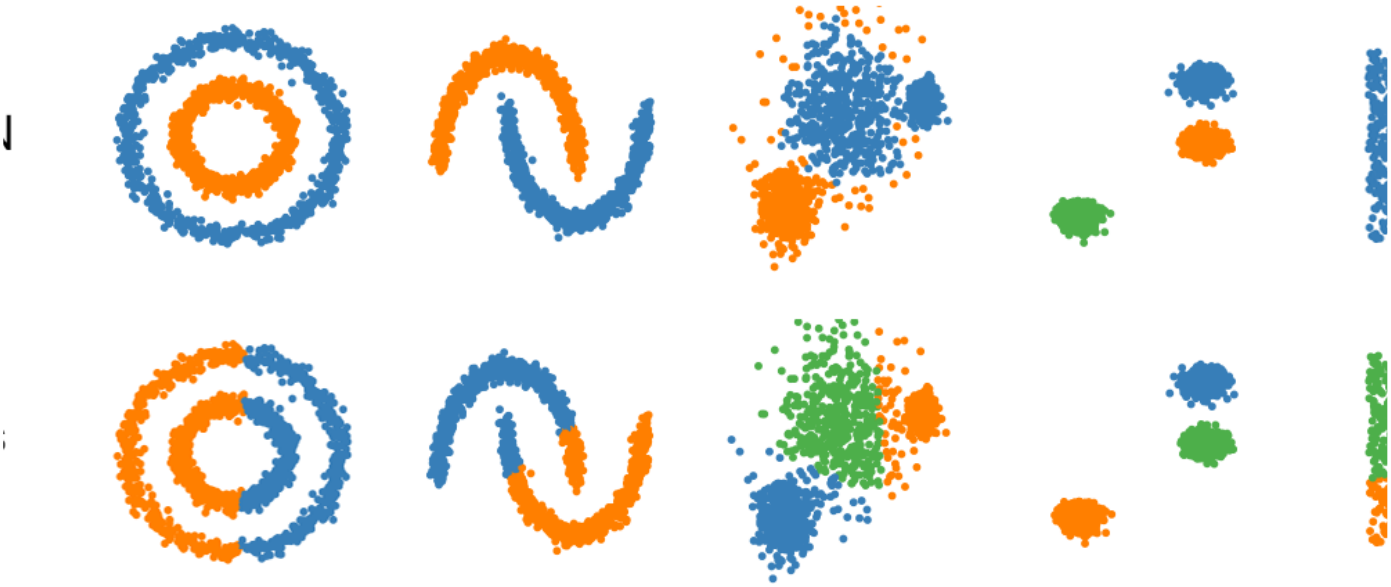


Tarique Akhtar

Clustering using HDBSCAN

HDBSCAN is one of the most popular and more recent algorithm in clustering.

2 min read · Jan 10




Christos Panourgias

Clustering with DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm that groups together points that are close...

4 min read · Mar 27



See more recommendations