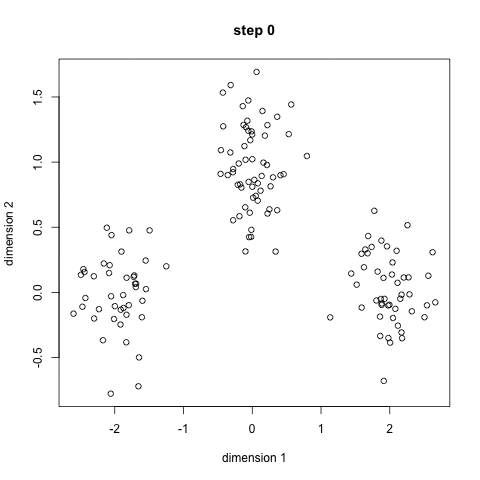
Clustering is a Machine Learning technique that involves the grouping of data points. Given a set of data points, we can use a clustering algorithm to classify each data point into a specific group. In theory, data points that are in the same group should have similar properties and/or features, while data points in different groups should have highly dissimilar properties and/or features. Clustering is a method of unsupervised learning and is a common technique for statistical data analysis used in many fields.

In Data Science, we can use clustering analysis to gain some val­uable insights from our data by seeing what groups the data points fall into when we apply a clustering algorithm. Today, we’re going to look at 5 popular clustering algorithms that data scientists need to know and their pros and cons!

**K-Means Clustering**

K-Means is probably the most well know clustering algorithm. It’s taught in a lot of introductory data science and machine learning classes. It’s easy to understand and implement in code! Check out the graphic below for an illustration.



K-Means Clustering

1. To begin, we first select a number of classes/groups to use and randomly initialize their respective center points. To figure out the number of classes to use, it’s good to take a quick look at the data and try to identify any distinct groupings. The center points are vectors of the same length as each data point vector and are the “X’s” in the graphic above.
2. Each data point is classified by computing the distance between that point and each group center, and then classifying the point to be in the group whose center is closest to it.
3. Based on these classified points, we recompute the group center by taking the mean of all the vectors in the group.
4. Repeat these steps for a set number of iterations or until the group centers don’t change much between iterations. You can also opt to randomly initialize the group centers a few times, and then select the run that looks like it provided the best results.

K-Means has the advantage that it’s pretty fast, as all we’re really doing is computing the distances between points and group centers; very few computations! It thus has a linear complexity *O*(*n*).

On the other hand, K-Means has a couple of disadvantages. Firstly, you have to select how many groups/classes there are. This isn’t always trivial and ideally with a clustering algorithm we’d want it to figure those out for us because the point of it is to gain some insight from the data. K-means also starts with a random choice of cluster centers and therefore it may yield different clustering results on different runs of the algorithm. Thus, the results may not be repeatable and lack consistency. Other cluster methods are more consistent.

K-Medians is another clustering algorithm related to K-Means, except instead of recomputing the group center points using the mean we use the median vector of the group. This method is less sensitive to outliers (because of using the Median) but is much slower for larger datasets as sorting is required on each iteration when computing the Median vector.

pandas as pd

Pandas is an open-source, BSD-licensed Python library providing high-performance,

easy-to-use data structures and data analysis tools for the Python programming language.

Python with Pandas is used in a wide range of fields including academic and

commercial domains including finance, economics, Statistics, analytics, etc.

What problem does pandas solve?

Python has long been great for data munging and preparation, but less so for data analysis and modeling.

pandas helps fill this gap, enabling you to carry out your entire data analysis workflow in Python

without having to switch to a more domain specific language like R.

Combined with the excellent IPython toolkit and other libraries,

the environment for doing data analysis in Python excels in performance, productivity, and the ability to collaborate.

pandas does not implement significant modeling functionality outside of linear and panel regression;for this, look to statsmodels and scikit-learn.

More work is still needed to make Python a first class statistical modeling environment,

but we are well on our way toward that goal.