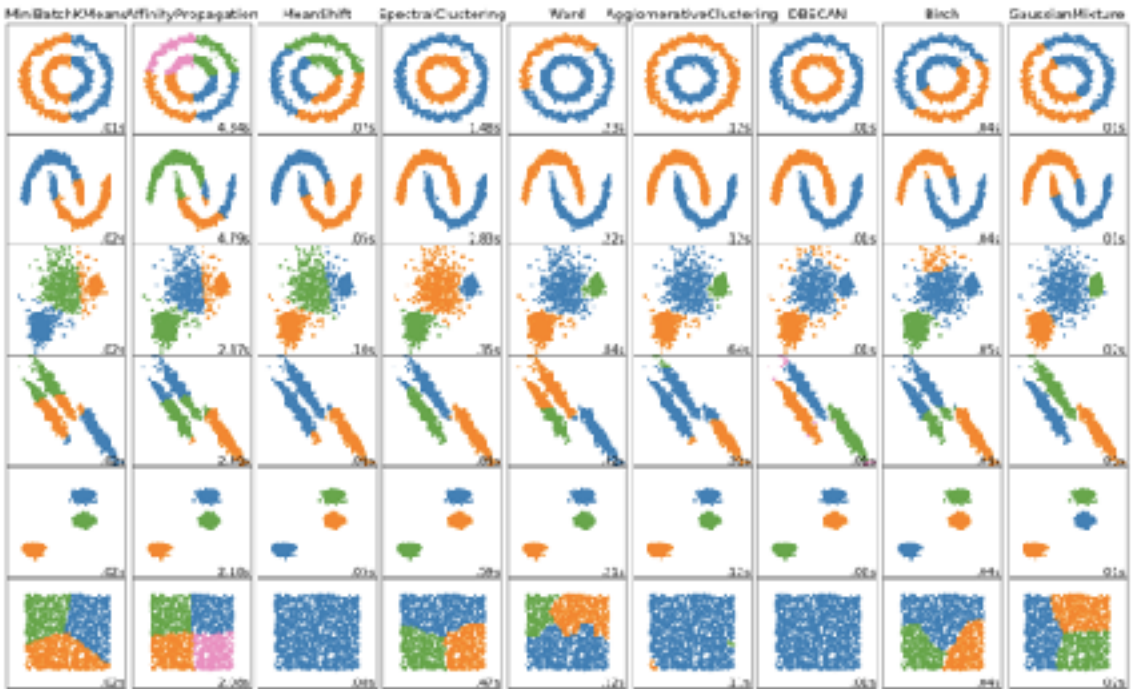
Intro to sklearn: A Whole Bunch of Clustering Models

Abstract

In order to implement Gaussian Mixture Model, I learn about the Python package sklearn, and I find that there are a whole bunch of interesting clustering models. After spending efforts learning about its feature, now I can easily implement any of these models onto our data.

About sklearn

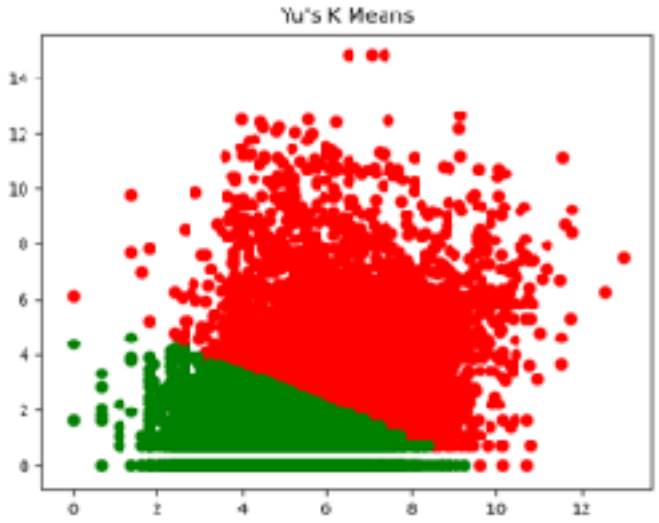


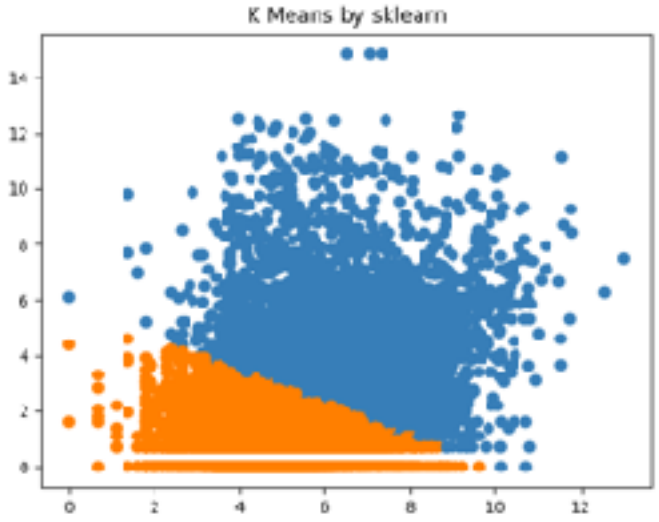
Downloaded image from scikit-learn.org

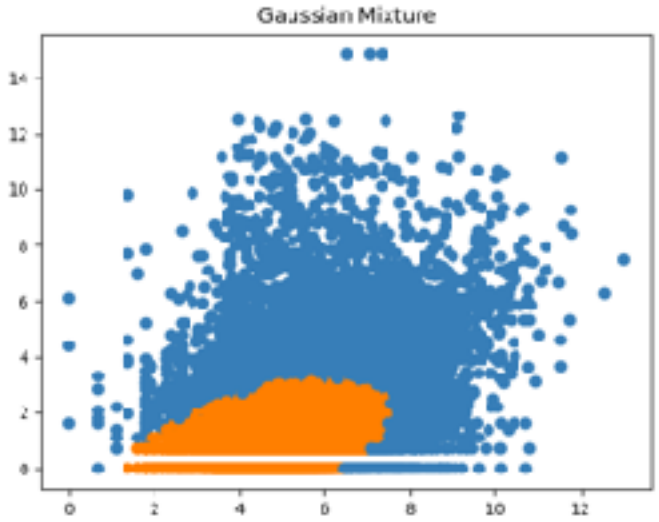
In the picture above, we can see six data sets clustered nine different model, each of them built into the sklearn package and is easy to implement. It is worth mentioning that the six data sets are also generated from sklearn package.

Since we are now bugged by bad clusterings, obviously these models are most valuable to our project.

My Implementation







At the beginning of my collaboration with Andrea, I composed the first image, *Yu's K Means*. At that time, I did not use sklearn, but implemented it with my own code. Comparing image 1 and 2, we can see that my implementation of K means is almost as good as professionals' who codes sklearn. I am genuinely proud of my work.

So now I have the ability to apply all clustering models to our data. Exciting, isn't it?

A Few Words about GMM

Before beginning coding, I did a little research about Gaussian Mixture Model. Here are a few tips:

- Unlike k means, Gaussian Mixture "softly" assign data points to clusters. During the process of clustering, a point can have 70% chance to belong to cluster 1, and 30% chance for cluster 2.
- For non-linear distribution, Gaussian Mixture also works well.
- It does not bias the cluster sizes to have specific structures as does by K-Means.

For more information, please refer to Difference between Gaussian Mixture Model and K means.

Next Step

Now that we have all these clustering models, all we need to do is to apply them to our data. To tell a clustering pattern is good or not, we often use two ways: to look at its image, and to analyze its statistical features.

Here is what I propose:

- Visualization. I can develop a tool to visualize all data clustering, not only for myself but also for you. With this tool, you can see clustering pattern without having to code, just filling in a config file.
- Statistics. I'm not too good at statistics, so it would be up to you to build a quantitative standard for "good" clustering. After you come up with such standard, I can code it if necessary.
- Understanding. I can do some research on all these clustering models, and try to figure out theoretically which ones would be more useful for us.