Data Analytic Lab 6

Introduction

In this lab we were given the task of coding an algorithm for K-Means clustering in Python. The notebook given came with code pre-written for pseudo-random number generation.

Method

K-Means is a classification method that groups points into “clusters”. First it uses some method to create a certain number of “centroid”, these are the center of each cluster, in this case it was done by randomly assigning each centroid to be one of the points within the data set. These centroid are each given an id to identify them and the points in the associated cluster. From there each point calculates its distance from each centroid, and is labeled with the id of the centroid it is closest with. This is calculated with the following equation:

d=sqrt((x2 -x1)^2 + (y2 -y1)^2)

The point is labeled, the associated centroid is updated to reflect the new number of points currently within it. Once all points have been updated, new centroid are calculated from by averaging the x and y coordinates of the all the points within the cluster. These new clusters replace the only ones and the process is repeated. This is done either until a set number of cycles is completed, or until the centroids stop moving between cycles.

Process

Having previously coded K-Means successfully in C, I mistakenly believed that it would be simple to recreate it in python. I was able to get it working sometimes, but had a bug that I was not able to ultimately fix which caused the program to terminate before the centroids stabilized. This happened when one of the centroid coordinated somehow became a NaN object. I spent a long time looking for an error in my equations, thinking that the issue was a divide by zero error. After a long time with the debugger trying to find this, I was unable to. I think the issue may have stemmed instead from my choice to use data frames to make some calculations simpler.