Kevin Eibeck

861227549

Late days used for this assignment = 0

Total late days used = 0

Assignment 3

Change line 5 in the code to the correct filepath

# Question 0: Getting real data

Approach: I use the readlines method followed by the split method to parse the data.

# Question 1: K-Means Clustering

Approach: In my initialize function I use the sample method from the random library and come up with K random numbers between 0 and 149. Then I initialize cluster centroids with the data point corresponding with the random number generated.

I created a while loop that will continually assign and calculate centroids until it repeats a cluster assignment. Afterwards it returns the smallest sum of errors that it calculated along with it’s cluster\_assignments and cluster\_centroids.

For assigning points to their cluster, I use a for nested for loop. The outer loop iterates over each point in x\_input. Then I have another for loop that calculates the distance between each data point and each centroid. Using the variable assignDist to keep track of the closest centroid for each data point.

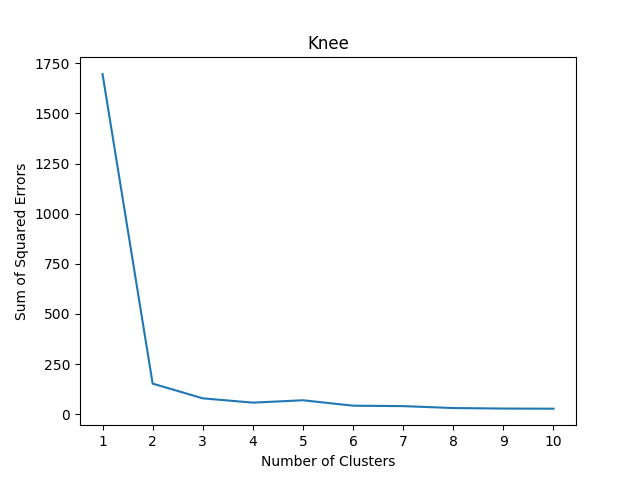
For determining if a cluster assignment has been calculated I use the list allAssignments to keep track of previously calculated assignments. Then I compare each entry in allAssignments to the current cluster\_assignments and if no difference is found then I return the minimum sum of errors, the cluster assignments and cluster centroids.

If the cluster assignment is unique then I append it to allAssignments and go to another for loop that that calculates the new centroid and updates cluster\_centroids

Sum of Squared Errors with (k=3) = 85.2425484135

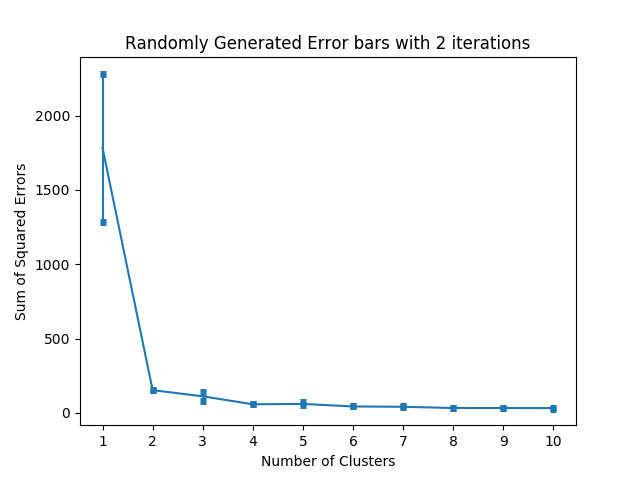
# Question 2: Evaluation

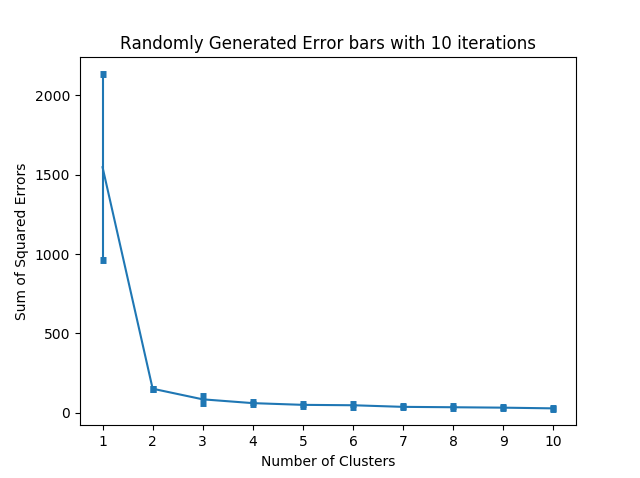
**Knee Plot**

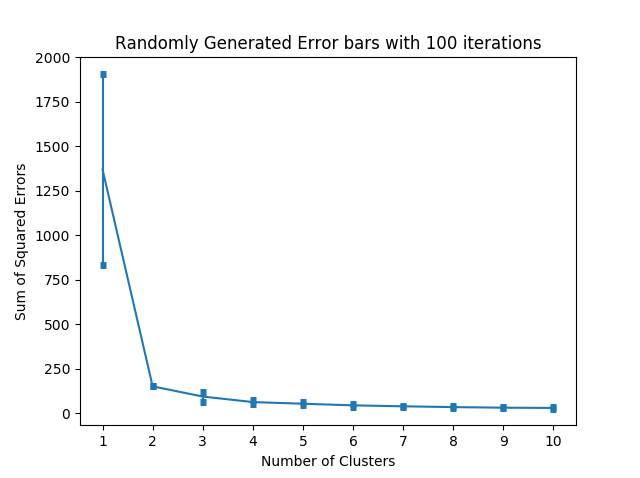


It seems as the knee appears when there are at least two clusters, which would mean that it doesn’t agree with the number of classes in the Iris dataset.

**Sensitivity Analysis**







# Question 3: K-Means++ Initialization

Approach: First I use randint to come up with the first centroid. Then I used a for loop that would iterate over the rest of cluster centroids. I then use a for loop to iterate over every point in x\_input to calculate their distance from their closest centroid. I then square that distance and append it to the list dist. I then use the random.uniform between 0 and the sum of dist and get a random number. I then use a for loop iterating over x\_input, while taking the sum of their distance squared. When that sum surpasses the random number I generated then you’ve found your next centroid. This gives more weight to the farther distance points.

# 

Question 4: Top Data Points per Cluster

Approach: In my function topData I use a for loop that iterates over the K passed in. I then use a priority to calculate the lowest distance points. Since priorityQueue returns the smallest entry, I used negative numbers for the distance. I then used a for loop to iterate over cluster\_assignments. Using an if to compare the cluster\_assignments to the current cluster I’m looking at I then use the distance function and the priority queue to find the 3 closest data points.