Nearest neighbor code:
In order to run the nearest neighbor algorithm, simply open the script and press run. This will print the nearest neighbor classification result, the k-nearest neighbor classification result, the raw data plotted, and the random test case plotted.

K-means clustering code:
Open the KMeansClustering_driver, designate a k value (number of clusters you want created) and press run. This prints a graph of the updated clusters displaying the results from k-means clustering.

File directory:
NearestNeighborClassification contains the functions and script to run the nearest neighbor algorithm.

KMeansClustering_functions contains the functions required for k-means clustering algorithm.

KMeansClustering_driver contains the main script to run and plot k-means clustering algorithm. (this actually also contains the functions because something seemed to be wrong with my path in that no matter what I did I could not seem to import the function module into the driver, so I left them at the top of the driver script.)

Cdk.csv contains the raw data I used to test my algorithms.

Functions:
Nearest neighbor:
openckdfile()

- Reads in the ckd data set
- Takes no inputs
- Returns glucose, hemoglobin, and classification arrays

normalizeData()

- scales glucose and hemoglobin data into values between 0 and one so one is not weighed more than the other
- Takes glucose, hemoglobin, and classification arrays
- Returns scaled versions of glucose, hemoglobin, and classification arrays

createTestCase()

- Makes a random glucose and hemoglobin value between 0 and 1
- Takes no inputs
- Returns a random glucose value and a random hemoglobin value

calculateDistanceArray()

- Finds the distance between every point and the random test case and stores it in an array
- Takes the scaled glucose and hemoglobin arrays and the random glucose and hemoglobin test case

- Returns an array of all the distances between the point and each other point in the data set

nearestNeighborClassifier()

- Finds the minimum distance in the distance array and assigns the test case the same classification as that point
- Takes the random test case, scaled glucose and hemoglobin arrays, and the classification array
- Returns a classification of the test case

kNearestNeighborClassifier()

- Picks a designated number of closest points to the test case and assigns the classification of the majority of these points
- Takes the random test case, scaled glucose and hemoglobin arrays, and the classification array
- Returns a classification of the test case

K-means Clustering:
openckdfile()
normalizeData()
createCentroid() (same as test case)
calculateDistanceArray()

- All of the above are same as nearest neighbor functions

nearestCentroid()

- Assigns each point in the set to one of k centroids
- Takes distance array as the input
- Returns an array of cluster assignments/clssifications

newCentroid()

- Calculates the new locations of the k centroids based on the averages for each point in that cluster
- Takes centroid position array as an input
- Returns new cluster positions

updateCentoid()

- Calls previous functions in sequence to continually update the centroid position
- Takes the scaled glucose and hemoglobin arrays and the centroid position
- Returns the final centroid position