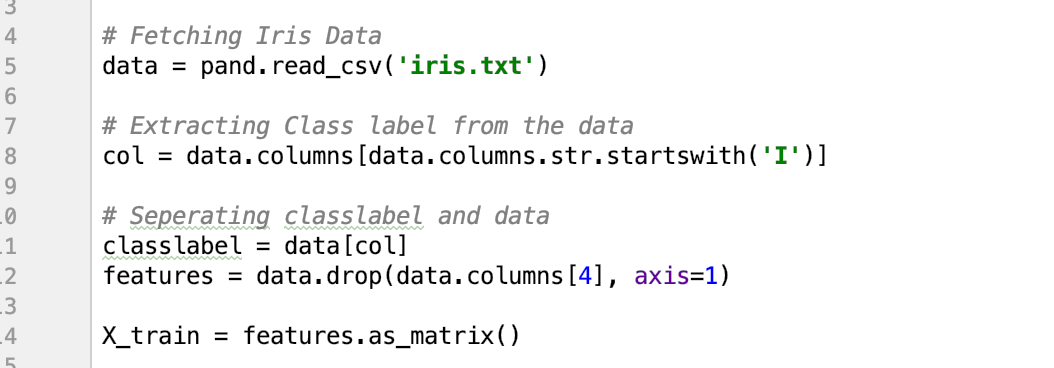
CSE-6363 Assignment 3 Swetha Vijaya Raghavan

1001551229

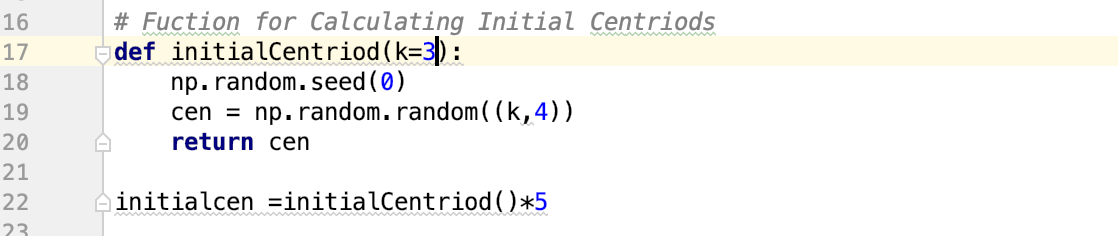
K-means clustering algorithm performed on Iris data.

First data preprocessing, of reading the data from iris.txt and then extracting the class label column from the whole data and X\_train has the all the features excluding class label column.

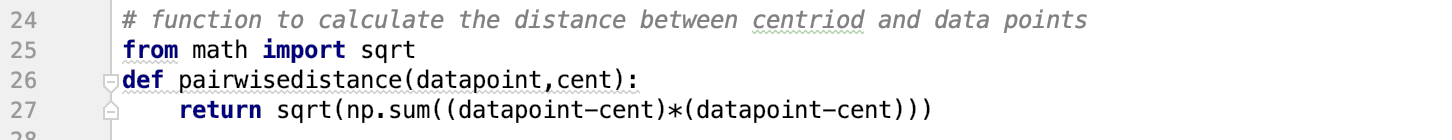


I have used k= 3 because we can evaluate, as there are 3 labels (0 Iris-setosa, 1 Iris-versicolor, 2 Iris-virginica) in the dataset. This can be done intuitively. Also, the data is uniformly distributed over all the clusters. A good clustering with smaller k such has 3 can have a lower Sum of Squared Error (SSE) than a poor clustering with higher k.

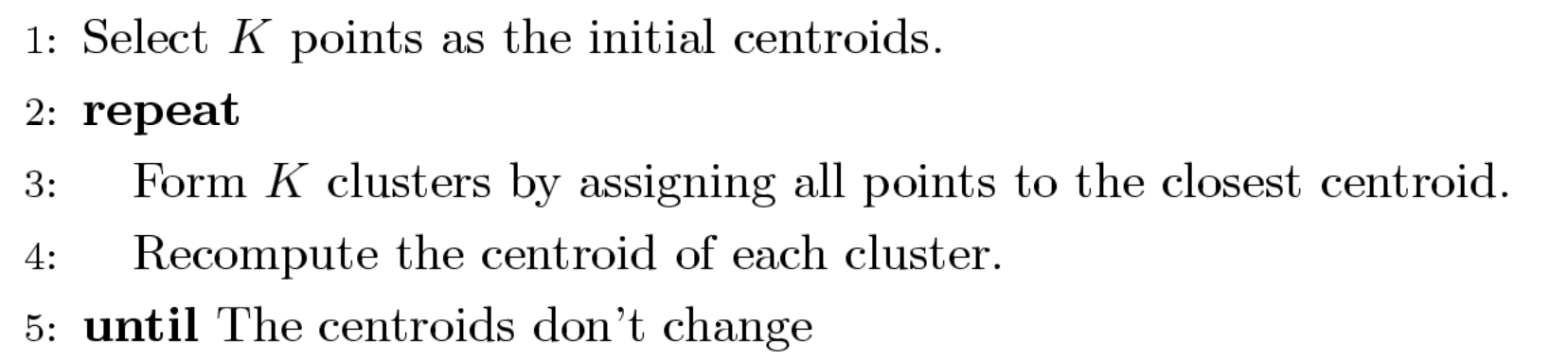
The below function will calculate the initial centroids to start our algorithm.



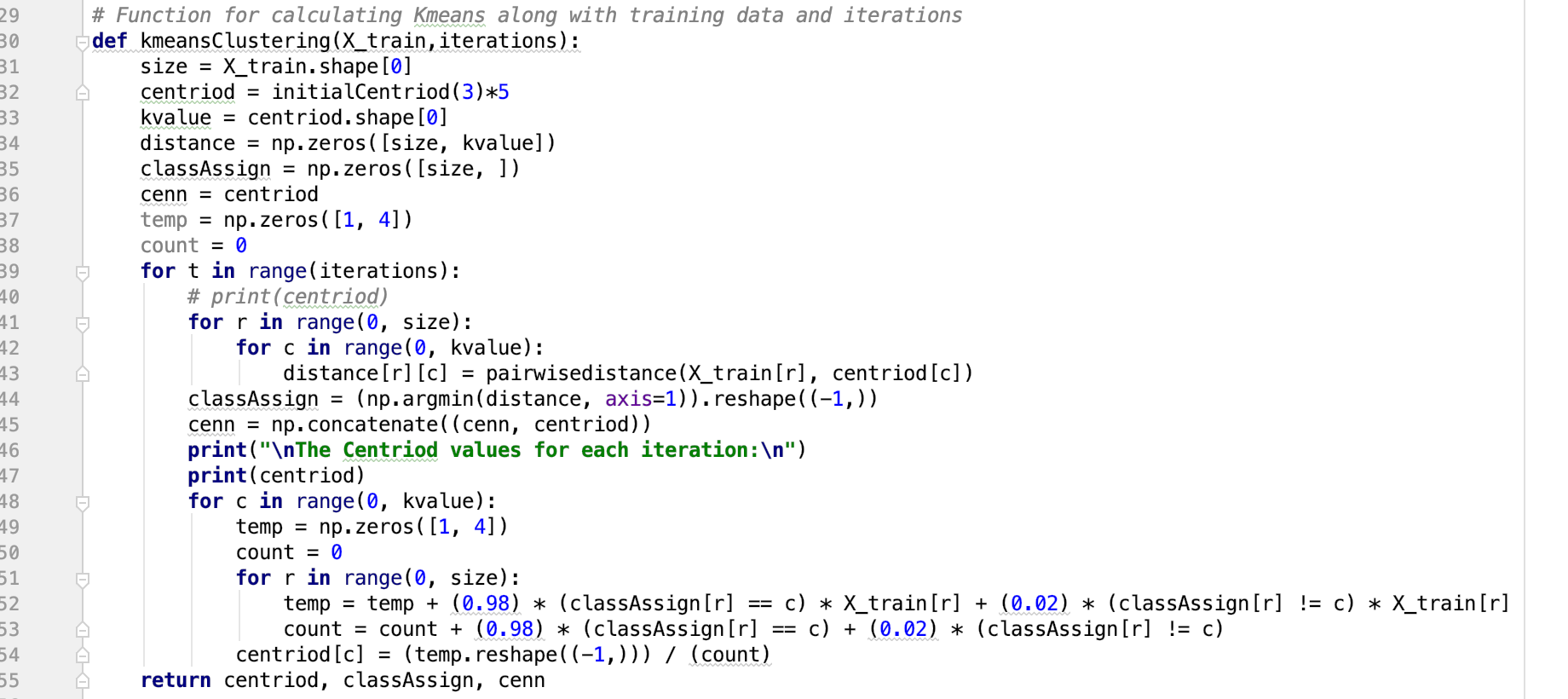
Function to calculate the Euclidean distance from centroid to each data point so as to form a neighborhood which easily identifies which data point belongs to which cluster.



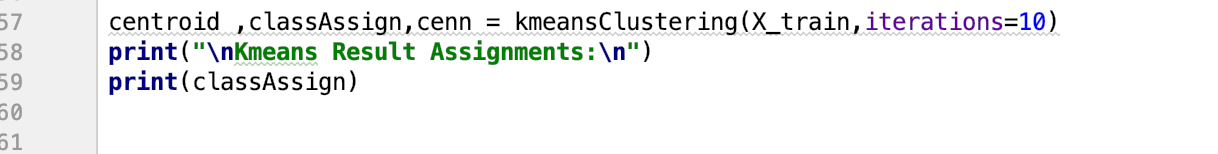
KmeansClustering is the function which performs the kmeans clustering algorithm (mentioned below) using 2 parameters the training data and number of iterations.



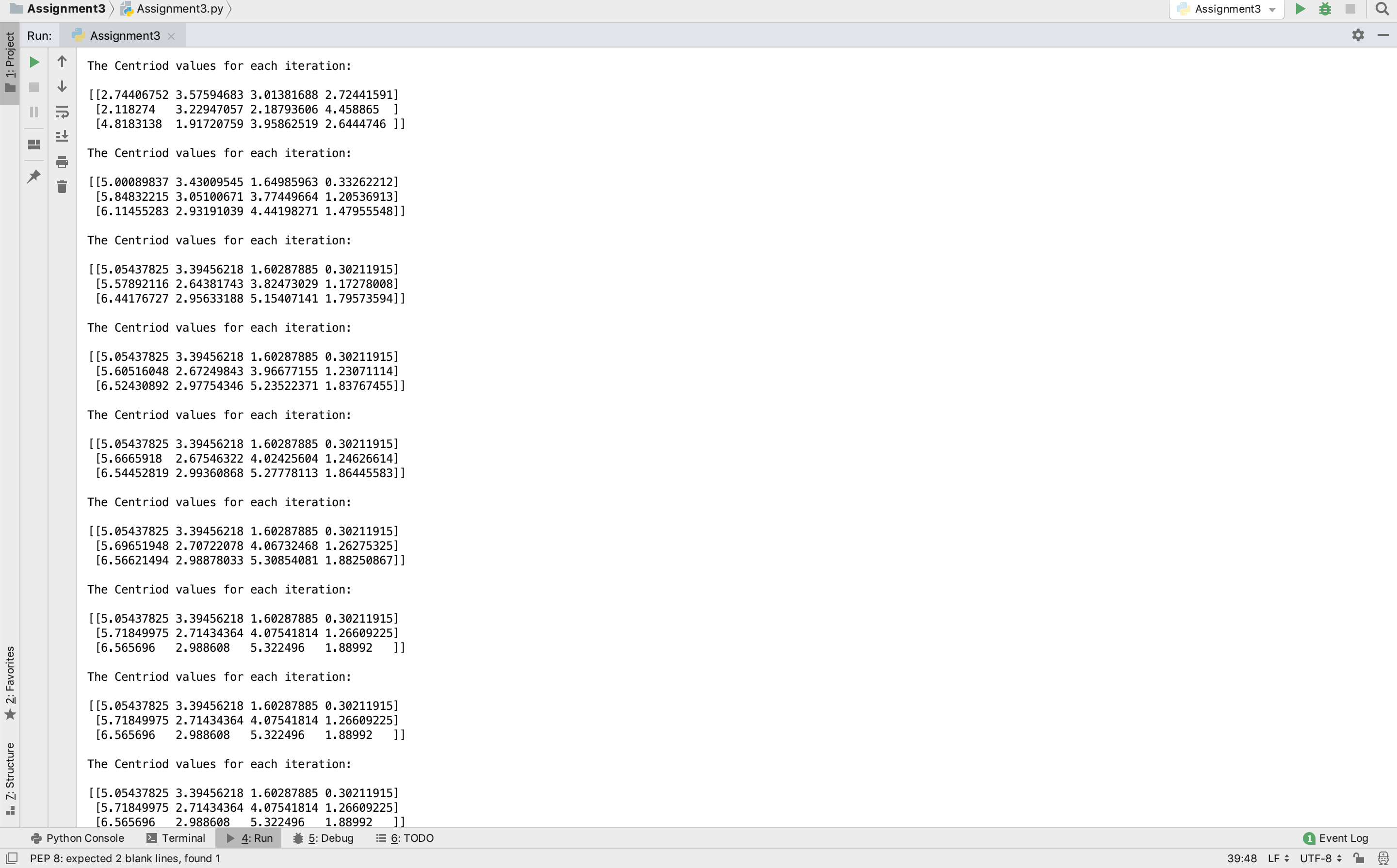
The Complexity of the above algorithm is O( n \* K \* I \* d ) where n=number of points(), K= number of clusters, I = number of iterations, d = number of attributes.



The kmeansClustering results in centroids computed in each iterations along with the assignments of the data points to each cluster.



10 Iterations centroids are shown below:





The result of our algorithm is:



References:

<http://madhugnadig.com/articles/machine-learning/2017/03/04/implementing-k-means-clustering-from-scratch-in-python.html>

<http://benalexkeen.com/k-means-clustering-in-python/>

<https://www.kaggle.com/andyxie/k-means-clustering-implementation-in-python>

GitHub & Wikipedia