Implementation

In this implementation for the initialization of the centroids each random datapoint is found with the ‘rand’ function. The function delivers a pseudorandom number. Through Initialization with the same seed, same results can be obtained. This random number is then multiplied with the number of datapoints minus one and one is added to the result. The addition and subtraction of one is needed in order to prevent over and underflow of the dataset array. To assign the clusters, the program iterates over all datapoints. For each datapoint, an array containing the Euclidian distances to each centroid is computed. The centroid with the least distance is then assigned to that datapoint. For moving the centroids, the mean of each parameter within the cluster is calculated. These mean values become the new centroid. To determine the amount of iterations needed, the previous cluster assignments are compared to the current cluster assignments after each iteration. If the cluster assignment doesn’t change for any point within two subsequent iterations, the loop stops. For Initialization, arrays of ones and zeros are used for the previous and current cluster assignment value. Alternatively, two arrays or scalars that differ from each other can be used. The size of the arrays is chosen to be the number of datapoints in order to speed up the program. Finally, the cluster assignments are combined with the input data and returned to the calling program.