KNN implementation using MRJob

Preparation:

1. Import the required packages such as MRJob, MRStep, re, pandas, and heapq.
2. To skip the first row (header row) and the unknown samples, which are called query here, of the csv file, the pattern (r'^[0-9].*[a-z]$') is defined.
3. To normalize the features of the queries to be able to use them in the KNN algorithm, "Iris.csv" file is converted to data frame (lines 10-26 of the "map_reduce_knn.py" file).

MapReduce job:

1. For each sample the mapper produces 4 pairs, each of which has 2 elements. The first element is a pair of ID and label ([ID, label]). The second element is squared difference between each feature of every sample and corresponding query feature.
2. The combiner aggregates so-far squared differences for each ID.
3. The first reducer ("reducer_get_EuDis") complete the task of combiner and produces pairs like below for each sample:

   (Euclidean distance from query, (ID, label))

4. The second reducer ("reducer_find_nearestSamples") find the nearest samples to the query and create a list of labels of the 15 nearest samples and finally yield the frequent label plus the ID of the query, (query ID, predicted label).

Commands to be executed in terminal:

1. To predict the label of each query, the corresponding ID of the query should be written in the lines 39 and 57 of the "KNN/map_reduce_knn.py" file, and the result are stored in a text file called "KNN/predicted_labels.txt" using the command below:

   "python KNN/map_reduce_knn.py KNN/Iris.csv > KNN/ predicted_labels.txt"