Robert Colvin (862060875)
CS226 Fall 2019
Programming Assignment 2

1. I used InputFormat of FileInputFormat for the MapReduce program to simply give it a path to the input file.
2. The input format for map is the string representation of the Text object that represents the input file (ie. the input format is Text, which is then converted to a String which is what the work is done on). The output format is key-value pairs of <Text, DoubleWritable>.
3. KNN (without pruning or some sort of dimensionality reduction) requires that we calculate the distance to every point in the dataset; the logic of the map function is to calculate each point's Euclidean distance from the given query point and emit key-value pairs of the form <PointID, distance>.
4. No combiner function is used.
5. I wrote a class called KNNPoint that simply bundles pointID and distance together. The Reducer class maintains a static PriorityQueue (MaxHeap) of KNNPoints. Each point is added to the heap until it is size k, at which point I check to see if the current KNNPoint's distance is less than the farthest distance in the heap; if so, I remove the farthest distance from the heap and add the current point. Since the reduce method runs once for every key and, as previously stated, I need to check every data point to figure out the k nearest, the reduce function does not perform final results. The cleanup function runs only once per reducer and by the time it's done reducing I should have a heap of size <= k, I used the cleanup function in the Reducer class to actually write to disk the final results: pointID and distance from the query point.
6. Number of mappers can vary. Number of reducers is set to one to simplify output management, but could be allowed to vary if I managed to set k to decrement after each result's write to disk.
7. Number of records shuffled between mappers and reducers: 10507403