Give It Some Thought

3. Design an algorithm that uses hill-climbing search to remove *redundant examples*. Hint: the initial state will contain the entire training set, the search operator will remove a single training example at a time (this removal must not affect behavior).
4. Describe an algorithm that uses hill-climbing search to remove *irrelevant attributes*. Hint: withhold some training examples on which you will test 1-NN's classifier's performance for different subsets of attributes.

Computer Assignments

1. Write a program whose input is the training set, a user-specified value of k, and an object, $\mathbf{x}$. The output is the class label of $\mathbf{x}$.
2. Apply the program implemented in the previous assignment to some of the benchmark domains from the UCI repository.[5] Always take 40 % of the examples out and reclassify them with the 1-NN classifier that uses the remaining 60 %.