Your task is to create a more efficient classifier for the MNIST task using oblique decision trees.

Do the following:

1) Implement a version of the ID3 algorithm that uses linear combinations of attributes in each internal node. To choose the linear combination associated with a node processing data D, try a number of random projection vectors x (of the same size as the attributes), find the optimal threshold t for each of them, and chose the combination of x and t that generated the highest information gain (or, equivalently, the smallest weighted entropy). The pseudo code of the algorithm would be as follows:
   a. If more than X% of examples in D belong to the same class, return a leaf node indicating that class
   b. for i=1:n
      i. xi = rand(784,1) // Random vector of length 784
      ii. find optimal threshold ti, such that the binary attribute D xi < ti has maximum information gain
   c. Use the combination of xi,ti that yielded the largest gain in the previous step to split D into Dl and Dh, and apply recursively to Dl and Dh.

2) Implement an ensemble of classifiers generated using the algorithm from item 1.

3) Evaluate the performance of your individual classifiers as well as ensembles of various sizes using the MNIST dataset

4) Write a report describing your results and include your source code as appendix. I am particularly interested in your accuracy and running times as a function of the various parameters in the algorithm.

5) Extra credit: Propose improvements to the algorithm described above.