Spam email messages are a major problem on the internet. This assignment explores how to construct a spam filter from email messages that have been identified as spam or ham.

1. Use Java (C or C++) to implement:

   (a) Decision Tree algorithm: `DecisionTree.java` has the `main` method

   (b) Decision Tree algorithm, limit the tree to have one fewer level than the first algorithm: `DecisionTree2.java` has the `main` method

   (c) Preprocessing: each attribute is a word (lowercase without punctuation) and has a value of Y (in the email) or N (not in the email); spam and ham are the two classes

   (d) Extra Credit (30 points): k-nearest neighbor algorithm: `KNN.java` has the `main` method, (k=1 for toy data set and k=3 for sa data set; Hamming distance–0 if same attribute value, 1 if different, sum over the attributes) [similar top-k file as in HW3]

2. Input:

   (a) email file

   (b) quiz file

3. Output:

   (a) screen:

      • Accuracy (percentage with 2 decimal places) on the email file
      • Accuracy (percentage with 2 decimal places) on the quiz file

   (b) tree file: human readable tree

   (c) email prediction file: *emailID correctClass predictedClass*

   (d) quiz prediction file: *emailID correctClass predictedClass*

4. Provide a report (pdf):

   (a) Compare the two algorithms:

      i. Accuracy performance
      ii. time/speed to construct and use the tree
      iii. space/memory

5. Provide `readme.txt`

   (a) how to compile your programs

   (b) how to run the two algorithms

   (c) sample output of each algorithm for each input data set

6. Submit: source code, report, and `readme.txt`