## Lab 3: Wikipedia Language Classification

To run the code:

* train <examples> <hypothesisOut> <learning-type>
  + examples is a file containing labeled examples
  + hypothesisOut specifies the file name to write your model to.
  + learning-type specifies the type of learning algorithm you will run, it is either "dt" or "ada".
* predict <hypothesis> <file>
  + hypothesis is a trained decision tree or ensemble created by your train program
  + file is a file containing lines of 15 word sentence fragments in either English or Dutch.

Feature Description:

1. Is the average word length > 5?

Average len = sum of word lengths / total number of words

True if average len > 5 else false

1. Is an English article present?

Checks the presence of English articles *an* and *the*. (‘a’ is present in both Dutch and English)

True if present else false

1. Is a Dutch dipthong present?

Checks for the presence of Dutch dipthongs

True if present else false

1. Is Dutch common word present?

Checks for presence of Dutch common words

True if present else false

1. Is English common word present?

Checks for presence of English common words

True if present else false

1. Are non-English characters present?

Checks for presence of non-English characters

True if present else False

Decision Tree Training Algorithm:

Information gain is used to determine the best split using a split function. Then we recursively build the tree on the 2 subsets (true and false) until we hit a leaf node (information gain = 0 or the max depth = no of features = 6). At the leaf node, En or Nl is assigned to the node as target label.

AdaBoost Training Algorithm:

Graphical user interface, text

Description automatically generated

For learning algorithm, L, we use the decision tree constructed earlier with decision tree height as 1.

Test cases:

|  |  |  |  |
| --- | --- | --- | --- |
| Test file | Train file | Decision Tree | AdaBoost |
| Provided train.dat | Provided test file | 100% | 100% |
| Train\_1k.dat | Provided test file | 100% | 100% |
| Train\_500.dat | Provided test file | 100% | 100% |
| Provided train.dat | Test.dat | 95% | 94% |
| Train\_1k.dat | Test.dat | 99% | 98% |
| Train\_500.dat | Test.dat | 99% | 98% |

Extra:

I find it interesting that the code worked this well. Maybe the test cases I had were not good enough or too few or had no edge cases?

Training of ada takes longer than dt. Please bear with it.