CSCI630 Lab 3 Report

Running the code –

Execute file lab3.py as per the usage mentioned on the website –

- **train <examples> <hypothesisOut> <learning-type>** should read in labeled examples and perform some sort of training.
    - 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". You should use (well-documented) constants in the code to control additional learning parameters like max tree depth, number of stumps, etc.
- **predict <hypothesis> <file>** should classify each line as either English or Dutch using the specified model.
    - 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.

Decision Tree –

Files ‘examples.dat’ and ‘train.dat’ contain the training data for creating ‘best.model’ which is the trained model for the decision tree.

File ‘decision.py’ can be referred for the code.

Features Chosen –

We scan each line word by word and compile a dataframe for features cumulatively at the end of the sentence.

1. Dutch Article: A sentence can be classified as dutch if we encounter a word that is for sure a dutch article. Once we encounter such word, we do not need to check on this attribute again for the rest of the sentence

2. Dutch Pronoun: A sentence can be classified as Dutch if we encounter a word that is for sure a dutch pronoun. Once we encounter a dutch pronoun we can stop checking on this attribute.

3. Word ending with ‘ij’: Dutch language has a very high probability of words ending with ‘ij’ than English. Hence once such a word is encountered, we can stop checking on this attribute for the rest of the sentence.

4. Consecutive duplicate letters: Dutch words have a high probability of words having same consecutive letters in them. But the word may fall into common English words as there are some words that adhere to this criterion. Hence this feature is checked for each word in our sentence.

5. Average word length in a sentence: Average word length in a dutch sentence is more than 5. Hence, we calculate this attribute once for the entire sentence and can classify it as dutch.

6. Dutch Common Words: Check once for this feature. Sentence can be classified if the word is a common dutch word. List of words gathered from Wikipedia.

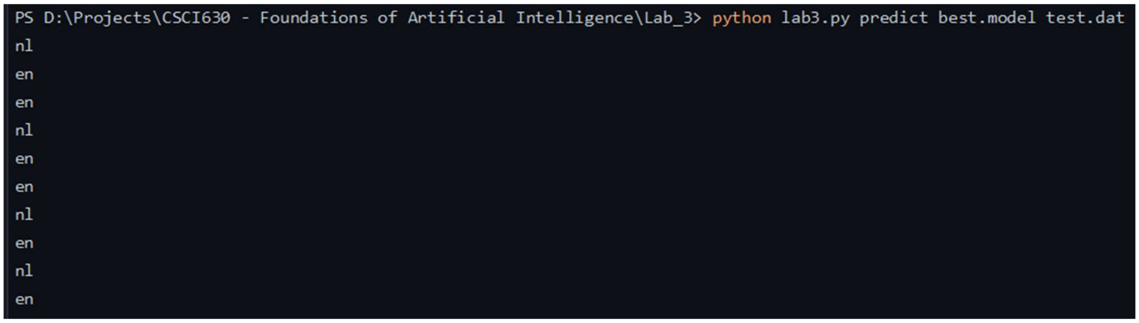
7. English Article: check if the word is an English article.

Testing Results –

Each decision tree level is selected based on the information gain for that feature. More often than not for my model the first split was done on the attribute which checks for Dutch articles, followed by Dutch Pronouns.

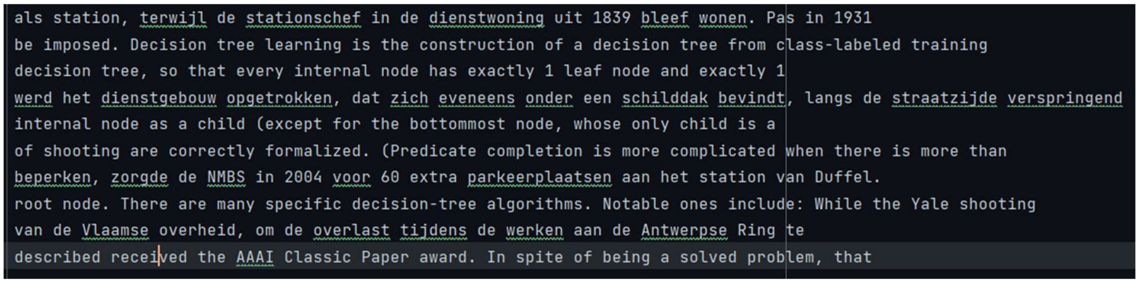
Every line of file test.dat was successfully classified.

Output -



```
PS D:\Projects\CSCI630 - Foundations of Artificial Intelligence\Lab_3> python lab3.py predict best.model test.dat
nl
en
en
nl
en
en
nl
en
nl
en
```

Test.dat -



```
als station, terwijl de stationschef in de dienstwoning uit 1839 bleef wonen. Pas in 1931
be imposed. Decision tree learning is the construction of a decision tree from class-labeled training
decision tree, so that every internal node has exactly 1 leaf node and exactly 1
werd het dienstgebouw opgetrokken, dat zich eveneens onder een schilddak bevindt, langs de straatzijde verspringend
internal node as a child (except for the bottommost node, whose only child is a
of shooting are correctly formalized. (Predicate completion is more complicated when there is more than
beperken, zorgde de NMBS in 2004 voor 60 extra parkeerplaatsen aan het station van Duffel.
root node. There are many specific decision-tree algorithms. Notable ones include: While the Yale shooting
van de Vlaamse overheid, om de overlast tijdens de werken aan de Antwerpse Ring te
described received the AAAI Classic Paper award. In spite of being a solved problem, that
```

AdaBoost-

Script adaBoost.py can be used to review code.

We start with creating stumps for each feature from our data frame. From that we choose the best stump depending on the least entropy among all the stumps. From testing more often than not stumps based on root node as Feature 1 (Dutch Article) and 6 (Dutch Common Words) were useful.

examples.dat was used to train the adaBoost model and test.dat was used for prediction.