**Foundations of AI – Lab2 - Writeup**

**Moinuddin Memon :mm6976@rit.edu**

**Feature Description:**

**1.Average Word Length of Sentence:**

Sum Of word lengths / Total Number of Words

If this property is below the threshold value of 5, it is set to True

Else, False.

2.**Average Consonant Counts in a Word in row**

Sum of Occurrences of consonants without a vowel / total number of words

True if the value is above 4.3

Else, False

**3.Occurence Of ‘the’ in the Sentence**

True if ‘the’ is present

Else, False

**4.Occurence of Dutch Common Words**

True if present

Else, False

**5. Occurrence of English Common Words**

True if present

Else, False

**Decision Tree Training Algorithm:**

* Get the Input Data in the form of a Feature matrix after extracting all the features of the input Data.
* We will find the Column with the best Information Gain using the Following Formula:

IG(column,Dataset)=Probability(column=True)\*entropy(target,true\_split)+Probability(column=False)\*entropy(target,false\_split)

Where entropy=-(probability(column=Val) \* log(probability,2))

* Probability (column=Val) is the probability of Val in the column.
* The column with the Maximum IG is used to split the data set into two subsets based on the truth value of that particular Column.
* This is done using a split function which return true and false tuples.
* Then using recursion we call the build tree function on the 2 subsets until the IG value returned is 0.This signifies that we have reached a Leaf node of the decision tree. There we select the target label as En or nl based on the number of occurences in the Leaf.

**AdaBoost Training Algorithm**

In contrast to a single tree in decision tree learning, we use a forest of trees of unit height to make a prediction. These trees are called stumps.

* Extracting all the features in a feature matrix .
* At first all the tuples are given a weight of 1/length of Feature matrix.
* Again we calculate the best column to split the dataset on.
* We call this a Stump and calculate the error and update the weights accordingly. i.e If correctly classified we decrease the weight else we increase it and then we normalize.
* This is done for K iterations and K stumps are returned at the end
* K is the number of features
* After all k iterations , we return k stumps and their z values as hypothesis

The following image shows the exact algorithm we implemented.

Text, letter

Description automatically generated

**Adaboost Prediction:**

Extract each row and classify them based on the weighted majority of z values of stumps.

**Decision Tree prediction:**

Extract each row and classify them based on the numbers of each label count present in the Leaf node.