Foundations of Artificial Intelligence
Lab 2

Nishi Pawan Agrawal (na1748)

The following write up is done with as per the lab2 instructions. This is the implementation and training and predicting data through the process of decision tree and adaboost. This implementation uses the following approaches to identify if the given input line is in Dutch or English.

Attributes:
In this implementation, I use 7 attributes to identify the key differences between English and Dutch. These attributes are as follows:

- Attribute 1

The Dutch script does not have a letter 'q' in most of the cases. But, in English the letter q is used at a better frequency.

- Attribute 2

The Dutch script does not have a letter 'x' in most of the cases. But, in English the letter x is used at a better frequency.

- Attribute 3

An English sentence has an average word count of 5 words. So, if the word has length of greater than 5, we say that it should be an English word. We calculate the average length of words in the sentence.

- Attribute 4

The word 'van' does not appear as frequently in the English script as it does in the Dutch sccript.

- Attribute 5

The English script uses the prepositions 'a', 'an', 'the' in most of the cases. These words do not appear in the Dutch Script.

- Attribute 6

In this attribute, we are looking for conjunctions in English like 'and' which do not appear the the Dutch language.

- Attribute 7

The English script does not have the words 'de', 'het' in most cases. But, in Dutch we use these words at a better frequency.

Decision Tree

Decision Tree is a learning technique in AI where we construct a tree to predict on the given input data. Based on the training data that we receive, we run through the consolidated attributes and formulate the training attributes into a list of lists. Each list inside of the list corresponds to one line of the input statement that we receive.

The next step is to calculate the entropy and information gain. We do this by going through all the attributes and we select the attribute which has the maximum information gain. Then we perform a split based on the true and false values of the selected attribute and repeat the same procedure until we reach a leaf node. When the maximum values of information gain is 0, we encounter a Leaf node. This is the base case for our recursion.

Adaboost

In adaboost, we are finding the attributes in a similar way we find it for the decision tree. Only difference is that the adaboost data has an additional list which holds the weight of each sample. First we go through the learning process. Initially, we take the weight of each sample as 1/Number of samples. (The sum of all weights should be equal to 1). We then create a new list which has the cumulative values of this list. The last value is 1. Then we take a random value between 0 and 1 and check which values in this new list corresponds to this in the dataset. We then get the position at which this dataset has the highest information gain. We return this new dataset, the weight list and the position of best split.

In the second step, we fill the tagret list with all the values of target from our current dataset. We then iterate over the dataset and check which values are not equal to the target and update the error value accordingly. In the next step, we repeat the same procedure, except this time, we check which values are equal to the target and update the dataset accordingly.

We then normailze this dataset and append the performance measure in a new list. This method repeates 7 times as there are 7 features that I have used. This will finally return a list of tuple in which each tuple has values: dataset, the list of performance measure and the position of best split for that dataset.