Lab 3

CSCI 630 - Foundation of Artificial Intelligence

Rutvik Pansare - rp2832@g.rit.du

1.The features selected to create a decision tree were:

1. Average word length – If the average word length was greater than 5 in the sample, then the function would return true, or else false. The intuition behind this feature is that Dutch words are usually longer in length and hence if the average length goes above 5, it is safe to assume that the sentence is in Dutch.

2. Contains Q – Checks if the sample sentence contains Q or not. The probability of the letter Q occurring in the Dutch language is 0.01% hence if the letter Q occurs in a sample sentence, it should be safe to assume that the sample sentence is in English. The method returns true if the sample contains Q, else it returns false.

3. Number of "ie" in the sentence - "ie" means "he" in Dutch and is a pretty common in Dutch sentences. If a sample sentence contains "ie" in it, it should be safe to assume that the sample is in Dutch. The function returns true if the number of "ie" are greater than 1 in the sample sentence.

4. Number of "de" in the sentence - "de" means "The" in Dutch and is common in Dutch sentences as well and is one of the most used Dutch word. If a sample sentence contains "de" in it, it should be safe to assume that the sample is in Dutch. The function returns true if the number of "de" are greater than 1 in the sample sentence.

5. Number of Dutch pronouns greater than 1 – This feature returns true if the number of Dutch pronouns in the sentence are greater than 1, else it returns false. Dutch pronouns and English pronouns are totally different from each other and can be a good way of differentiating between the two languages. This feature checks of the most common Dutch pronouns are present in the sample or not.

6. If word that are unique to Dutch are present or not – This feature returns true if the sample sentence contains words that are unique to Dutch language, or else it returns false. It checks for the following words – "en","aan","als", "voor" ,"de", "die", "habben","ze", "het". The probability of any one of these words occurring in the sentence is very high and hence can be used as a classifier.

7. If word that are unique to English are present or not – This feature returns True if the sample sentence contains words that are unique to English. It checks for the following words – "and", "to" ,"as" ,"for" ,"the", "were", "which" ,"have", "they". The most common English words can be found at - https://en.wikipedia.org/wiki/Most_common_words_in_English , The probability of

any one of these words occurring in the sentence is very high and hence can be used as a classifier.

2. The decision tree classifier is a tree like classifier that has conditional control statements. The decision tree has a root node from which the split begins and has a combination of parent and leaf nodes. The parent nodes are nothing but different features on which the split Is created for decision making. The tree uses the above 7 features to split the tree into sub – branches. The features that provides the maximum information gain is selected as the root node. The method "find_importance" helps in doing this by calculating the entropy of each feature split. The feature max_depth helps in deciding the maximum depth to which the tree will have incase the depth is increasing on a larger value because of impure sample data, this was kept to 5 as default. The decision tree also had another parameter called min_sample_split which specified the number of samples that were allowed for the tree to be further split into branches.
Results -
The best feature that had the highest information gain was the 7th feature which checked if the sample sentence has words unique to the Dutch language or not.
And the Maximum depth that was best for prediction was found to be 4.
The decision tree had an accuracy of 100% when it was tested on the 10 samples provided in the Lab 3 description page.

3. Adaboost is a boosting technique used as an ensemble learning in Machine Learning. Decision stumps were created with depth 1 using a modified version of decision tree where the maximum depth allowed was 1. For each decision stump, the error was calculated based on how many incorrect predictions were made by the stump. Then based on the error value the amount of say of each stump was calculated. Amount of say determines the significance of each stump in the final prediction of test sample. After the amount of say is calculated the weights of each sample is updated according to the correctness of the sample's prediction using the method "Updateweights" and a mew data set was formed which constituted of incorrectly guessed samples majorly. This was done by the method "getSplit". This is calculated for every decision stump using the fit() method. Once the model is trained, the test samples are passed to the predict() method which tests the sample on every decision stump created and favors the stump's answers which had higher "Amount of say".
Results –
The Adaboost() classifier predicted the 10 sample sentences with 100% accuracy and the most informative stumps were the ones with 1st i.e. "Average word length" and the 6th stump i.e., "Contains_words_unique_to_Dutch"

4. Deciding the stumps was the main task in decision trees. Initially the features were looking for at least 5 unique words of either English or Dutch in the sentence but testing found that finding only 1 unique word was a good indicator of the language of the sample. Hence the features only look for at least a single unique word. The "contains_Q" feature was not of much help because even the word Q is not frequent in the English language.