**Selection of Features:**

* To select Distinct features, I am using grammar rules to indicate whether my given training data contains those features or not.
* If the given data in training set contains those features, I am considering that the language is English language
* If the data in given training set does not contains those features, then I am considering the language as Dauch.
* I am using below features to identify the sentence contain those features or not.
  + Article: a, an, the
  + Pronoun: I, me, you, etc.
  + Preposition: under, above, of, to, etc.
  + Conjunction: and, but, not, etc.
  + Auxiliary: am, are, is, etc.

**Description of Decision Tree:**

* Based on my features I am generating decision tree. First, I am calculating reminder of each feature, in given training set data and then selecting the feature that has minimum reminder compared to other features.
* After finding out feature which has minimum reminder then I am splitting the data based on that selected feature if that feature is true means that it is English language it will go to the left of my decision tree otherwise it will predict its Dauch end go to right side of the decision tree.
* This process will continue until there are no new features left to split or I am getting the minimum reminder comparing all the features is 1 which means that there is no new gain of information possible by splitting the given set of data based on feature.
* The maximum depth of the given decision tree is based on the number of features we have to identify the given test data.
* For our case the maximum depth is 5 because we have at max 5 distinct feature on which we can split our data or predict the language.

**Decision Tree Testing Results:**

* I have tested the given decision tree implementation on 100-line test data. Out of which 50 are Dauch and 50 are English.
* Out of 50 the decision trees predict **42** are English and out of 50 the decision tree predicts **50** Dauch.
* I have also submitted the test file on which I have tested my results.

**Description of Boosting:**

* For boosting algorithm, I am using the adaboost algorithm and I am using the weighted decision tree as the learning algorithm in the adaboot algorithms.
* Then I am calculating error based on my leaf nodes so if I say that that leaf Noad is true and I have as set that is of Duch that means that that is incorrectly classified and it should be and error in my classification.
* After calculating error, I am generating error correction factor on which I need to decrease the weight of my corrected classified set of values.
* After calculating correction factor the new weight are generated and then I am regenerating new weights and then calculating the hypothesis weights
* After that I am normalizing the weights and again running the adaboost algorithm till the time I am not reached to k.

**how many trees turned out to be useful?**

* For my algorithm I have found out that k=10 turned out to be usefull.

**own testing**  **:**

Accuracy of adaboost is 92%