CSCI 630 Lab3 Writeup- Michael Lee (ml3406)

- Training set: 150 examples; Testing set: 15 examples
- Features:

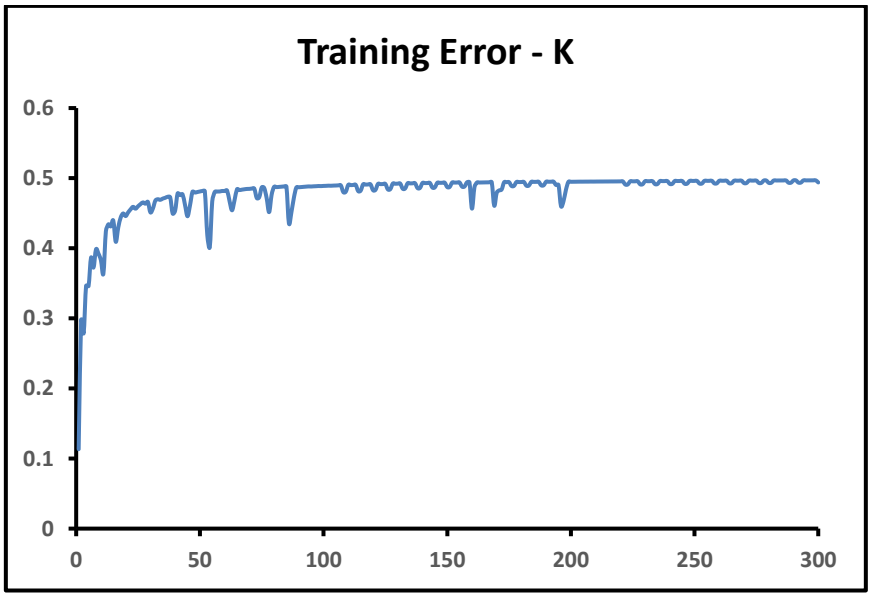
1. Does the sentence (example) contain the word "de"
   Reason: “de” is common in Dutch
2. Does the sentence contain the word "van"
   Reason: “van” is common in Dutch
3. If there are at least two words longer than 10 characters
   Reason: words in Dutch seems longer in average
4. If the sum of ASCII values of all characters in the sentence >= 8000
   Reason: from 3. => the sum of ASCII of Dutch should be larger
5. If there is at least 1 word starts with “z”
   Reason: there are few words start with “z” in English
6. Is the average ASCII value of all words’ first character >= 100
   Reason: no specific reason
7. Does the sentence contain the word "are"
   Reason: “are” is common in English
8. Does the sentence contain "zi"
   Reason: I saw many “zi” in Dutch
9. Is the average word length > 7
   Reason: words in Dutch seems longer in average

- Parameters were found by try-and-error. Error rates of different learning algorithms with different max-depths are shown in the table on the next page. As you can see in the table, the only difference between dt and Ada happens when max depth = 1, which shows that boosting does help the performance of a weak learning algorithm (decision stump is a weak learner because it classifies the examples with only one feature, in which the accuracy depends on the data we use and the feature we choose). Decision tree learning algorithm becomes more robust as the max depth grows. In my case, a decision tree with max depth 2 is getting an accuracy of 100% on predicting

my testing set. (My features are fairly distinguishable between English and Dutch, and thus, my decision tree with max depth of 2 becomes a good learner.) The influence of boosting becomes less obvious while applied to good learning algorithms. If we use some other weak learning algorithms, boosting might become more impactive.

Max-Depth	Decision Tree	AdaBoost (K = 20)
1	1/15	0/15
2	0/15	0/15
3	0/15	0/15
4	0/15	0/15
5	0/15	0/15
6	0/15	0/15
7	0/15	0/15
8	0/15	0/15
9	0/15	0/15

- As you can see in the graph below, 50 trees (K) turn out to be useful (approaching a training error of 0.5). I used K = 20 according to the textbook (R&N 18.10). (By try-and-error, 4 trees is enough to get a 100% accuracy in my case.)



P.S. max-depth is hard-coded in the constructor of my main class.