Determination of Authorship

by: Aseem Mehta

1. Problem Statement:
   Code is supposed to predict the accuracy of the model using both Decision Tree and Logistic Classifiers. Features need to be selected which will help with providing a good accuracy of the model.
   Files included: README, lab2.py, ACD.txt, HM.txt, JA.txt

2. Features:

   Individual Feature result

Feature	Accuracy for 2 books	Accuracy for 3 books	Feature Used
'	44.87	35.71	No
!	53	36.3	No
?	61.4	44.4	No
the	70.1	54.6	Yes
,	74.3	50	Yes
"	76.5	57.5	Yes
-	77.4	60	No
_	80.4	53.4	Yes
.	82.83	57.7	Yes
;	84.63	59.79	Yes

   The table contains accuracy obtained via Decision Tree classification for every feature.
   We can see Accuracy of these features decreases as number of authors are increased from 2 to 3. Features are selected by their ability to predict the authors accurately. Most of the features consist of special characters, and one feature used is the word 'the'. Few features with words such as 'at', 'a', 'is', etc were also checked, however the prediction was not up to the mark.

3. Decision Tree classifier:
   All the files ACD.txt, HM.txt and JA.txt are scanned at the start of the code. While Scanning these files features are extracted to reduce the time taken. These features can be used for both Decision tree as well as Logistic classifiers.
   For decision tree data is split into 2 parts training and testing. In the training process training data is used to calculate the gain of every question. Entropy function is used to calculate initial and subsequent entropies. Questions related to highest gain are used in the model. Gain and related questions are stored together. The code achieved an accuracy of 92% for 2 authors and 80% for 3 authors using features ('the',",_.;;).
   Class Question stores the questions asked to calculate the gain. Decision_Node stores the node values and reference to left and right trees. Find_best_splits is used to find the best gain. Question, Node, Gain Entropy define the decision tree model.

This model is used to predict the values in testing data. All the predicted values are compared with ground truths and accuracy is printed.

4. Logistic classifier
   Data is split into X_train, X_test, y_train, y_test. Values of y_train and y_test are changed to 0 and 1. Gradient descent is used to calculate the weights of the data using x_train and y_train with rate = 10**-6 and iterations can be chosen by user.

   Accuracy for 2 authors
   Accuracy of system for 100 iteration = 45%
   Accuracy of system for 1000 iteration = 87%
   Accuracy of system for 2000 iteration = 86.7
   Accuracy of system for 50000 iteration = 90.0%
   Accuracy of system for 100000 iteration = 86.14%

   Accuracy for 3 authors
   Accuracy of system for 100 iteration = 56.1%
   Accuracy of system for 1000 iteration = 41.63%
   Accuracy of system for 2000 iteration = 42%

   As the training period can be very high the system also informs the % of training completed. When taking the value of sigmoid function as 0.5, the code predicted the data accurately 54% of time, changing this value to 0.65 showed quite an improvement to 87% and increasing the value further to 0.7 reduced the accuracy to 70%.

5. Results
   Decision Tree:
      2 Author: 92%
      3 Author: 80%
   Logistic Classifier
      2 Author: 86%
      3 Author: 56.1%