Foundation of Intelligent System

Lab 2

Wikipedia Language Classification

Problem Understanding:

Investigated the use of decision trees and boosted decision stumps to classify text as one of three languages. Specifically, the task comprises of collecting data and train (in several different ways) some decision stumps/trees so that when given a 15 word segment of text from either the English or Dutch Wikipedia, the code will state to the best of its ability which language the text is in.

Code Run Process:

1. My submission consists of dataPreProcessor.py, train.py, predict.py, adaPredict.py, data.csv, English.txt, Dutch.txt and Italian.txt
2. The dataPreProcessor.py and train.py are the most important file as dataPreProcessor.py consists of code which deals with data collection, feature extraction and CSV generation and train.py consists of code which builds and trains the decision tree model using decision tree algorithm used in the same file. train.py also performs AdaBoost algorithm.
3. data.csv is the training data file generated by dataPreProcessor.py and it is given as input to the train.py and the name of this file is being hard-coded in the train.py
4. Once dataPreProcessor.py is successfully executed, next step is to run train.py.
5. train.py will generate predict.py which is decision tree predictor file and adaPredict.py which is decision stump predictor file.
6. There are various user inputs asked during the execution of all the files.
7. Following are the images for step by step execution of this program.

```
C:\Users\avs23\AppData\Local\Programs\Python\Python35-32\python.exe C:/Users/avs23/Desktop/FIS_Lab_2/dataPreProcessor.py
Merge Training Data file 1: Yes 2: No2
Enter the filename to read:English.txt
Select the language for given filename    1: English 2: Dutch 3: Italian 1
Reading TextFile English.txt
Data Preprocessing
Feature Selection
Generating Training Data file

Process finished with exit code 0
```

8. **Run dataPreProcessor.py for 4 times**. The **first 3 times to take all the language file input** and **4th time to merge the individual language input** into single training data.

```
C:\Users\avs23\AppData\Local\Programs\Python\Python35-32\python.exe C:/Users/avs23/Desktop/FIS_Lab_2/dataPreProcessor.py
Merge Training Data file 1: Yes 2: No1
Training Data merged and CSV created

Process finished with exit code 0
```

9. Running train.py

```
C:\Users\avs23\AppData\Local\Programs\Python\Python35-32\python.exe C:/Users/avs23/Desktop/FIS_Lab_2/train.py
Reading Training Data and generating predictor program
Do you want to check the accuracy 1: Yes  2: No1
Enter English language text document filenameEnglish.txt
Enter Dutch language text document filenameDutch.txt
Enter Italian language text document filenameItalian.txt
Checking Accuracy
Accuracy for English : 99.33736
Accuracy for Dutch : 99.09054
Accuracy for Italian : 98.34311
Do you want to run AdaBoost   1: Yes    2:No 1
Starting AdaBoost
Enter English language text document filenameEnglish.txt
Enter Dutch language text document filenameDutch.txt
Accuracy for English : 99.60439
Accuracy for Dutch : 97.08046

Process finished with exit code 0
```

10. Generated predict.py and adaPredict.py

English:

```
C:\Users\avs23\AppData\Local\Programs\Python\Python35-32\python.exe C:/Users/avs23/Desktop/FIS_Lab_2/predict.py
Enter the sentence you want to predict to school and to maintain a local store on the public square where the latest
ENGLISH

Process finished with exit code 0
```

Dutch:

```
C:\Users\avs23\AppData\Local\Programs\Python\Python35-32\python.exe C:/Users/avs23/Desktop/FIS_Lab_2/predict.py
Enter the sentence you want to predict de getouwen der wevers en de banken der spinners met hunne ontelbare spillen en bobijnen
DUTCH

Process finished with exit code 0
```

Italian:

```
C:\Users\avs23\AppData\Local\Programs\Python\Python35-32\python.exe C:/Users/avs23/Desktop/FIS_Lab_2/predict.py
Enter the sentence you want to predict punto luminoso l'artista non è felice perchè spesso manca all'uomo glorioso quasi tutto perchè la
ITALIAN

Process finished with exit code 0
```

Feature Description:

I have used 13 features and 1 target variable. The 13 features are as follows:

Feature 1: The most common and frequently appearing Dutch words such as **"de", "het"**, **"dat"**, **"en"**, **"een"**, **"voor"**, **"van"**, **"welke"**, **"te"**, **"hij"**, **"zij"**, **"op"**, **"ik"** and **"bij".**

Feature 2: The most common and frequently appearing Italian words such as **"che", "il"**, **"ll"**, **"e"**, **"un"**, **"di"**, **"per"** and **"era".**

Feature 3: The most common and frequently appearing English words such as **"the", "but"**, **"for"**, **"which"**, **"that"**, **"and"** and **"not".**

Feature 4: Letters that do not occur in Italian language as they have 21 alphabets **"j", "k", "w", "x"** and **"y".**

Feature 5: Special Italian characters such as **ù, ò, ì, è and à.**

Feature 6: Consecutive vowels frequently used in Dutch language.

Feature 7: Most occurring substring **"ijk", "sch"** and **"ijn"** in Dutch.

Feature 8: Other common English words such as **"is", "was", "of"** and **"all".**

Feature 9: l' and **d'** most frequently used in Italian language.

Feature 10: "come" and "a" common across many languages.

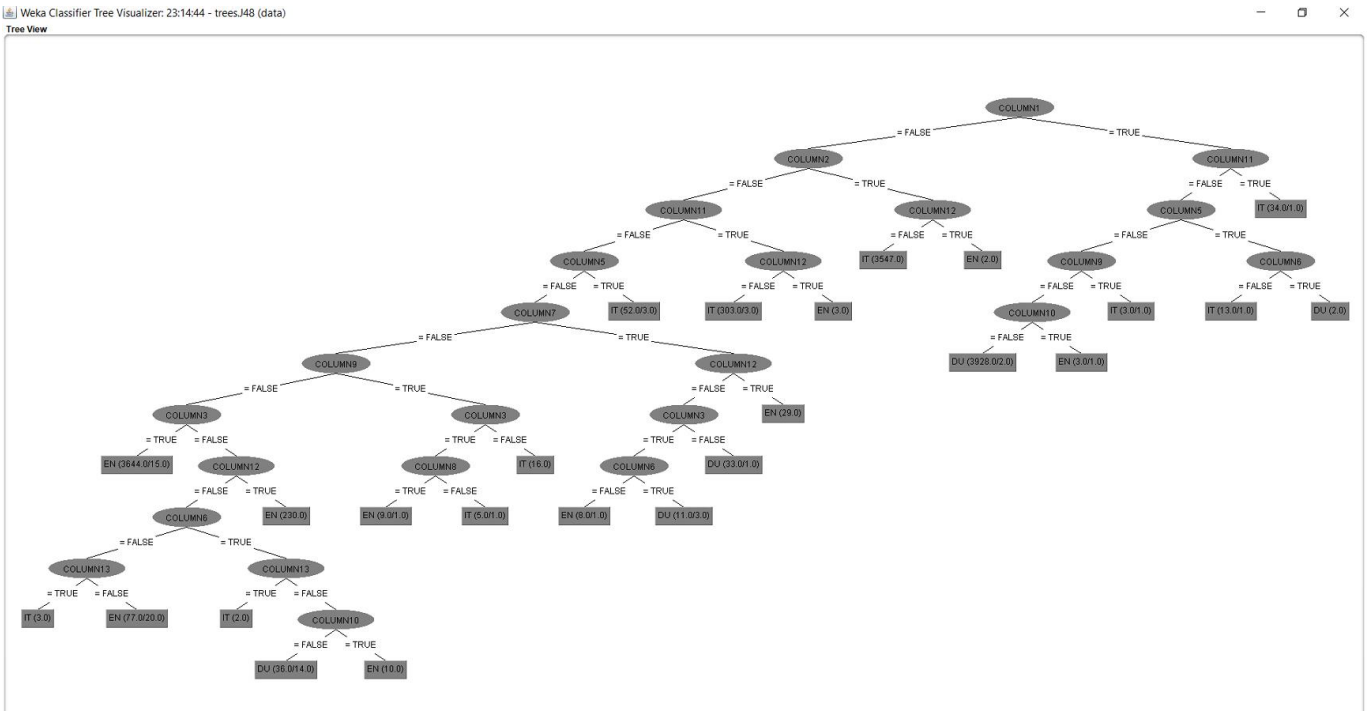
Feature 11: Other Italian common words such as **mi, si, le, ma, la** and **se.**

Feature 12: Common English words "he", "she", "it" and "they"

Feature 13: Count of most occurring letter **"i"** in Dutch and Italian.

Decision Tree:

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       11914               99.2585 %
Incorrectly Classified Instances        89                0.7415 %
Kappa statistic                          0.9889
Mean absolute error                      0.0073
Root mean squared error                  0.065
Relative absolute error                  1.6426 %
Root relative squared error             13.7793 %
Total Number of Instances            12003

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.991    0.005    0.989      0.991    0.990      0.985    0.998     0.993     EN
                 0.996    0.003    0.993      0.996    0.995      0.992    0.998     0.996     DU
                 0.991    0.002    0.995      0.991    0.993      0.989    0.998     0.996     IT
Weighted Avg.    0.993    0.004    0.993      0.993    0.993      0.989    0.998     0.995

=== Confusion Matrix ===

    a    b    c   <-- classified as
 3965   24   12 |    a = EN
    9 3986    6 |    b = DU
   34    4 3963 |    c = IT
```



The decision tree algorithm which I implemented also implements zeroR algorithm to predict the best attribute along with impurity calculator gini index. The Gini Index is calculated by subtracting the sum of the squared probabilities of each class from one. It favors larger partitions.

Also, I have calculated the accuracy to predict the correct language based on the model I generated here. Below is the output of my decision tree model.

```
C:\Users\avs23\AppData\Local\Programs\Python\Python35-32\python.exe C:/Users/avs23/Desktop/FIS_Lab_2/train.py
Reading Training Data and generating predictor program
Do you want to check the accuracy 1: Yes  2: No1
Enter English language text document filenameEnglish.txt
Enter Dutch language text document filenameDutch.txt
Enter Italian language text document filenameItalian.txt
Checking Accuracy
Accuracy for English : 99.33736
Accuracy for Dutch : 99.09054
Accuracy for Italian : 98.34311
Do you want to run AdaBoost   1: Yes    2:No2
Exit

Process finished with exit code 0
```

Accuracy for English is 99.33%, for Dutch it is 99.09% and for Italian it is 98.34% which is quite close to the model which Weka tool generated (99.25%).

Boosting Description:

The AdaBoosting algorithm which I have implemented is quite similar to the one given in the textbook. The learning algorithm used by the adaboost algorithm is oneR which returns a single rule result and is compared with the label or target variable of every instance if it does not match we further calculate the error rate. We then calculate the hypothesis weight for every feature and return the result. If it is greater than 0 it classifies as “Dutch” or else “English”.

```
C:\Users\avs23\AppData\Local\Programs\Python\Python35-32\python.exe C:/Users/avs23/Desktop/FIS_Lab_2/train.py
Reading Training Data and generating predictor program
Do you want to check the accuracy 1: Yes  2: No2
Decision Tree Done
Do you want to run AdaBoost   1: Yes    2:No1
Starting AdaBoost
Enter English language text document filenameEnglish.txt
Enter Dutch language text document filenameDutch.txt
Accuracy for English : 99.60439
Accuracy for Dutch : 97.08046

Process finished with exit code 0
```

AdaBoost Accuracy for English is 99.60% and for Dutch it is 97.08%.

Output of adaPredict.py:

```
C:\Users\avs23\AppData\Local\Programs\Python\Python35-32\python.exe C:/Users/avs23/Desktop/FIS_Lab_2/adaPredict.py
Enter the sentence you want to predict managed to buy a house and lot to furnish it comfortably to send his children
ENGLISH

Process finished with exit code 0
```

```
C:\Users\avs23\AppData\Local\Programs\Python\Python35-32\python.exe C:/Users/avs23/Desktop/FIS_Lab_2/adaPredict.py
Enter the sentence you want to predict er gaan het zal er een leventje zijn dat gij er in uwen ouden dag
DUTCH

Process finished with exit code 0
```