CSE 537 - Artificial Intelligence

Project - 5

Pavan Kumar Peela - 112045988
Saquib Ali Khan - 112129882

Dated: 8th December 2018

A. Clickstream Mining with Decision Trees:

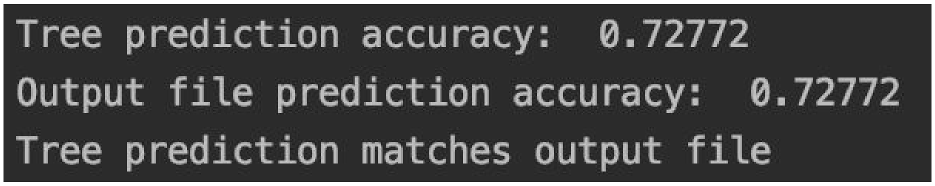
We are given a task posed in KDD Cup 2000 which involves mining click-stream data collected from Gazelle.com and we need to determine whether a visitor view another page on the site or will he leave given a set of page views.

The following are the observations tried and observed:

1. Experiment 1:

   P-value = 0.01

   Output:

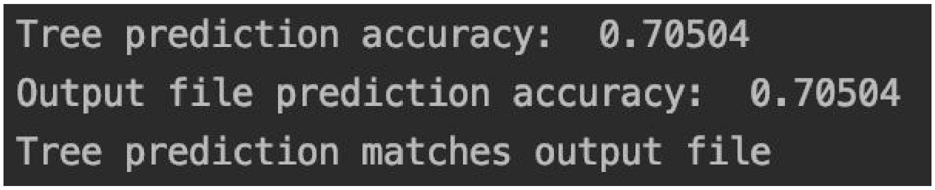
   ```
   Tree prediction accuracy:  0.72772
   Output file prediction accuracy:  0.72772
   Tree prediction matches output file
   ```

   Number of Internal Nodes: 108
   Number of leaf Nodes: 433
   Time took: 225.22s

2. Experiment 2:

   P-value = 0.05

   Output:

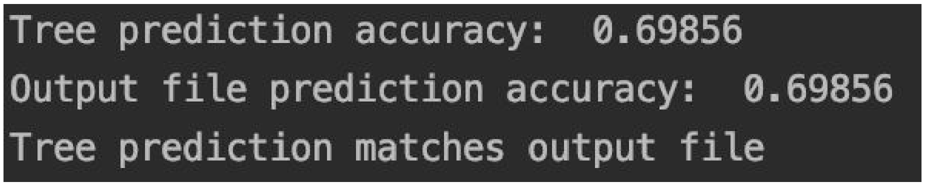
   ```
   Tree prediction accuracy:  0.70504
   Output file prediction accuracy:  0.70504
   Tree prediction matches output file
   ```

   Number of Internal Nodes: 638
   Number of leaf Nodes: 2553
   Time taken: 564.39s

3. Experiment 3:

   P-value = 0.08

   Output:

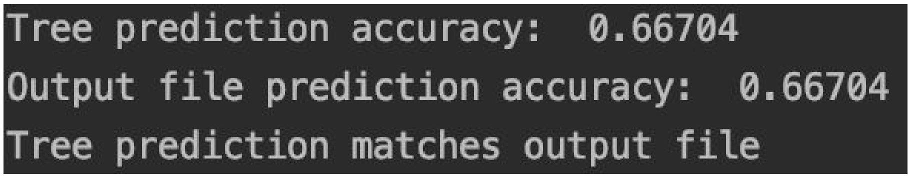
   

   ```
   Tree prediction accuracy:  0.69856
   Output file prediction accuracy:  0.69856
   Tree prediction matches output file
   ```

   Number of Internal Nodes: 813.23s

Number of leaf Nodes: 3253
Time taken: 825.18

4. Experiment 4:
P-value = 0.15
Output:



```
Tree prediction accuracy:  0.66704
Output file prediction accuracy:  0.66704
Tree prediction matches output file
```

Number of Internal Nodes: 3482
Number of leaf Nodes: 13929
Time taken: 1444.00s

Observations:

It can be observed that as the magnitude of ‘p-value’ increases, the number of nodes expanded also increase, as we cut short the pruning as the value increases and eventually traversing full tree when p_value is 1.

When the p_value is 1, we expand the whole tree and as a result, there is no bias in the learning. This may result in the error in training to be zero, but most of the time this results in over-fitting. It takes a lot more time than the rest.

When p_value is 0.01, on the lower side, we expand very minimal part of the tree as excessive pruning of the tree takes place.

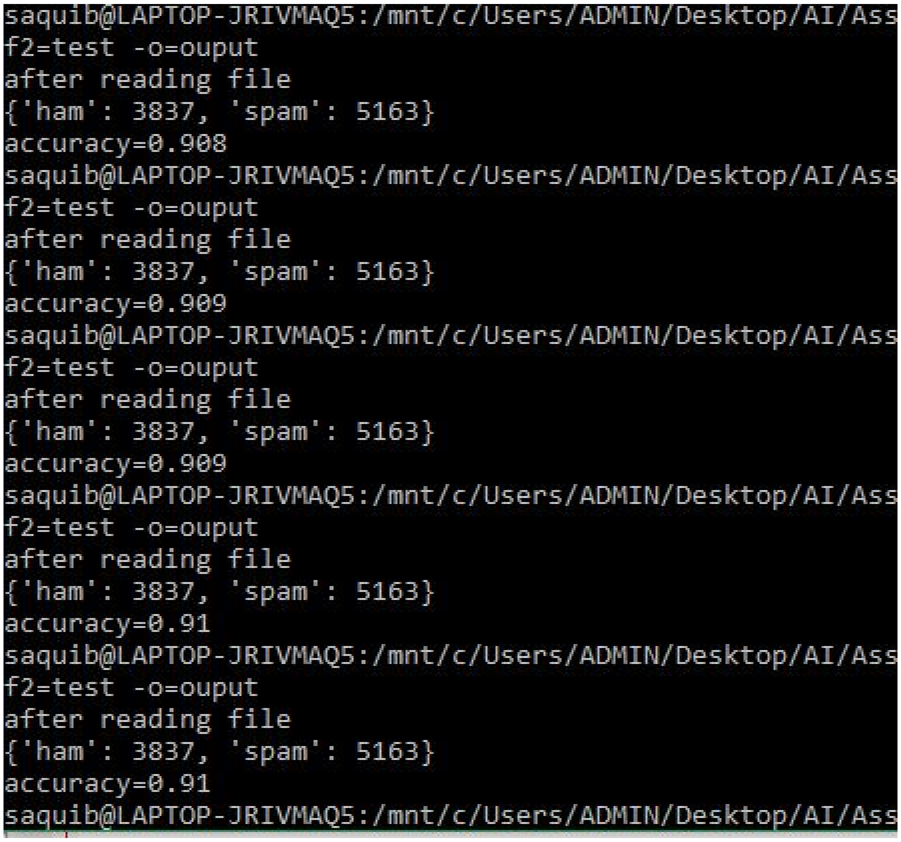
We achieve high accuracy for p_value of 0.01, when we tested with p_value 0.08, the accuracy is minutely reduced to 0.6986 and reduces as we keep on increasing the p_value

B. Naive Bayes Classifier:

The task is to implement a Naive Bayes Classifier for the purpose of text classification.
Also to try for various values of Smoothing Parameters.

In particular I have used Laplace smoothing in my implementation.
The model gives a decent amount of accuracy when Laplace smoothing of 1 is given to the naive Bayes implementation. The accuracy comes out to be 90.08 on the test set.

Following is the screenshot of results with accuracy on changing the values of the smoothing parameter. The accuracy shows a slight increase in the test set on increasing the smoothing value.



```
saquib@LAPTOP-JRIVMAQ5:/mnt/c/Users/ADMIN/Desktop/AI/Ass
f2=test -o=ouput
after reading file
{'ham': 3837, 'spam': 5163}
accuracy=0.908
saquib@LAPTOP-JRIVMAQ5:/mnt/c/Users/ADMIN/Desktop/AI/Ass
f2=test -o=ouput
after reading file
{'ham': 3837, 'spam': 5163}
accuracy=0.909
saquib@LAPTOP-JRIVMAQ5:/mnt/c/Users/ADMIN/Desktop/AI/Ass
f2=test -o=ouput
after reading file
{'ham': 3837, 'spam': 5163}
accuracy=0.909
saquib@LAPTOP-JRIVMAQ5:/mnt/c/Users/ADMIN/Desktop/AI/Ass
f2=test -o=ouput
after reading file
{'ham': 3837, 'spam': 5163}
accuracy=0.91
saquib@LAPTOP-JRIVMAQ5:/mnt/c/Users/ADMIN/Desktop/AI/Ass
f2=test -o=ouput
after reading file
{'ham': 3837, 'spam': 5163}
accuracy=0.91
saquib@LAPTOP-JRIVMAQ5:/mnt/c/Users/ADMIN/Desktop/AI/Ass
```

Discussion on implementation:

Implementation is quite straightforward.
The arguments have been parsed to take input files.
Each line of training file contains a sequence of space-separated words and counts for each mail. These have been easily parsed from file to maintain a dictionary called class_word_count = {}, which basically maintains the word count for each word corresponding to each type of mail.

This dictionary together with other variables like number of unique words in the training set, number of hams, number of spams, etc have been used to calculate conditionals corresponding to each class type and word, which are then directly used on test mails to calculate the ham or spam probability using words of that test mail.

Alpha is the value of smoothing parameter which is being passed to classifier function to be used inside.