Machine Learning Assignment 2
Decision Trees
Submission deadline: March 15, 2019

In this assignment, you will be required to implement the Decision Tree algorithm from scratch using both (a) Information Gain and (b) Gini Index, to decide on the splitting attribute.
In Part 1, you will implement it on a small toy dataset to gain confidence in your implementation.
In Part 2, you will be given a larger real-world dataset. For both the parts, you will be asked to compare the accuracy of your implemented model with that of scikit learn.

Part 1

(Description) Consider that you are willing to buy a car and you have collected information having four attributes 'price', 'maintenance', 'capacity' and 'airbag', and are trying to predict whether a given car is 'profitable' or not. Assume all the four attributes are categorical, with discrete values.

(Dataset) Download the training and test data here. The sheet labelled "training data" contains the data to be trained on. The sheet named as "test data" contains the data on which you have to test your model.

(Tasks)
(A) Train your decision tree classifier on the train-data (where you will use "profitable"), using the impurity measure:

a. Information Gain
b. Gini Index

Test your model on test-data (where the "profitable" label is unseen). After prediction, report the individual accuracies on the test data obtained using (a) and (b). Note that the "profitable" field should not be used in the classification process.

For both cases, write your program such that it prints out the decision tree, in a particular format. For example, assume that your decision tree looks like the following - the attribute "price" is the root node and "maintenance" is the 2nd level node and "capacity" is the third level node (under maintenance = low). "yes" and "no" specifies the final value of "profitable". Then the program should print out the decision tree as follows:

```
price = low
| maintenance = low
        | capacity = 4 : yes
| maintenance = high : no
```

Where subsequent levels are at increasing indentations from the left.

(B) Repeat the experiment using the decision tree algorithm implemented in scikit learn, using both Information Gain and Gini index. Report the accuracies on test data.

(Deliverables) Your report should contain :

1. The decision tree in the format provided in (A)
2. The value of Information Gain and Gini Index of the root node using :
    a. Your model
    b. scikit learn
3. The labels generated on the test data and accuracy on the test data using :
    a. Your model
    b. scikit learn

Part 2

(Description) In this part, you will implement the decision tree algorithm to learn a classifier that can assign a topic (science, sports, atheism etc.) to any news article.

(Dataset)
Train and test your algorithms with a subset of the 20 newsgroup dataset. More precisely, you will use the documents on alt.atheism and comp.graphics newsgroup. To simplify your implementation, these articles have been pre-processed and converted to the bag of words model. Each article is converted to a vector of binary values such that each entry indicates whether the document contains a specific word or not.

Download the training set (traindata.txt) and test set (testdata.txt) of articles with their correct newsgroup label (trainlabel.txt, testlabel.txt) here.

Each line of the files traindata.txt and testdata.txt are formatted “docId wordId” which indicates that word *wordId* is present in document *docId*. The files trainlabel.txt and testlabel.txt indicate the category (1=alt.atheism or 2=comp.graphics) for each document (docId = line number). The file *words.txt* indicates which word corresponds to each wordId (denoted by the line number).

(Tasks)
(A) Implement the decision tree learning algorithm. Here, each decision node corresponds to a word feature, which is selected by maximizing the information gain.

Design your algorithm to take as input a maximum depth. If a branch of the decision tree reaches this specified depth, it should not be grown further.

Experiment with your algorithm by building trees with increasing maximum depth until a full tree is obtained.
Report the training and testing accuracy (i.e., percentage of correctly classified articles) of each tree by producing a graph with two curves - one curve for training accuracy and one curve for testing accuracy - as a function of the maximum depth of the tree.
Report also the tree that achieved the highest testing accuracy.

(B) Use the decision tree algorithm implemented in scikit learn, using Information Gain, to classify the same data. Report the accuracies on test data.

(Deliverables) Your report should contain :

1. A graph showing the training and testing accuracy as the maximum depth increases.
2. Does overfitting occur? If yes, after what maximum depth does overfitting occur?
3. A brief discussion of the word features selected by the decision tree that achieved the highest testing accuracy. In your opinion, did all the word features selected make sense?

Submission Instructions

1. Submit separate codes for Part 1 and Part 2
2. Submit a README file which will contain the instructions on how to execute your codes
3. Submit separate report files for Part 1 and Part 2

All source codes, result files and the final report must be uploaded via the course Moodle page, as a **single compressed file (.tar.gz or .zip)**.
The compressed file should be named as: **{ROLL_NUMBER}_ML_A2.zip or {ROLL_NUMBER}_ML_A2.tar.gz** Example: If your roll number is 16CS60R00, then your submission file should be named as 16CS60R00_ML_A2.tar.gz or 16CS60R00_ML_A2.zip

You can use C / C++ / Java / Python for writing the codes; no other programming language is allowed. You cannot use any library/module meant for Machine Learning for implementing your models (except where mentioned). You can use libraries for other purposes, such as generation and formatting of data. Also you **should not use any code available on the Web**. **Submissions found to be plagiarised or having used ML libraries will be awarded zero marks for all the students concerned**.

***Note that the evaluators can deduct marks if the deliverables are not found in the way that has been asked for the assignment.

Submission deadline: March 15, 2019, 23:59 IST [hard deadline]

For any questions about the assignment, contact the following TAs:
1. Abhisek Dash (assignmentad @ gmail . com)
2. Paheli Bhattacharya (paheli.cse.iitkgp @ gmail . com)