

## CS561/571 - ARTIFICIAL INTELLIGENCE LAB

### ASSIGNMENT-9: Decision Tree

**(Read all the instructions carefully & adhere to them.)**

**Date: 15th November 2022**

**Deadline: 29th November, 2022**

**Total Credit: 30**

#### Instructions:

1. The assignment should be completed and uploaded by 29th Nov, 2022, 11:59 PM IST.
2. Markings will be based on the correctness and soundness of the outputs. Marks will be deducted in case of plagiarism.
3. Proper indentation and appropriate comments are mandatory.
4. Make proper documentation of all results and observations with their analysis.
5. You should zip all the required files and name the zip file as roll\_no\_of\_all\_group\_members .zip, e.g., 1501cs11\_1201cs03\_1621cs05.zip.
6. Upload your assignment (the zip file) in the following link:  
<https://www.dropbox.com/request/64IjDH5dtI1dd3AvXvTq>
7. For any queries regarding this assignment, you can contact:  
Deeksha Varshney ( [deeksha.varshney2695@gmail.com](mailto:deeksha.varshney2695@gmail.com) ) or  
Gitanjali Kumari ( [gitanjali.singh228@gmail.com](mailto:gitanjali.singh228@gmail.com) ) or  
Samrat Mukherjee ( [samratpisv123@gmail.com](mailto:samratpisv123@gmail.com) )

#### Questions

Write a Python program that implements Question classification using Decision Tree classifier.

#### Example

**Question:** What is the temperature at the center of the earth ? **Class:** NUM, which refers to the question that looks for the numeric type answer.

#### Dataset

**Training Set:** [http://cogcomp.org/Data/QA/QC/train\\_5500.label](http://cogcomp.org/Data/QA/QC/train_5500.label)

**Test Set:** [http://cogcomp.org/Data/QA/QC/TREC\\_10.label](http://cogcomp.org/Data/QA/QC/TREC_10.label)

Use only the coarse grained class label to build your model. For more details about the dataset follow these paper: <https://goo.gl/jAJFKQ>

## Features

- (a) Length of the question
- (b) Lexical Features: Word n-gram.
- (c) Syntactic Features: Parts of speech tag unigrams.

Implement n-gram (  $n=1,2$  and 3) features for each question instance. You may choose only the most frequent n-grams to provide as a feature for your model.

For  $n=1$ , use 500 most frequent 1-gram, similarly use 300 and 200 most frequent n-grams, for  $n=2$  and 3 respectively. For parts of speech tag unigrams, first you need to get a POS tag for each question instance. Use can use any library like Stanford POS tag-ger see <https://nlp.stanford.edu/software/tagger.shtml> NLTK POS tagger see <http://www.nltk.org/book/ch05.html> etc. Similar to lexical features use 500 most frequent 1-gram to build the model. For more details about the features, see section 2.3 in the following paper <https://goo.gl/X7X7ox>

## Result and Evaluation

- Report the 10-fold cross-validation results in terms of precision, recall, and F-score.
- Report results of feature ablation study and state which feature has contributed most towards correctly predicting a particular class
- Report precision, recall, and F-score measures on test sets using models based on the gini index, mis-classification error and cross-entropy.
- Show whether errors propagated by one model are corrected by other models or not. If yes, then report how many percent of samples are corrected.

Ex. Observe how many samples are mis-classified using gini index based model but correctly classified by mis-classification error and cross-entropy based model