CS771 Assignment 2

TANEYA SONI, SONAM, ANIKET KUMAR, VEDIT KUMAWAT, ABHISHEK KUMAR PATHAK

TOTAL POINTS

36 / 40

QUESTION 1

1 Algorithm description 10 / 10

✓ **+ 10 pts** *A valid description of the algorithm and hyperparameters involved.*

- 2 pts Minor mistakes, insufficient details or unclear description

+ 0 pts Completely wrong or else unanswered

QUESTION 2

2 Code evaluation 26 / 30

+ 0 pts Correct

+ 26 *Point adjustment*

GROUP NO: 53

Grading scheme for code:
Training time tt (in sec): tt < 2 (10 marks), 2 <= tt < 4 (9 marks), 4 <= tt < 6 (8 marks), 6 <= tt < 8 (7 marks), 8 <= tt < 10 (6 marks), tt >= 10 (5 marks)
Model size ms (in MB): ms < 1024 (5 marks), 1024 <= ms < 2048 (4 marks), ms >= 2048 (3 marks)
Queries per round qr: qr < 3.65 (15 marks), 3.65 <= qr < 3.70 (14 marks), 3.70 <= qr < 3.75 (13 marks), 3.75 <= qr < 4.00 (11 marks), 4.00 <= qr < 5.00 (9 marks), qr >= 5.00 (7 marks)
The above 3 marks are each multiplied by the model accuracy (ac), which is a number in the interval [0,1], to obtain final scores.

tt = 7.231 sec : 7 marks
ms = 1532.3 KB : 4 marks
qr = 3.645 : 15 marks
TOTAL: 26 marks
Model accuracy ac: 1.0
ADJUSTED TOTAL (TOTAL * ac): 26.0 marks

COMMENT: The code gave the following error on execution. The code was minimally modified to get it running. AttributeError: 'Node' object has no attribute 'entropy'

Assignment-2

Sonam
221110057
sonamk22@iitk.ac.in

Aniket Kumar
190134
aniketkr@iitk.ac.in

Vedit Kumawat
190957
vedit@iitk.ac.in

Taneya Soni
221110062
taneyas22@iitk.ac.in

Abhishek Kumar Pathak
22111002
akpathak@iitk.ac.in

1 Question

Give detailed calculations explaining the various design decisions you took to develop your decision tree algorithm. This includes the criterion to choose the splitting criterion at each internal node (which essentially decides the query word that Melbo asks when that node is reached), criterion to decide when to stop expanding the decision tree and make the node a leaf, any pruning strategies and hyperparameters etc.

We tried with the three commonly used splitting criteria entropy, Gini index, and information ratio.

We did not try pruning because it will only be helpful in decreasing the test time and model size, but will result in increase in training time and decrease in win ratio. Because we are targeting for 100% win ratio also test time is not a criteria for grading, so it will not be very helpful for our requirements.

Entropy: Entropy is a measure of the impurity of a set of data. It is calculated as the sum of the negative logarithms of the probability of each class label in the dataset. The entropy value ranges between 0 and 1. A value closer to 1 means the data is more impure, and a value closer to 0 means the data is pure.

$$Entropy(D) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

where,

D is the dataset

k is the number of classes in D

pi is the proportion of samples in D that belong to class i

Gini index: Gini index is another measure of impurity of a set of data. It is calculated as the sum of the squared probabilities of each class label subtracted from 1. The Gini index value ranges between 0 and 1. A value closer to 0 means the data is pure, and a value closer to 1 means the data is more impure.

$$Gini(D) = 1 - \sum_{i=1}^{k} (p_i)^2$$

where,

D is the dataset

k is the number of classes in D

pi is the proportion of samples in D that belong to class i

Information gain ratio: Information gain ratio is a measure of the quality of a split. The Information Gain Ratio is calculated by dividing the Information Gain by the Split Information. The Split Information is the entropy of the attribute's values and indicates the amount of randomness in the attribute's distribution.

$$Information\ Gain\ Ratio(T, a) = \frac{Information\ Gain}{Split\ Information}$$

$$IG(T, a) = H(T) - H(T|a)$$

where T is random variable H(T/a) is entropy of T given the value of attribute a.

$$SplitInformation(T) = -\sum_{i=1}^{n} \left(\frac{\mathrm{N}(t_i)}{\mathrm{N}(t)} \cdot \log_2 \frac{\mathrm{N}(t_i)}{\mathrm{N}(t)} \right)$$

where X is random Variable with possible values x1,x2,,,xi.
$N(t_i)$ being the number of times that t_i occurs divided by the total count of events N(t) where t is set of events.

We have used top-down greedy approach with entropy(min) based splitting criteria for this assignment.

The table below shows measurements of Training time, win fraction, average queries taken by model to guess the word in each round and model size.

	Training time(s)	win fraction	average steps	model size
Entropy	2.57	1.0	3.99	1079215.0
Gini-index	2.69	1.0	4.11	1074494
Information ratio	2.61	1.0	3.99	1079357

We saw that Entropy gave the best results so we used entropy as the splitting criteria.
The top-down greedy approach is a commonly used method for constructing decision trees in machine learning. In this approach, the tree is built starting from the root node and recursively splitting the data based on the attribute that provides the most information gain.

Here are the steps for building a decision tree using the top-down greedy approach:

Step 1: We start with the entire dataset at the root node of the tree .

step 2: We calculate the entropy (or another impurity measure, such as Gini index or information gain ratio) for each word present in that node.

Step 3: We choose the attribute with the least entropy and use it to split the node words into two or more subsets. Each subset will correspond to a branch of the decision tree.

Step 4: We repeat steps 2 and 3 recursively for each subset until a stopping criterion is met. The Stopping criterion is single sample per leaf node .

Step 5: The word present at the leaf node will be used as label for classification.

The decision tree is now ready to use for prediction. The top-down greedy approach can quickly build decision trees and is easy to understand and interpret.

1 Algorithm description **10 / 10**

✓ **+ 10 pts** *A valid description of the algorithm and hyperparameters involved.*

- 2 pts Minor mistakes, insufficient details or unclear description

+ 0 pts Completely wrong or else unanswered

2 Code evaluation 26 / 30

+ 0 pts Correct

+ 26 *Point adjustment*

GROUP NO: 53

Grading scheme for code:
Training time tt (in sec): tt < 2 (10 marks), 2 <= tt < 4 (9 marks), 4 <= tt < 6 (8 marks), 6 <= tt < 8 (7 marks), 8 <= tt < 10 (6 marks), tt >= 10 (5 marks)
Model size ms (in MB): ms < 1024 (5 marks), 1024 <= ms < 2048 (4 marks), ms >= 2048 (3 marks)
Queries per round qr: qr < 3.65 (15 marks), 3.65 <= qr < 3.70 (14 marks), 3.70 <= qr < 3.75 (13 marks), 3.75 <= qr < 4.00 (11 marks), 4.00 <= qr < 5.00 (9 marks), qr >= 5.00 (7 marks)
The above 3 marks are each multiplied by the model accuracy (ac), which is a number in the interval [0,1], to obtain final scores.

tt = 7.231 sec : 7 marks
ms = 1532.3 KB : 4 marks
qr = 3.645 : 15 marks
TOTAL: 26 marks
Model accuracy ac: 1.0
ADJUSTED TOTAL (TOTAL * ac): 26.0 marks

COMMENT: The code gave the following error on execution. The code was minimally modified to get it running. AttributeError: 'Node' object has no attribute 'entropy'