INTRODUCTION MACHINE LEARNING

EXERCISE 5

Bauhaus-Universität Weimar

TEACHER:
Johannes Kiesel

GROUP:
Group 16

SUBMITTED BY:

Aaron Perez Herrera
Cesar Fernando Gamba Tiusaba
Chun Ting Lin
Olubunmi Emmanuel Ogunleye

Exercise 1: Decision Trees (1+1+1+1+1+1=6 Points)

a. Name these concepts:
   (a1) xIA: This represents a partition of the feature space where condition A is met.
   (a2) T: This symbol denotes a decision tree.
   (a3) t: This symbol represents a node within the decision tree.
   (a4) X(t): This refers to the feature vector associated with a specific node t in the decision tree.
   (a5) D(t): This symbolizes the subset of the example set D that is represented by the node t.
   (a6) ΔI: This signifies the reduction in impurity resulting from a split.

b. Name this expression: X = {x E X : x| E B} U {x E X : xI B}
   This represents a partitioning of the dataset X into two subsets based on whether the attribute value xIA belongs to the subset B or not.

c. What are the three requirements of an impurity function?
   1. Non-negativity: The impurity function must be non-negative (ı(D)≥0\iota(D) \geq 0ı(D)≥0 for any dataset DDD).
   2. Maximum at uniform distribution: The impurity function reaches its maximum when all classes in DDD are uniformly distributed.
   3. Minimum at pure distribution: The impurity function is zero when all examples in DDD belong to the same class.

d. What is the hypothesis space of decision trees?
   The set of possible decision trees over D forms the hypothesis space H.

e. What is the search space of the ID3 algorithm?
   The search space of the ID3 algorithm consists of all possible trees that can be constructed by recursively partitioning the dataset using attributes based on information gain.

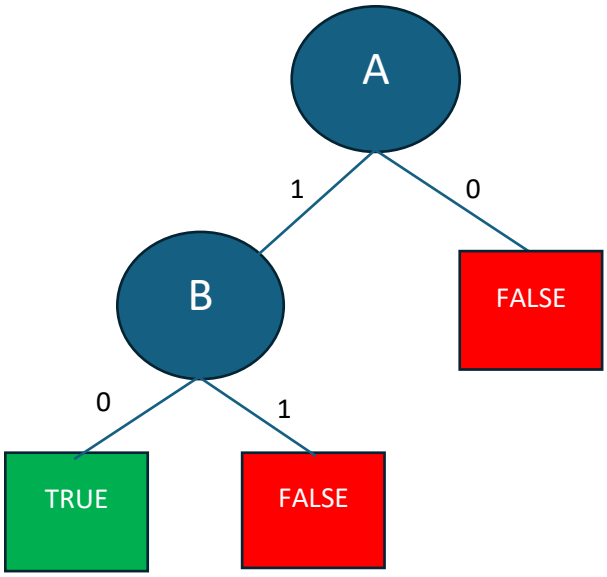
f. What is the difference between the inductive bias of the candidate elimination algorithm and that of the ID3 algorithm? Hint: search bias and restriction bias.
   - Candidate elimination algorithm: Combines search bias (consistent hypotheses with training data) and restriction bias (focus on version space).
   - ID3 algorithm: Uses search bias to prefer smaller trees with higher information gain, but its restriction bias assumes all target concepts can be represented as decision trees.

Exercise 2: Decision Trees (1+1+1+0=3 Points)

Construct by hand decision trees corresponding to each of the following Boolean formulas. The examples (x, c) E D consist of a feature vector x where each component corresponds to one of the Boolean variables (A, B, ...) used in the formula, and each example corresponds to one interpretation (i.e. assignment of 0/1 to the Boolean variables). The target concept c is the truth value of the formula given that interpretation. Assume the set D contains examples with all possible combinations of attribute values.

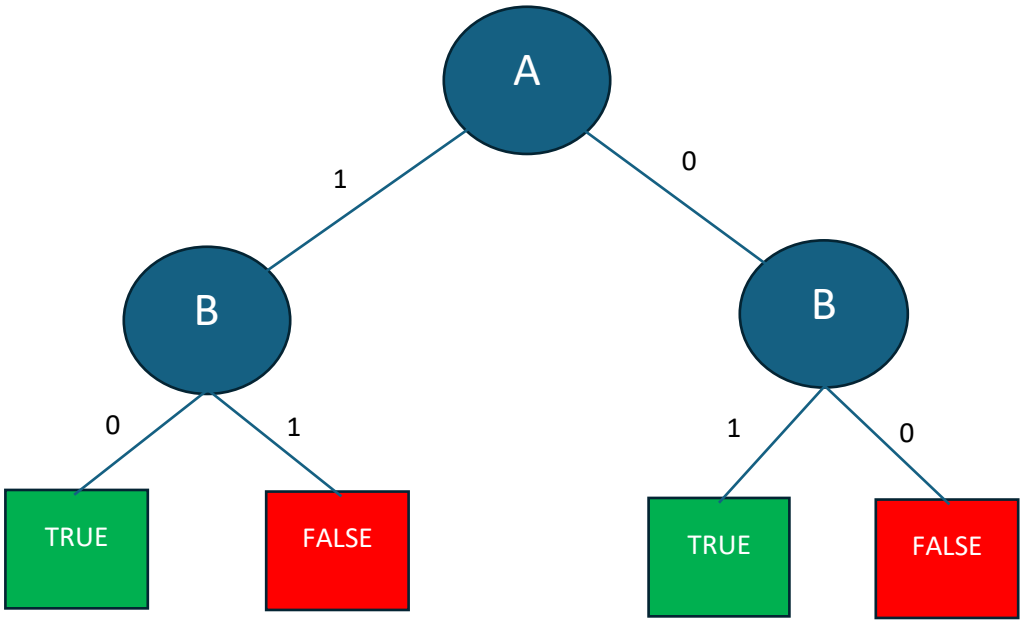
Hint: It may be helpful to write out the set D for each formula as a truth table.

(a) A ^ (NOT B)



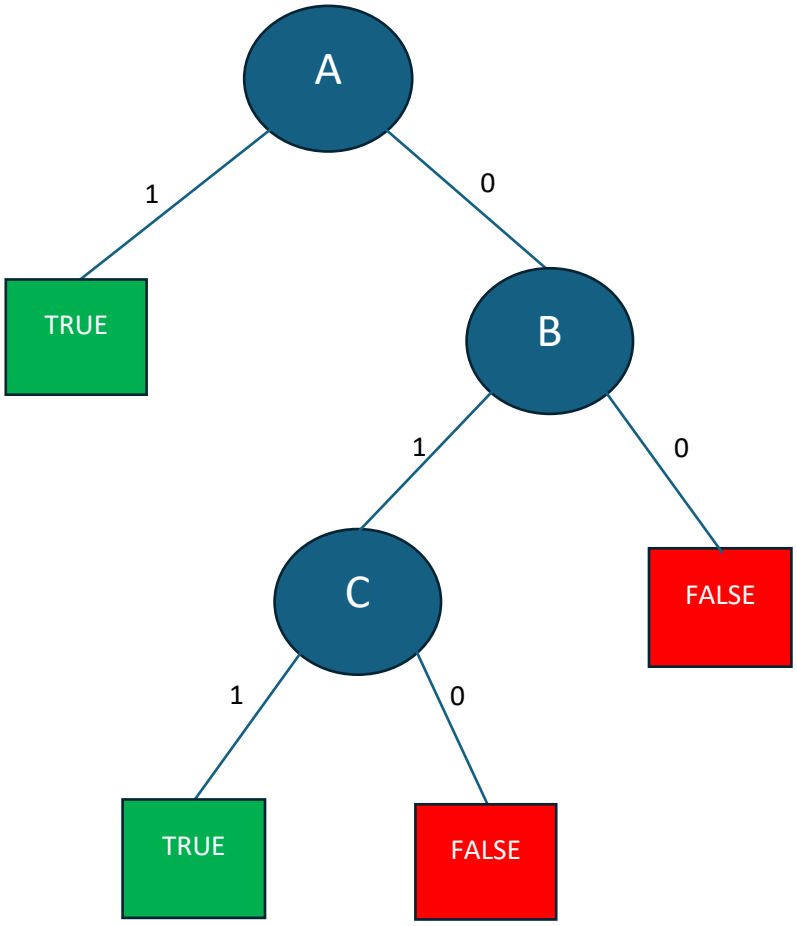
A	B	A and NOT(B)
0	0	0
0	1	0
1	0	1
1	1	0

(b) A XOR B



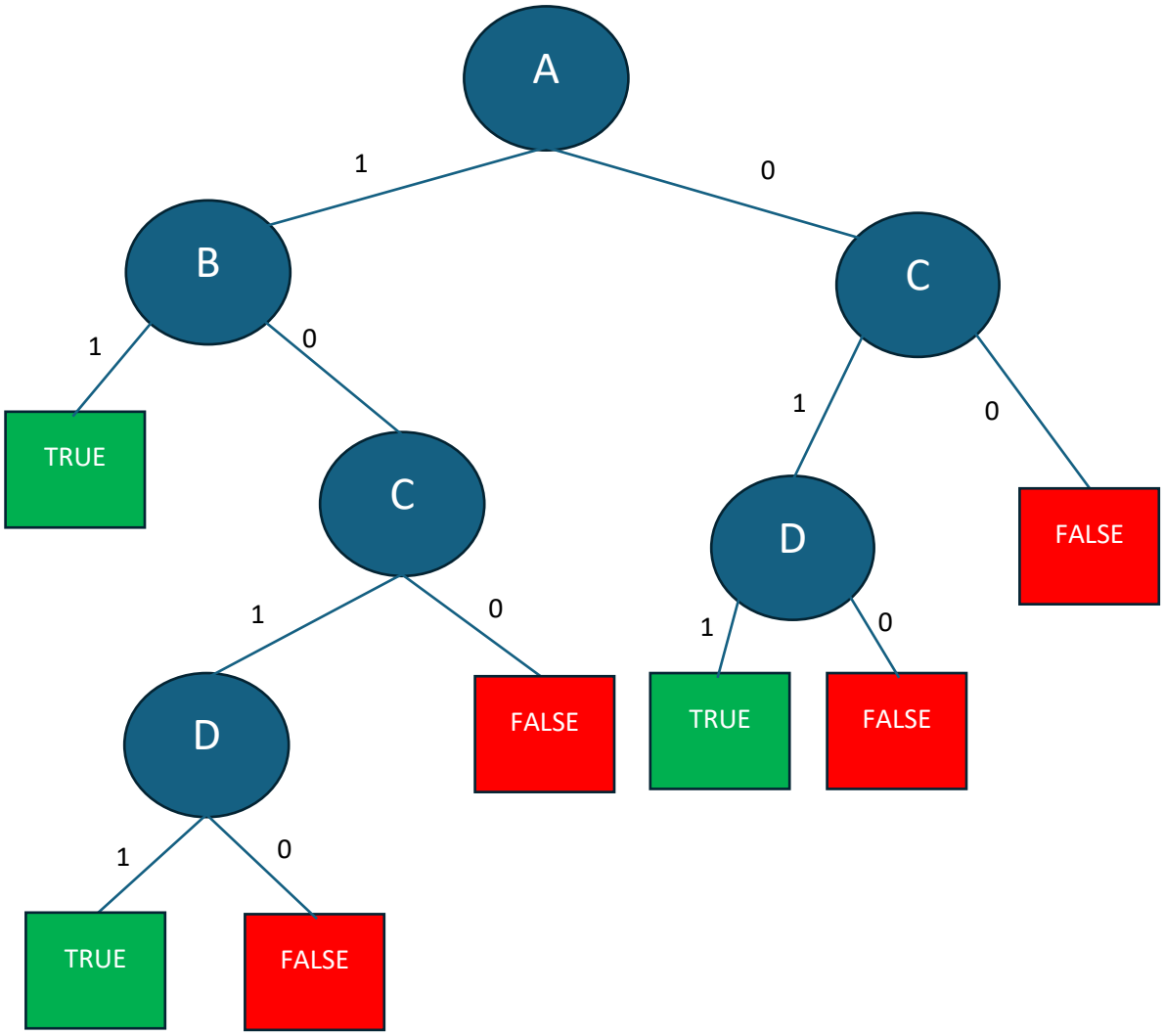
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

(c) A V (B ^ C)



A	B	C	A or (B and C)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

(d) (A ^ B) V (C ^ D)

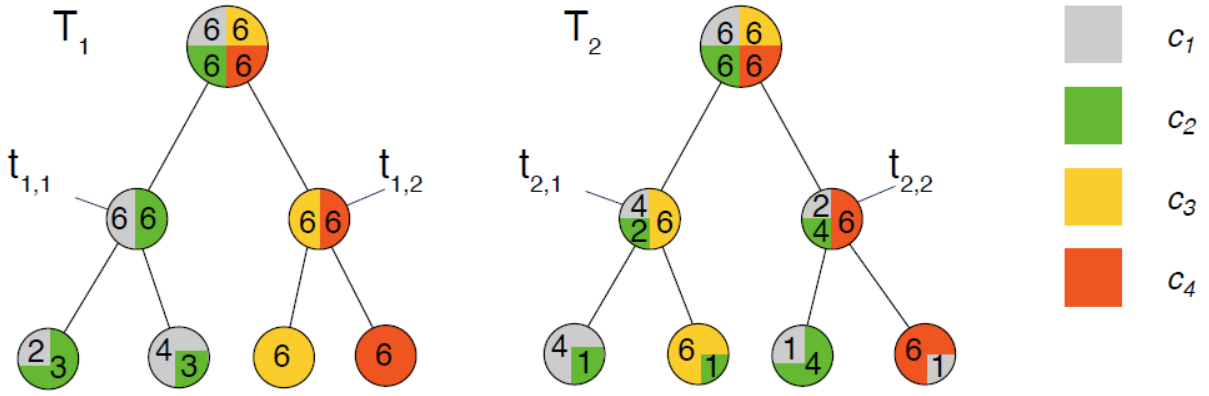


A	B	C	D	(A and B) or (C and D)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0

1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Exercise 3: Impurity Functions (3+0+0=3 Points)

Let D be a set of examples over a feature space X and a set of classes C = {c1, c2, c3, c4}, with |D| = 24. Consider the following illustration of two possible decision trees, T1 and T2 – the colors represent the classes present in each subset D(ti) represented by node ti,j of Ti; the numbers denote how many examples of each class are present.



a) First, consider only the first split that each of the two trees makes: compute $\Delta\iota(D, \{D(t1,1),D(t1,2)\})$ and $\Delta\iota(D, \{D(t2,1),D(t2,2)\})$ with (1) the misclassification rate $\iota_{misclass}$ and (2) the entropy criterion $\iota_{entropy}$ as splitting criterion. Interpret the results: which of {D(t1,1), D(t1,2)} or {D(t2,1), D(t2,2)} is the better first split?

T1 → misclassification

$$\iota_{miss}(D) = 1 - \max\left(\frac{6}{24}, \frac{6}{24}, \frac{6}{24}, \frac{6}{24}\right) = 0.75$$

Subsets

$$\iota_{miss}(D1) = 1 - \max\left(\frac{6}{12}, \frac{6}{12}, \right) = 0.50$$

$$\iota_{miss}(D2) = 1 - \max\left(\frac{6}{12}, \frac{6}{12}, \right) = 0.50$$

$$\Delta\iota_{miss}(D\{D1, D2\}) = \iota_{miss}(D) - \left(\frac{12}{24} * \iota_{miss}(D1) + \frac{12}{24} * \iota_{miss}(D2)\right) = 0.25$$

T2 → entropy

$$\iota_{ent}(D) = 1 - \left(\frac{6}{24} * \log_2\left(\frac{6}{24}\right) + \frac{6}{24} * \log_2\left(\frac{6}{24}\right) + \frac{6}{24} * \log_2\left(\frac{6}{24}\right) + \frac{6}{24} * \log_2\left(\frac{6}{24}\right)\right) = 2$$

Subsets

$$\iota_{ent}(D1) = 1 - \left(\frac{4}{12} * \log_2\left(\frac{4}{12}\right) + \frac{2}{12} * \log_2\left(\frac{2}{12}\right) + \frac{6}{12} * \log_2\left(\frac{6}{12}\right)\right) = 1.50$$

$$\iota_{ent}(D2) = 1 - \left(\frac{2}{12} * \log_2\left(\frac{2}{12}\right) + \frac{4}{12} * \log_2\left(\frac{4}{12}\right) + \frac{6}{12} * \log_2\left(\frac{6}{12}\right)\right) = 1.50$$

$$\Delta\iota_{ent}(D\{D1, D2\}) = \iota_{ent}(D) - \left(\frac{12}{24} * \iota_{ent}(D1) + \frac{12}{24} * \iota_{ent}(D2)\right) = 0.5$$

Interpretation

Based in this 1st split we can identify as the better tree the T2. This is due the highest result of "Δι" is in T2, this indicates that this tree offers a split the reduce better the impurity.

b) If we compare T1 and T2 in terms of their misclassification rate on D, which one is the better decision tree?

In T1:

$$\iota_{miss}(leaf1) = 1 - \max\left(\frac{2}{5}, \frac{3}{5}\right) = 0.4$$

$$\iota_{miss}(leaf2) = 1 - \max\left(\frac{4}{7}, \frac{3}{7}\right) = 0.43$$

$$\iota_{miss}(leaf3) = 1 - \max\left(\frac{6}{6}, \frac{0}{6}\right) = 0.0$$

$$\iota_{miss}(leaf4) = 1 - \max\left(\frac{6}{6}, \frac{0}{6}\right) = 0.0$$

$$W\iota_{miss} = \left(\frac{5}{24} * 0.4 + \frac{7}{24} * 0.43 + \frac{6}{24} * 0 + \frac{6}{24} * 0\right) = 0.21$$

In T2:

$$\iota_{miss}(leaf1) = 1 - \max\left(\frac{4}{5}, \frac{1}{5}\right) = 0.2$$

$$\iota_{miss}(leaf2) = 1 - \max\left(\frac{6}{7}, \frac{1}{7}\right) = 0.14$$

$$\iota_{miss}(leaf3) = 1 - \max\left(\frac{4}{5}, \frac{1}{5}\right) = 0.2$$

$$\iota_{miss}(leaf4) = 1 - \max\left(\frac{6}{7}, \frac{1}{7}\right) = 0.14$$

$$W\iota_{miss} = \left(\frac{5}{24} * 0.4 + \frac{7}{24} * 0.43 + \frac{5}{24} * 0 + \frac{7}{24} * 0\right) = 0.17$$

The tree T2 is better due to his lower weighted error.

c) Assuming the splits shown are the only possibilities, which of T1 or T2 would the ID3 algorithm construct, and why?

The tree T2 would be the one constructed with entropy. And this is because ID3 search the feature that maximizes the impurity reduction "Δı".

Exercise 4 : Decision Trees (5+0=5 Points)

Given is the following dataset to classifiy whether a dog is dangerous or well-behaved in character:

Color	Fur	Size	Character (C)
brown	ragged	small	well-behaved
black	ragged	big	dangerous
black	smooth	big	dangerous
black	curly	small	well-behaved
white	curly	small	well-behaved
white	smooth	small	dangerous
red	ragged	big	well-behaved

a) Use the ID3 algorithm with ıentropy as the impurity function to determine the tree T.

$$\iota_{ent}(D) = 1 - \left(\frac{4}{7} * \log_2\left(\frac{4}{7}\right) + \frac{3}{7} * \log_2\left(\frac{3}{7}\right)\right) = 0.985$$

1st Split

– Color:

$$\iota_{ent}(color = brown) = 1 - \left(\frac{1}{1} * \log_2\left(\frac{1}{1}\right) + \frac{0}{1} * \log_2\left(\frac{0}{1}\right)\right) = 0$$

$$\iota_{ent}(color = black) = 1 - \left(\frac{1}{3} * \log_2\left(\frac{1}{3}\right) + \frac{2}{3} * \log_2\left(\frac{2}{3}\right)\right) = 0.918$$

$$\iota_{ent}(color = white) = 1 - \left(\frac{1}{2} * \log_2\left(\frac{1}{2}\right) + \frac{1}{2} * \log_2\left(\frac{1}{2}\right)\right) = 1$$

$$\iota_{ent}(color = red) = 1 - \left(\frac{1}{1} * \log_2\left(\frac{1}{1}\right) + \frac{0}{1} * \log_2\left(\frac{0}{1}\right)\right) = 0$$

$$W\iota_{ent} = \left(\frac{1}{7} * 0 + \frac{3}{7} * 0.918 + \frac{2}{7} * 1 + \frac{1}{7} * 0\right) = 0.679$$

$$\Delta\iota_{ent}(size) = \iota_{ent}(D) - W\iota_{ent} = 0.321$$

– Fur:

$$\iota_{ent}(fur = ragged) = 1 - \left(\frac{2}{3} * \log_2\left(\frac{2}{3}\right) + \frac{1}{3} * \log_2\left(\frac{1}{3}\right)\right) = 0.918$$

$$\iota_{ent}(fur = smooth) = 1 - \left(\frac{1}{1} * \log_2\left(\frac{1}{1}\right) + \frac{0}{1} * \log_2\left(\frac{0}{1}\right)\right) = 0$$

$$\iota_{ent}(fur = curly) = 1 - \left(\frac{1}{1} * \log_2\left(\frac{1}{1}\right) + \frac{0}{1} * \log_2\left(\frac{0}{1}\right)\right) = 0$$

$$W\iota_{ent} = \left(\frac{3}{7} * 0.918 + \frac{2}{7} * 0 + \frac{1}{7} * 0\right) = 0.393$$

$$\Delta\iota_{ent}(fur) = \iota_{ent}(D) - W\iota_{ent} = 0.592$$

– Size:

$$\iota_{ent}(size = small) = 1 - \left(\frac{3}{4} * \log_2\left(\frac{3}{4}\right) + \frac{1}{4} * \log_2\left(\frac{1}{4}\right)\right) = 0.811$$

$$\iota_{ent}(size = big) = 1 - \left(\frac{1}{3} * \log_2\left(\frac{1}{3}\right) + \frac{2}{3} * \log_2\left(\frac{2}{3}\right)\right) = 0.918$$

$$W\iota_{ent} = \left(\frac{4}{7} * 0.811 + \frac{3}{7} * 0.918\right) = 0.857$$

$$\Delta\iota_{ent}(size) = \iota_{ent}(D) - W\iota_{ent} = 0.143$$

- We pick the high $\Delta\iota_{ent}$. So, Fur is the best feature for 1st split.

2nd Split

$$\iota_{ent}(D_{ragged}) = 1 - \left(\frac{4}{7} * \log_2\left(\frac{4}{7}\right) + \frac{3}{7} * \log_2\left(\frac{3}{7}\right)\right) = 0.985$$

– Color:

$$\iota_{ent}(color = brown) = 1 - \left(\frac{1}{1} * \log_2\left(\frac{1}{1}\right) + \frac{0}{1} * \log_2\left(\frac{0}{1}\right)\right) = 0$$

$$\iota_{ent}(color = black) = 1 - \left(\frac{1}{1} * \log_2\left(\frac{1}{1}\right) + \frac{0}{1} * \log_2\left(\frac{0}{1}\right)\right) = 0$$

$$\iota_{ent}(color = white) = 1 - \left(\frac{0}{0} * \log_2\left(\frac{0}{0}\right) + \frac{0}{0} * \log_2\left(\frac{0}{0}\right)\right) = 0$$

$$\iota_{ent}(color = red) = 1 - \left(\frac{1}{1} * \log_2\left(\frac{1}{1}\right) + \frac{0}{1} * \log_2\left(\frac{0}{1}\right)\right) = 0$$

$$W\iota_{ent} = \left(\frac{1}{3} * 0 + \frac{1}{3} * 0 + \frac{0}{3} * 0 + \frac{1}{3} * 0\right) = 0$$

$$\Delta\iota_{ent}(size) = \iota_{ent}(D) - W\iota_{ent} = 0.918$$

– Size:

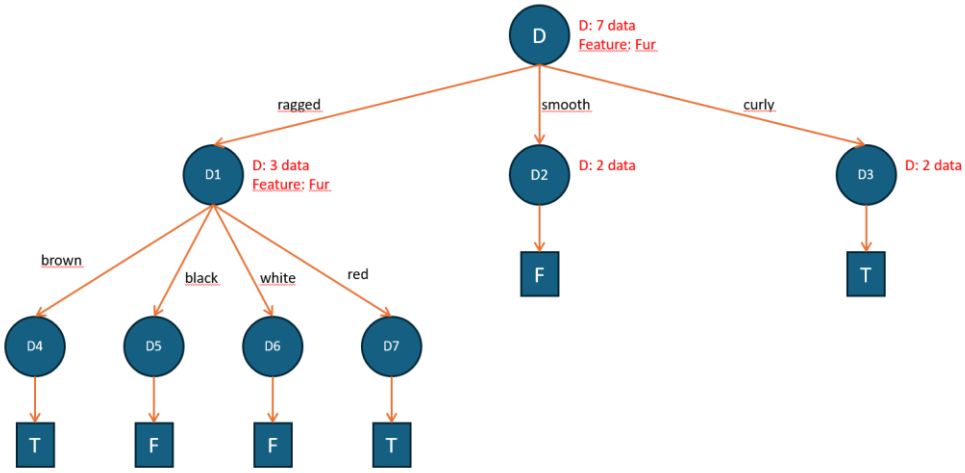
$$\iota_{ent}(size = small) = 1 - \left(\frac{0}{1} * \log_2\left(\frac{0}{1}\right) + \frac{1}{1} * \log_2\left(\frac{1}{1}\right)\right) = 0$$

$$\iota_{ent}(size = big) = 1 - \left(\frac{1}{2} * \log_2\left(\frac{1}{2}\right) + \frac{1}{2} * \log_2\left(\frac{1}{2}\right)\right) = 1$$

$$W\iota_{ent} = \left(\frac{1}{3} * 0 + \frac{2}{3} * 1\right) = 0.67$$

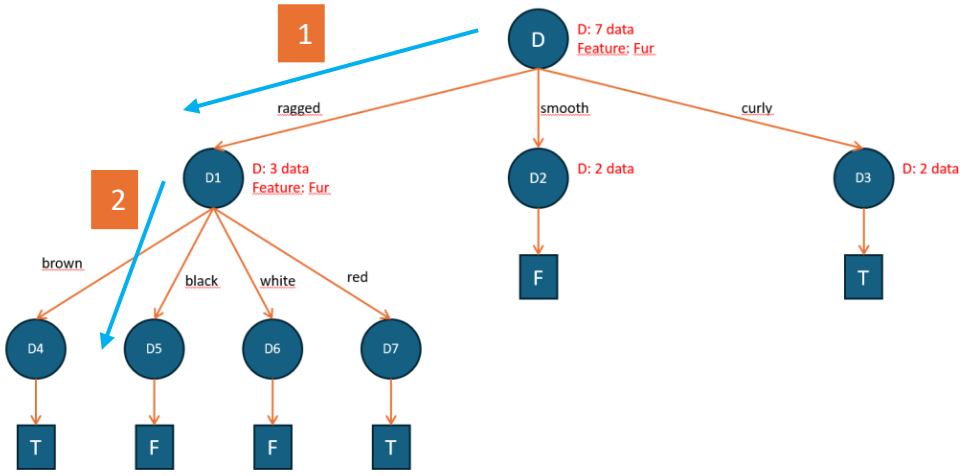
$$\Delta\iota_{ent}(size) = \iota_{ent}(D) - W\iota_{ent} = 0.248$$

- We pick the high $\Delta\iota_{ent}$. So, Size is the best feature for 2nd split.



b) Classify the new example (Color=black, Fur=ragged, Size=small) using T.

Following the upper tree we obtain that the new example is classified as "Dangerous". (same as false)



Exercise 5:

(a) Determine the labels of all nodes using the given cost function.

The cost function is defined as:

$$cost(c', c) = \begin{cases} 1 & \text{if } c' \neq c, c \in C \\ 0 & \text{otherwise} \end{cases}$$

1. Training Data:
   - Edibility: “toxic” or “edible.”
   - Nodes split by “Size” into small and large.
2. Determine labels:
   - For each leaf node, assign the most common label from the corresponding subset. If there’s a tie, any label can be assigned.
3. Subsets:
   - Size = small: Instances 1, 2, 4 → Toxic: 1, Edible: 2. Majority: Edible.
   - Size = large: Instances 3, 5 → Toxic: 1, Edible: 1. Tie, assign any (e.g., Edible).

(b) Devise a new cost function

The new cost function is designed to heavily penalize the classification of toxic mushrooms as edible. It can be expressed as:

$$\text{cost}(c', c) = \begin{cases} 10, & \text{if } c' = \text{Edible and } c = \text{Toxic}, \\ 1, & \text{if } c' \neq c \text{ and } c \neq \text{Toxic}, \\ 0, & \text{otherwise.} \end{cases}$$

The goal is to design a cost function that penalizes classifying a toxic mushroom as edible more heavily than other errors

This ensures that misclassifying toxic mushrooms has a much higher penalty than other types of errors.

(c) Compute the misclassification costs

For the given decision tree and training data:

Original Cost Function:

$$\text{cost}(c', c) = \begin{cases} 1, & \text{if } c' \neq c, \\ 0, & \text{otherwise.} \end{cases}$$

- Misclassification costs:

- Small subset: 1 toxic misclassified → Cost = 1
- Large subset: 1 toxic misclassified → Cost = 1
- Total Cost: 1 + 1 = 2

New Cost Function:

$$\text{cost}(c', c) = \begin{cases} 10, & \text{if } c' = \text{Edible and } c = \text{Toxic}, \\ 1, & \text{if } c' \neq c \text{ and } c \neq \text{Toxic}, \\ 0, & \text{otherwise.} \end{cases}$$

- Misclassification costs:
- Small subset: 1 toxic misclassified → Cost = 10
- Large subset: 1 toxic misclassified → Cost = 10
- Total Cost: 10 + 10 =20

Exercise 6 : P Classification with CART Decision Trees (1+1+1+1+1+1+1+1=8 Points)

In this exercise, you will implement the CART algorithm for constructing decision trees for predicting whether a given text was written by a human or generated by a language model. Submit the file with your predictions for the test set along with your other solutions.
Download and use these files from Moodle (the tsv files are the same as in the last sheet):

- features-train.tsv: Feature vectors for each example in the training set.
- features-test.tsv: Feature vectors for each example in the test set.
- labels-train.tsv: Quality scores for each example in the training set.
- programming_exercise_decision_trees.py: Template for the programming exercise. It contains function stubs for each function mentioned below, as well as functions implemented in the previous exercises. Use the following command to run the program:
  python3 programming_exercise_decision_trees.py Note: The program will read the above-mentioned tsv files from the data folder that should be in the same directory as the program.
- requirements.txt: Requirements file for the template; can be used to install dependencies.

a) Implement a function most_common_class to find the most common class in the dataset.
b) Implement a function gini_impurity that computes the Gini index for the given set of example classes C (slide ML:VI-79).
c) Implement a function gini_impurity_reduction that computes the Gini impurity reduction of a binary split (slide ML:VI-50).
d) Implement a function possible_thresholds that returns all possible thresholds for splitting the example set X along the given feature. Pick thresholds as the mid-point between all pairs of distinct, consecutive values in ascending order.
e) Implement a function find_best_split that finds the best split based on the Gini impurity reduction for the given set of examples X and the given set of classes C.
f) Implement the id3_cart function to construct a CART decision tree with the modified ID3 algorithm (slides ML:VI-109, ML:VI-22). The function should return the root node of the tree.
g) Implement a function train_and_predict. This function should train the model on the training set and return the predictions for the test set. What is the misclassification rate on the training set?
h) Run the plot_misclassification_rates function to plot the misclassification rate on the training set for different depths of the decision tree.

You will find the program and predictions in the folder attached with this pdf file.