CS145 Howework 2

Important Note: HW2 is due on **11:59 PM PT, May 1 (Monday)**. Please submit the exported pdf file on GradeScope.

Print Out Your Name and UID

Name: Joshua Mares, UID: 005154394

Before You Start

You need to first create HW2 conda environment specified in the `cs145hw2.yml` file. If you have `conda` properly installed, you may create, activate or deactivate using:

```
conda env create -f cs145hw2.yml
conda activate hw2
conda deactivate
```

OR

```
conda env create --name NAMEOFYOURCHOICE -f cs145hw2.yml
conda activate NAMEOFYOURCHOICE
conda deactivate
```

To view the list of all your environments, use the following command:

```
conda env list
```

Conda Docs (https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html)

You must not delete any code cells in this notebook. If you change any code outside the blocks that you are allowed to edit (between `STRART/END YOUR CODE HERE`), you need to highlight these changes. You may add some additional cells to help explain your results and observations.

In [1]:
```
import numpy as np
import pandas as pd
%load_ext autoreload
%autoreload 2
```

If you can successfully run the code above, there will be no problem for environment setting.

1. Decision trees

1.1 Attribute Selection

For classifiers, misclassification rate is commonly used to measure the final performance. However, when selecting which attribute to split in a decision tree, we often use other measures, such as information gain, gain ratio, and Gini index. Let's investigate these different measures through the following problem.

Note: To compute the misclassification rate of a classification tree with N data points, K classes, and M leaf nodes:

In a node m, m = 1, ..., M, let N_m denote the number of data points, and N_{mk} denote the number of data points in class k. The prediction under majority vote can thus be written as $j = argmax_k N_{mk}$. The misclassification rate of this node is $R_m = 1 - \frac{N_{mj}}{N_m}$. The misclassification rate of the entire tree would be $R = \frac{\sum_{m=1}^{M} R_m * N_m}{N}$

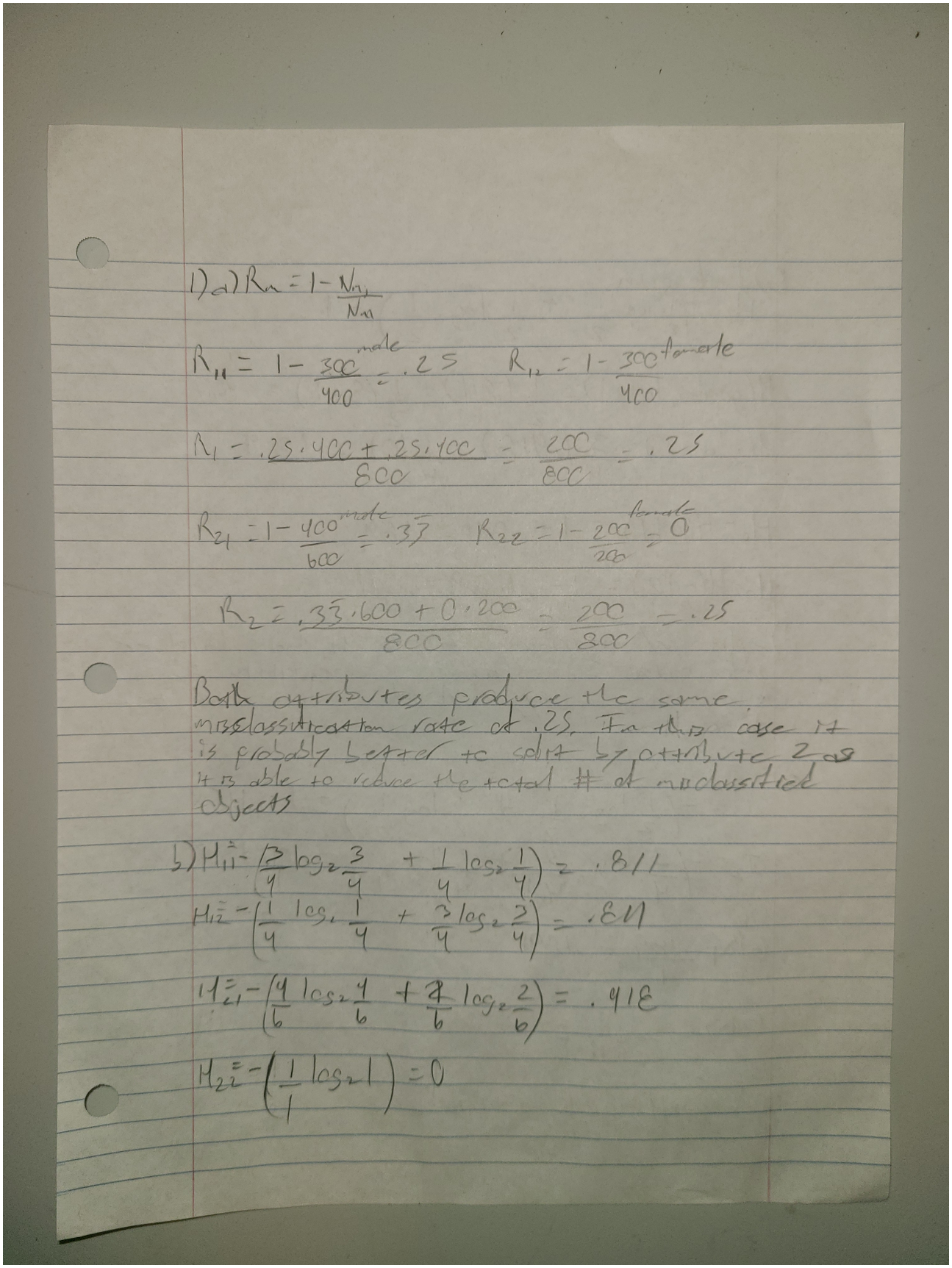
Questions

Note: this question does not require any coding.

Suppose our dataset includes a total of 800 people with 400 males and 400 females, and our goal is to perform gender classification. Consider two potential attributes we can split on in a decision tree. Splitting on the first attribute results in a node11 with 300 male and 100 female, and a node12 with 100 men and 300 women. Splitting on the second attribute results in a node21 with 400 men and 200 women, and a node22 with no men and 200 women.

1. Which attribute would you prefer if we optimize for misclassifcation rate, and why?
2. What is the entropy of each of these four node?
3. What is the information gain of each of the two splits?
4. Which attribute do you prefer if we optimize for measurement is information gain?
5. What is the gain ratio (normalized information gain) of each of the two splits? Which attribute would you prefer if we optimize for gain ratio?

Answers



1) a) $R_n = 1 - \frac{N_{m_i}}{N_m}$

$R_{11} = 1 - \frac{300}{400}^{\text{male}} = .25 \quad R_{12} = 1 - \frac{300}{400}^{\text{female}}$

$R_1 = \frac{.25 \cdot 400 + .25 \cdot 400}{800} = \frac{200}{800} = .25$

$R_{21} = 1 - \frac{400}{600}^{\text{male}} = .33 \quad R_{22} = 1 - \frac{200}{200}^{\text{female}} = 0$

$R_2 = \frac{.33 \cdot 600 + 0 \cdot 200}{800} = \frac{200}{800} = .25$

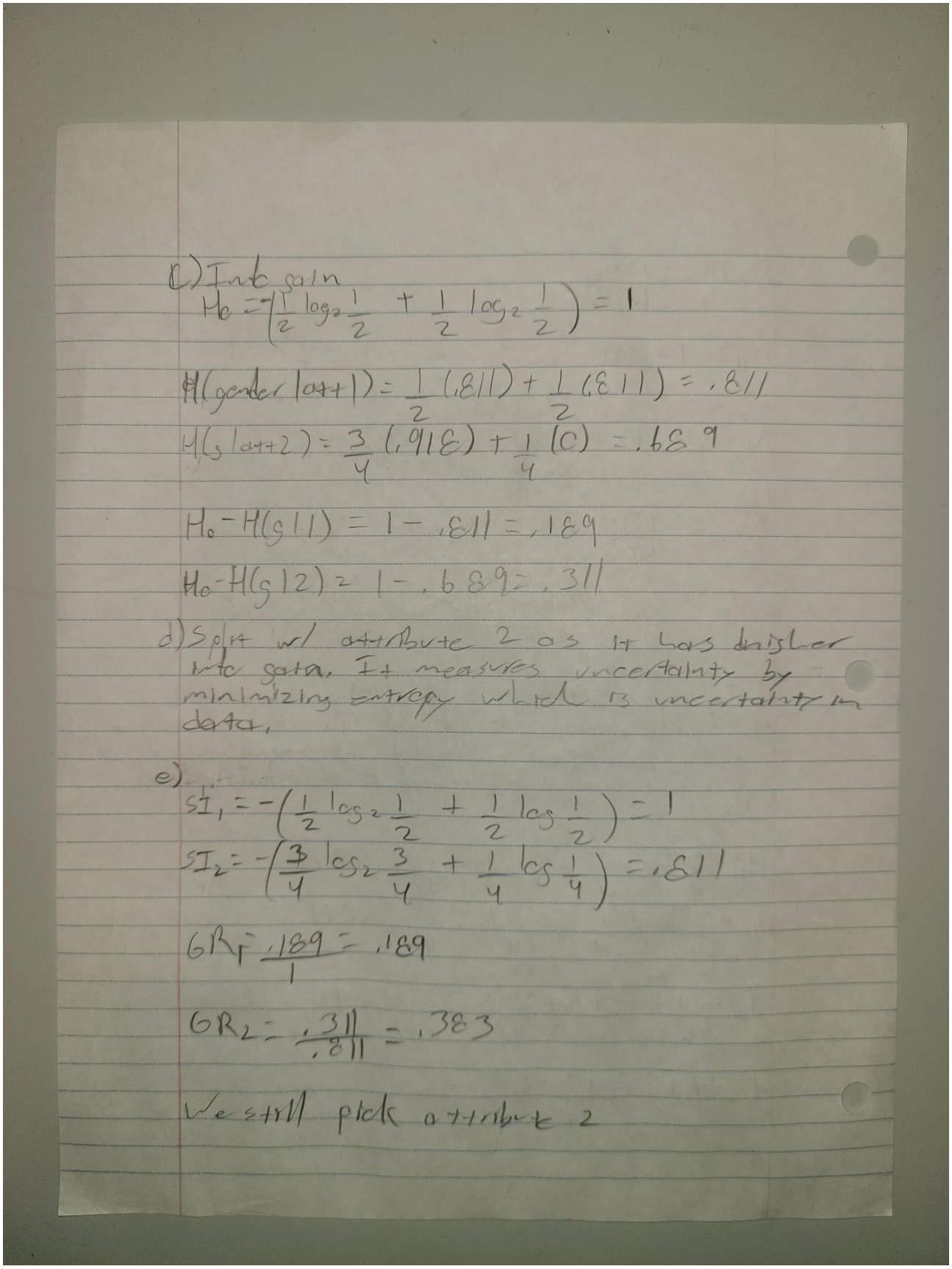
Both attributes produce the same misclassification rate of .25. For this case it is probably better to split by attribute 2 as it is able to reduce the total # of misclassified objects

b) $H_{11} = -\left(\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right) = .811$

$H_{12} = -\left(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}\right) = .811$

$H_{21} = -\left(\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6}\right) = .918$

$H_{22} = -\left(\frac{1}{1}\log_2 1\right) = 0$



1.2 Implementation

In this section, we are going to build a decision tree model to predict the the animal `type` . We will use the `zoo` dataset, which has been preprocessed and splited into `decision-tree-train.csv` and `decision-tree-test.csv` for you.

In [2]:
```
from hw2code.decision_tree import DecisionTree
mytree = DecisionTree()
mytree.load_data('./data/decision-tree-train.csv','./data/decision-tree-test.csv')
# The size of the training data: (80, 17) and testing data: (21, 17)
print('Training data shape: ', mytree.train_data.shape)
print('Testing data shape:', mytree.test_data.shape)
```

```
Training data shape:  (80, 17)
Testing data shape: (21, 17)
```

1.2.1 Infomation gain

Complete the `make_tree` and `compute_info_gain` function in `decision_tree.py` .

Train you model using `info_gain` measure to classify `type` and print the test accuracy.

Hint: the test accuracy should be above 80%.

In [3]:
```
mytree = DecisionTree()
mytree.load_data('./data/decision-tree-train.csv','./data/decision-tree-test.csv')
test_acc= 0
#=======================#
# START YOUR CODE HERE  #
#=======================#
mytree.train(y_name="type", measure="info_gain")
test_acc = mytree.test("type")
#=======================#
#   END YOUR CODE HERE   #
#=======================#
print('Test accuracy is: ', test_acc)
```

```
best_feature is:  legs
best_feature is:  fins
best_feature is:  toothed
best_feature is:  eggs
best_feature is:  hair
best_feature is:  hair
best_feature is:  toothed
best_feature is:  aquatic
Test accuracy is:  0.8571428571428571
```

1.2.2 Gain ratio

Complete the `compute_gain_ratio` function in `decision_tree.py` .

Train you model using `gain_ratio` measure to classify `type` and print the test accuracy.

Hint: the test accuracy should be above 80%.

In [4]:
```
mytree = DecisionTree()
mytree.load_data('./data/decision-tree-train.csv','./data/decision-tree-test.csv')
test_acc = 0
#=======================#
# START YOUR CODE HERE  #
#=======================#
mytree.train(y_name="type", measure="gain_ratio")
test_acc = mytree.test("type")
#=======================#
#   END YOUR CODE HERE   #
#=======================#
print('Test accuracy is: ', test_acc)
```

```
best_feature is:  feathers
best_feature is:  backbone
best_feature is:  airborne
best_feature is:  predator
best_feature is:  milk
best_feature is:  fins
best_feature is:  legs
Test accuracy is:  0.8095238095238095
```

End of Homework 2