Homework 4

1 Exercises

1.1.2 Consider the training examples shown in Table 3.5 for a binary classification problem

a) Compute the Gini index for the overall collection of training examples.

Gini index = $1 - (2 \times 0.5^2) = 0.5$

b) Compute the Gini index for the Customer ID attribute.

Customer ID is a unique identifier, meaning each customer has a distinct ID, so splitting by Customer ID would create a pure partition for each instance , which means Gini = 0 for each partition which would make the overall Gini also 0.

c) Compute the Gini index for the Gender attribute.

Gini(Male) = $1 - (6/10)^2 - (4/10)^2 = 0.48$

Gini(Female) = $1 - (6/10)^2 - (4/10)^2 = 0.48$

Gini(Gender) = $(10/20)*0.48 + (10/20)*0.48 = 0.48$

d) Compute the Gini index for the Car Type attribute using multiway split.

Gini(Family) = $1 - (1/4)^2 - (3/4)^2 = 0.375$

Gini(Sports) = $1 - (8/8)^2 - (0/8)^2 = 0$

Gini(Luxury) = $1 - (1/8)^2 - (7/8)^2 = 0.2188$

Gini(Car Type) = $(4/20)*0.375 + (8/20)*0 + (8/20)*0.2188 = 0.1625$

e) Compute the Gini index for the Shirt Size attribute using multiway split.

Gini(Small) = $1 - (3/5)^2 - (2/5)^2 = 0.48$

Gini(Medium) = $1 - (3/7)^2 - (4/7)^2 = 0.4898$

Gini(Large) = $1 - (2/4)^2 - (2/4)^2 = 0.5$

Gini(Extra Large) = $1 - (2/4)^2 - (2/4)^2 = 0.5$

Gini(Shirt Size) = $(5/20)*0.48 + (7/20)*0.4898 + (4/20)*0.5 + (4/20)*0.5 = 0.4914$

f) Which attribute is better, Gender, Car Type, or Shirt Size?

Car Type because it has the lowest gini among the three attributes.

g) Explain why Customer ID should not be used as the attribute test condition even though it has the lowest Gini.

Customer ID is a unique identifier with no generalization power, it doesn't provide any meaningful insights into the relationships between attributes or classes. Therefore, it's not useful for predicting future instances.

1.1.3 Consider the training examples shown in Table 3.6 for a binary classification problem

a. What is the entropy of this collection of training examples with respect to the class attribute?

Entropy = $-(4/9)*\log_2(4/9) - (5/9)*\log_2(5/9) = 0.9911$

b. What are the information gains of a1 and a2 relative to these training examples?

Entropy for a1:

$(4/9)\ [-(3/4)*\log_2(3/4) - (1/4)*\log_2(1/4)] + (5/9)\ [-(1/5)*\log_2(1/5) - (4/5)*\log_2(4/5)] = 0.7616$

Information gain for a1: $0.9911 - 0.7616 = 0.2294$

Entropy for a2:
$(5/9)\ [-(2/5)*\log_2(2/5) - (3/5)*\log_2(3/5)] + (4/9)\ [-(2/4)*\log_2(2/4) - (2/4)*\log_2(2/4)] = 0.9839$
Information gain for a2: 0.9911 - 0.9829 = 0.0072

c. For a3, which is a continuous attribute, compute the information gain for every possible split.
The values of a3 are: 1.0, 3.0, 4.0, 5.0, 5.0, 6.0, 7.0, 7.0, 8.0. The possible split points are the midpoints between these values:
(1.0 + 3.0) / 2 = 2.0
(3.0 + 4.0) / 2 = 3.5
(4.0 + 5.0) / 2 = 4.5
(5.0 + 6.0) / 2 = 5.5
(6.0 + 7.0) / 2 = 6.5
(7.0 + 8.0) / 2 = 7.5

The entropy for the parent set is H(S) = 0.9911.
Information Gain for Each Split:
Split at 2.0: Gain = 0.1427
Split at 3.5: Gain = 0.0026
Split at 4.5: Gain = 0.0728
Split at 5.5: Gain = 0.0072
Split at 6.5: Gain = 0.0183
Split at 7.5: Gain = 0.1022

The best split point for a3 based on the information gain is at 2.0, since it has the highest information gain of 0.1427.

d. What is the best split (among a1, a2 and a2) according to the information gain?
a1 produces the best split because it has the largest delta difference in entropy.

e. What is the best split (between a1 and a2) according to the misclassification error rate?
a1:
CE(T) = 1 - max[3/4, 1/4] = 1 - 3/4 = 0.25
CE(F) = 1 - max[1/5, 4/5] = 1 - 4/5 = 0.2
error rate(a1) = (4/9)*0.25 + (5/9)*0.2 = 0.2222

a2:
CE(T) = 1 - max[2/5, 3/5] = 1 - 3/5 = 0.4
CE(F) = 1 - max[2/4, 2/4] = 1 - 1/2 = 0.5
error rate(a2) = (5/9)*0.4 + (4/9)*0.5 = 0.4444

Since the misclassification error rate for a1 is smaller, it produces the better split

f. What is the best split (between a1 and a2) according to the Gini index?

a1:
Gini(T) = 1 - $(3/4)^2$ - $(1/4)^2$ = 0.375
Gini(F) = 1 - $(1/5)^2$ - $(4/5)^2$ = 0.32
Gini(a1) = (4/9)*0.375 + (5/9)*0.32 = 0.3444

a2:
Gini(T) = 1 - $(2/5)^2$ - $(3/5)^2$ = 0.48
Gini(F) = 1 - $(2/4)^2$ - $(2/4)^2$ = 0.5
Gini(a2) = (5/9)*0.48 + (4/9)*0.5 = 0.4889

Since the gini index for a1 is smaller, it produces the better split

1.1.5 Consider the following data set for a binary class problem.
a. Calculate the information gain when splitting on A and B. Which attribute would the decision tree induction algorithm choose?
The overall entropy before splitting: $-0.4\log(0.4) - 0.6\log(0.6) = 0.9710$
The information gain after splitting on A is:

$E_{A=T} = -(4/7)\log(4/7) - (3/7)\log(3/7) = 0.9852$
$E_{A=F} = -(3/3)\log(3/3) - (0/3)\log(0/3) = 0$
$\Delta = 0.9710 - (7/10)*0.9852 - (3/10)*0 = 0.2813$

The information gain after splitting on B is:

$E_{B=T} = -(3/4)\log(3/4) - (1/4)\log(1/4) = 0.8113$
$E_{B=F} = -(1/6)\log(1/6) - (5/6)\log(5/6) = 0.65$
$\Delta = 0.9710 - (4/10)*0.8113 - (6/10)*0.65 = 0.2565$

Therefore, attribute A will be chosen to split the node since its information gain is larger.
b. Calculate the gain in the Gini index when splitting on A and B. Which attribute would the decision tree induction algorithm choose?
The overall Gini index before splitting: $1 - 0.4^2 - 0.6^2 = 0.48$
The gain in the Gini index after splitting on A is:

$G_{A=T} = 1 - (4/7)^2 - (3/7)^2 = 0.4898$
$G_{A=F} = 1 - (3/3)^2 - (0/3)^2 = 0$
$\Delta = 0.48 - (7/10)*0.4898 - (3/10)*0 = 0.1371$

The gain in the Gini index after splitting on B is:

$G_{B=T} = 1 - (1/4)^2 - (3/4)^2 = 0.3750$
$G_{B=F} = 1 - (1/6)^2 - (5/6)^2 = 0.2778$
$\Delta = 0.48 - (4/10)*0.3750 - (6/10)*0.2778 = 0.1633$

Therefore, attribute B will be chosen to split the node.

c. Figure 3.11 shows that entropy and the Gini index are both monotonically increasing on the range [0, 0.5] and they are both monotonically decreasing on the range [0.5, 1]. Is it possible that information gain and the gain in the Gini index favor different attributes? Explain.

Yes, in some cases, an attribute that results in moderately pure splits may have higher information gain but lower Gini gain. Conversely, an attribute that produces splits with more extreme class distributions may have a higher gain in the Gini index but lower information gain. Therefore, the two metrics can favor different attributes in decision tree induction even though these measures have similar range and are monotonous. We can see this illustrated by the results in parts (a) and (b).

1.2 Consider the task of building a classifier from random data, where the attribute values are generated randomly irrespective of the class labels. Assume the data set contains instances from two classes, "+" and "-." Half of the data set is used for training while the remaining half is used for testing.

a. Suppose there are an equal number of positive and negative instances in the data and the decision tree classifier predicts every test instance to be positive. What is the expected error rate of the classifier on the test data?

Given that half of the data is used for testing, and there is an equal number of positive and negative instances. Let N be the number of test instances. We know that there are N/2 positive instances and N/2 negative instances.

The classifier predicts every test instance as positive:

- It will correctly predict the N/2 positive instances.
- It will incorrectly classify the N/2 negative instances as positive.

The expected error rate is the proportion of incorrectly classified instances:

Error rate = (Number of incorrect predictions) / (Total test instances) = (N/2) / N = 0.5

b. Repeat the previous analysis assuming that the classifier predicts each test instance to be positive class with probability 0.8 and negative class with probability 0.2.

In this case, the classifier predicts each test instance as positive with a probability of 0.8, and negative with a probability of 0.2.

For the N/2 positive instances:

- It correctly predicts them as positive with a probability of 0.8
- Thus, the error for positive instances is $1 - 0.8 = 0.2$

For the N/2 negative instances:

- It incorrectly predicts them as positive with a probability of 0.8.
- So, the error for negative instances is 0.8.

Error rate = [0.2*(N/2) + 0.8*(N/2)] / N = 0.5

c. Suppose two-thirds of the data belong to the positive class and the remaining one-third belong to the negative class. What is the expected error of a classifier that predicts every test instance to be positive?

In this case, two-thirds (2N/3) of the test instances are positive and one-third (N/3) of the test instances are negative.

The classifier predicts every test instance as positive:

- It correctly classifies the 2N/3 positive instances.
- It incorrectly classifies the N/3 negative instances as positive.

Error rate = (Number of incorrect predictions) / (Total test instances) = (N/3) / N = 0.33

d. Repeat the previous analysis assuming that the classifier predicts each test instance to be positive class with probability 2/3 and negative class with probability 1/3.

The classifier predicts positive with probability 2/3 and negative with probability 1/3.

For the 2N/3 positive instances:

- It correctly predicts them as positive with a probability of 2/3.
- Thus, the error for positive instances is $1 - 2/3 = 1/3$

For the N/3 negative instances:

- It incorrectly predicts them as positive with a probability of 2/3.
- So, the error for negative instances is 2/3.

Error rate = [(2N/3)*(1/3) + (N/3)*(2/3)] / N

Error rate = (2N/9 + 2N/9) / N

Error rate = 4/9 = 0.444.

1.3 Using the confusion matrix from multiclass.Rmd notebook (from Lecture 7), create a binary-class confusion matrix using the "one-vs-many" strategy for each class. Then, for each class, compute the sensitivity, specificity and precision to two decimal places. Show all work, including the binary class confusion matrices.

2 Practicum problem

Q2.1(b)(iii) Between sensitivity and specificity, focus on the value that is lower between the two, and provide some justification for why that is the case.

Specificity is lower than sensitivity because the model is better at identifying employees who are likely to leave ('Yes') compared to predicting employees who will stay ('No'). This could be due to the imbalance in the dataset or features that are more predictive of attrition

Q2.1(b)(v) Do you think this is a good model? Justify your answer.
This model may not be optimal, as decision trees can overfit the training data and perform poorly on unseen data, especially with imbalanced datasets like this one. However, the AUC score indicates moderate predictive power, suggesting the model can still provide some valuable insights.

Q2.2(b)(ii) Do you think this is a good model? Justify your answer.
This model is likely better than the original unbalanced model since it addresses the class imbalance issue. However, decision trees can still suffer from overfitting, and additional validation techniques like cross-validation may be necessary for more robust results.

Q2.3(c) Did the pruning help improve the tree?
Pruning helped improve the model by reducing overfitting, which is evident in the confusion matrix results, where the accuracy or balance between sensitivity and specificity shows a favorable improvement.