DOKUZ EYLUL UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CME 3402 CONCEPTS OF PROGRAMMING LANGUAGES

ASSIGNMENT 1: DECISION TREE CONSTRUCTION PYTHON

By

Boran Bereketli - 2022510105

Ramazan Denli - 2022510111

Emre Akkaya - 2022510085

Lecturers

Asst.Prof.Dr. Yunus Doğan Res.Asst.

Fatih Dicle

Res.Asst. Muharremcan Gülye

IZMIR

28.05.2025

1. Introduction

This study involves building and analyzing a decision tree using a classification dataset. The main objective is to develop a program capable of predicting outcomes based on input data. The implementation should support the creation of decision trees that vary in size and attribute configuration.

2. Tested Dataset Descriptions

Weather Dataset

Filename:weather.csv

Number of records: 14

Features:

- outlook: (overcast, rainy, sunny)
- temperature: (cool, mild, hot)
- humidity: (normal, high)
- windy: (TRUE, FALSE)

Target label: play (yes, no)

Contact Lenses Dataset

Filename: contact_lenses.csv

Number of records: 24

Features:

- age: (young , pre-presbyopic, presbyopic)
- spectacle-prescrip: (hypermetrope, myope)
- astigmatism: (yes, no)
- tear-prod-rate: (normal, reduced)

Target label: contact-lenses (hard, none, soft)

Breast Cancer Dataset

Filename: breast_cancer.csv

Number of records: 277

Features:

- age: (20-29, 30-39, 40-49, 50-59, 60-69, 70-79)
- menopause: (ge40, lt40, premeno)

- tumor-size: (0-4, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 5-9, 50-54)
- inv-nodes: (0-2, 12-14, 15-17, 24-26, 3-5, 6-8, 9-11)
- node-caps: (yes, no)
- deg-malig: (1, 2, 3)
- breast: (right, left)
- breast-quad: (central, left_low, left_up, right_low, right_up)
- irradiat: (yes, no)

Target label: Class (no-recurrence-events, recurrence-events)

3. Decision Tree Construction

Weather dataset is used as an example.

Entropy Formula

Entropy is calculated by the given formula.

$$Entropy(S) = -\sum_{i=1}^{n} p_i \cdot \log_2(p_i)$$

Example:

```python
def entropy(rows):
    counts = result_counts(rows)
    length = len(rows)
    probabilities = [count / length for count in counts.values()]
    if length == 0:
        return 0
    return -sum(p * log2(p) for p in probabilities if p > 0)
```

Information Gain Formula

```python
def information_gain(left, right, current_uncertainty):
    p = float(len(left)) / (len(left) + len(right))
    return current_uncertainty - p * entropy(left) - (1 - p) * entropy(right)
```

Information gain is calculated by:

$$InformationGain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v)$$

Example:

```
Total entropy of current node: 0.940

Evaluating attribute: outlook
  sunny (5): 3 no, 2 yes → Entropy = 0.971
  overcast (4): 4 yes → Entropy = -0.000
  rainy (5): 3 yes, 2 no → Entropy = 0.971
  Gain(S, outlook) = 0.940 - Weighted Entropy = 0.694 → Gain = 0.247
```

3.3 Comparison of Attributes

Information gain is calculated for every attribute first.

```
Evaluating attribute: outlook
  sunny (5): 3 no, 2 yes → Entropy = 0.971
  overcast (4): 4 yes → Entropy = -0.000
  rainy (5): 3 yes, 2 no → Entropy = 0.971
  Gain(S, outlook) = 0.940 - Weighted Entropy = 0.694 → Gain = 0.247

Evaluating attribute: temperature
  hot (4): 2 no, 2 yes → Entropy = 1.000
  mild (6): 4 yes, 2 no → Entropy = 0.918
  cool (4): 3 yes, 1 no → Entropy = 0.811
  Gain(S, temperature) = 0.940 - Weighted Entropy = 0.911 → Gain = 0.029

Evaluating attribute: humidity
  high (7): 4 no, 3 yes → Entropy = 0.985
  normal (7): 6 yes, 1 no → Entropy = 0.592
  Gain(S, humidity) = 0.940 - Weighted Entropy = 0.788 → Gain = 0.152

Evaluating attribute: windy
  FALSE (8): 2 no, 6 yes → Entropy = 0.811
  TRUE (6): 3 no, 3 yes → Entropy = 1.000
  Gain(S, windy) = 0.940 - Weighted Entropy = 0.892 → Gain = 0.048
```

Outlook has highest gain.

3.5 Loop

This procedure is repeated for each subtree until the entire decision tree is built. Each node splits according to the possible values of the selected attribute. For instance, in the screenshot above, the subtree corresponding to outlook = sunny is currently being generated.Then a leaf node is found, it can no longer branch so the result is used.

```
Evaluating attribute: outlook
 Only one unique value. Skipping.

Evaluating attribute: temperature
  mild (3): 2 yes, 1 no → Entropy = 0.918
  cool (2): 1 yes, 1 no → Entropy = 1.000
  Gain(S, temperature) = 0.971 - Weighted Entropy = 0.951 → Gain = 0.020

Evaluating attribute: humidity
  high (2): 1 yes, 1 no → Entropy = 1.000
  normal (3): 2 yes, 1 no → Entropy = 0.918
  Gain(S, humidity) = 0.971 - Weighted Entropy = 0.951 → Gain = 0.020

Evaluating attribute: windy
  FALSE (3): 3 yes → Entropy = -0.000
  TRUE (2): 2 no → Entropy = -0.000
  Gain(S, windy) = 0.971 - Weighted Entropy = 0.000 → Gain = 0.971

Best attribute : windy (Gain = 0.971)
```

4. Final Decision Tree

Console output is given below.

```
Decision Tree:
outlook:
  sunny ->
    humidity:
      high ->
        Prediction: no
      normal ->
        Prediction: yes
  overcast ->
    Prediction: yes
  rainy ->
    windy:
      FALSE ->
        Prediction: yes
      TRUE ->
        Prediction: no
```

5. Program Execution and Prediction

Interactive User Input

Inputs are case-insensitive. If tree does not have the result it will be unknown. The program allows manual entry of values. Attributes are entered one by one.

```
Enter feature values for prediction (leave blank to exit):
  outlook: overcast
  temperature: hot
  humidity: high
  windy: FALSE

Prediction: yes

Enter feature values for prediction (leave blank to exit):
  outlook: rainy
  temperature: mild
  humidity: high
  windy: FALSE

Prediction: yes

Enter feature values for prediction (leave blank to exit):
  outlook: sunny
  temperature: hot
  humidity: high
  windy: TRUE

Prediction: no
```

6. Other Decision Trees

Breast Cancer Dataset

Too large to show.

```
deg-malig:
  3 ->
    inv-nodes:
      0-2 ->
        tumor-size:
          15-19 ->
            age:
              40-49 ->
                Prediction: recurrence-events
              30-39 ->
                Prediction: no-recurrence-events
              60-69 ->
                Prediction: no-recurrence-events
          35-39 ->
            age:
              40-49 ->
                Prediction: no-recurrence-events
              30-39 ->
                Prediction: recurrence-events
              50-59 ->
                Prediction: no-recurrence-events
          40-44 ->
            Prediction: no-recurrence-events
          20-24 ->
            age:
              30-39 ->
                breast-quad:
                  central ->
                    Prediction: no-recurrence-events
                  left_up ->
                    Prediction: recurrence-events
              50-59 ->
                Prediction: no-recurrence-events
              40-49 ->
                Prediction: no-recurrence-events
              60-69 ->
                Prediction: recurrence-events
              70-79 ->
                Prediction: no-recurrence-events
          30-34 ->
            breast-quad:
              central ->
                Prediction: recurrence-events
              right_up ->
                age:
                  50-59 ->
                    Prediction: recurrence-events
                  40-49 ->
                    node-caps:
```

Contact Lenses Dataset

```
Decision Tree:
tear-prod-rate:
  reduced ->
    Prediction: none
  normal ->
    astigmatism:
      no ->
        age:
          young ->
            Prediction: soft
          pre-presbyopic ->
            Prediction: soft
          presbyopic ->
            spectacle-prescrip:
              myope ->
                Prediction: none
              hypermetrope ->
                Prediction: soft
      yes ->
        spectacle-prescrip:
          myope ->
            Prediction: hard
          hypermetrope ->
            age:
              young ->
                Prediction: hard
              pre-presbyopic ->
                Prediction: none
              presbyopic ->
                Prediction: none
```

7. Conclusion

The decision tree has been successfully built for given datasets with different sizes and attributes. It correctly classifies instances based on input data. In the weather example, the view attribute was identified as the most informative root node and the tree was built accordingly. The model is able to make accurate predictions using the attributes and values in the dataset.