MACHINE LEARNING Laboratory

5th Semester, Academic Year 2025

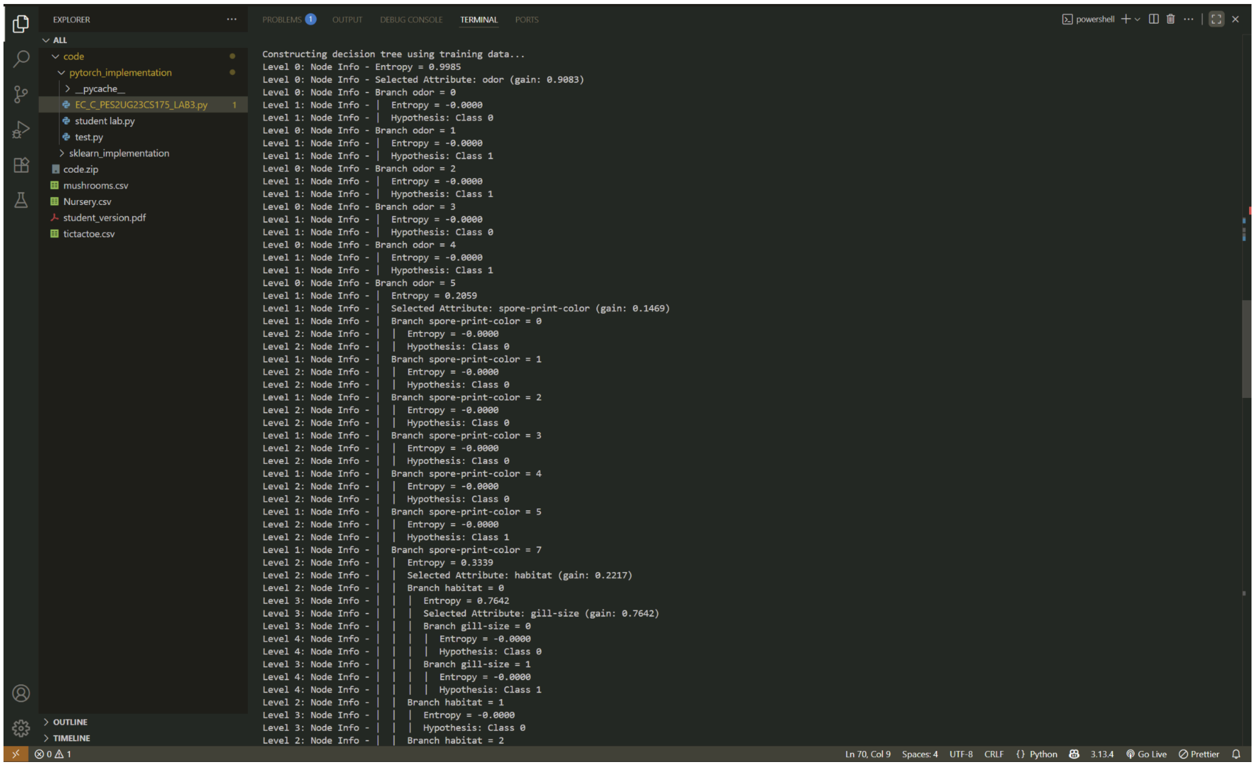
Date:19-08-2025

Name: Dhruv Thakur	SRN:PES2UG23CS175	Section C

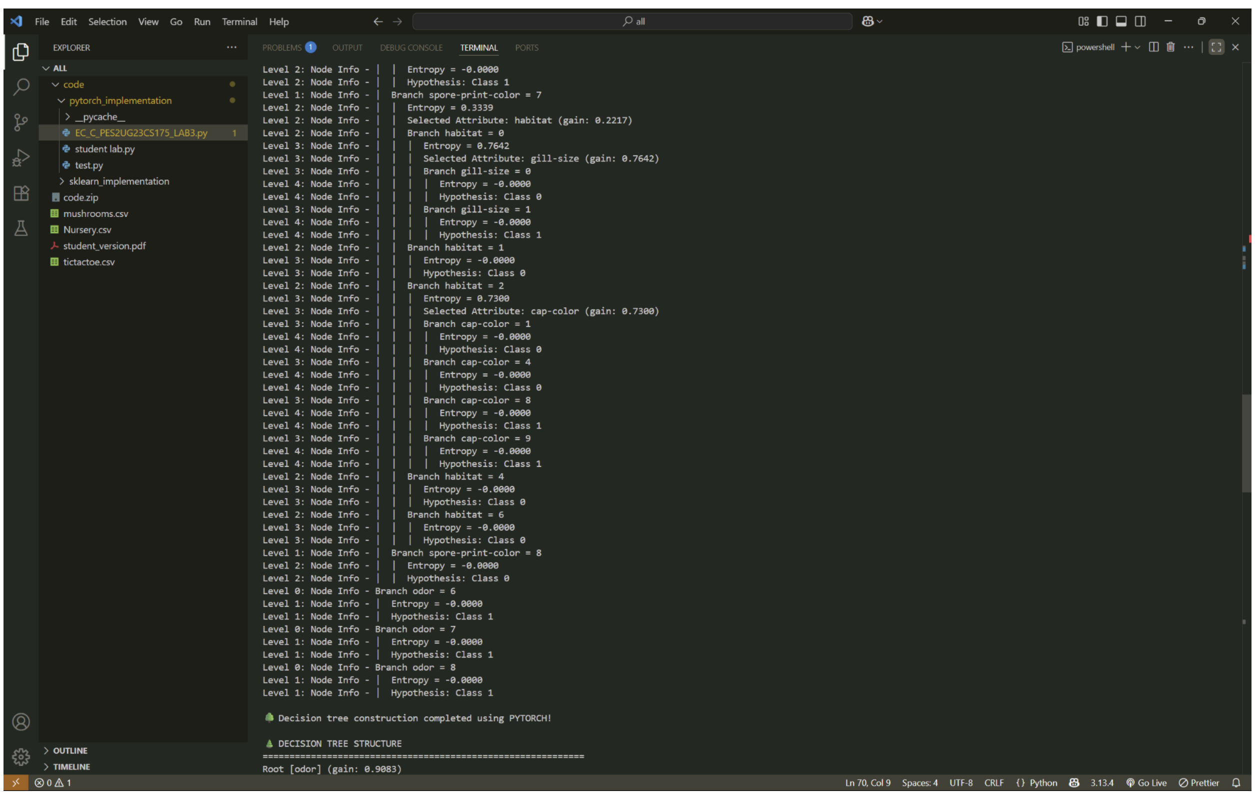
Week#____3______

MUSHROOMS.CSV

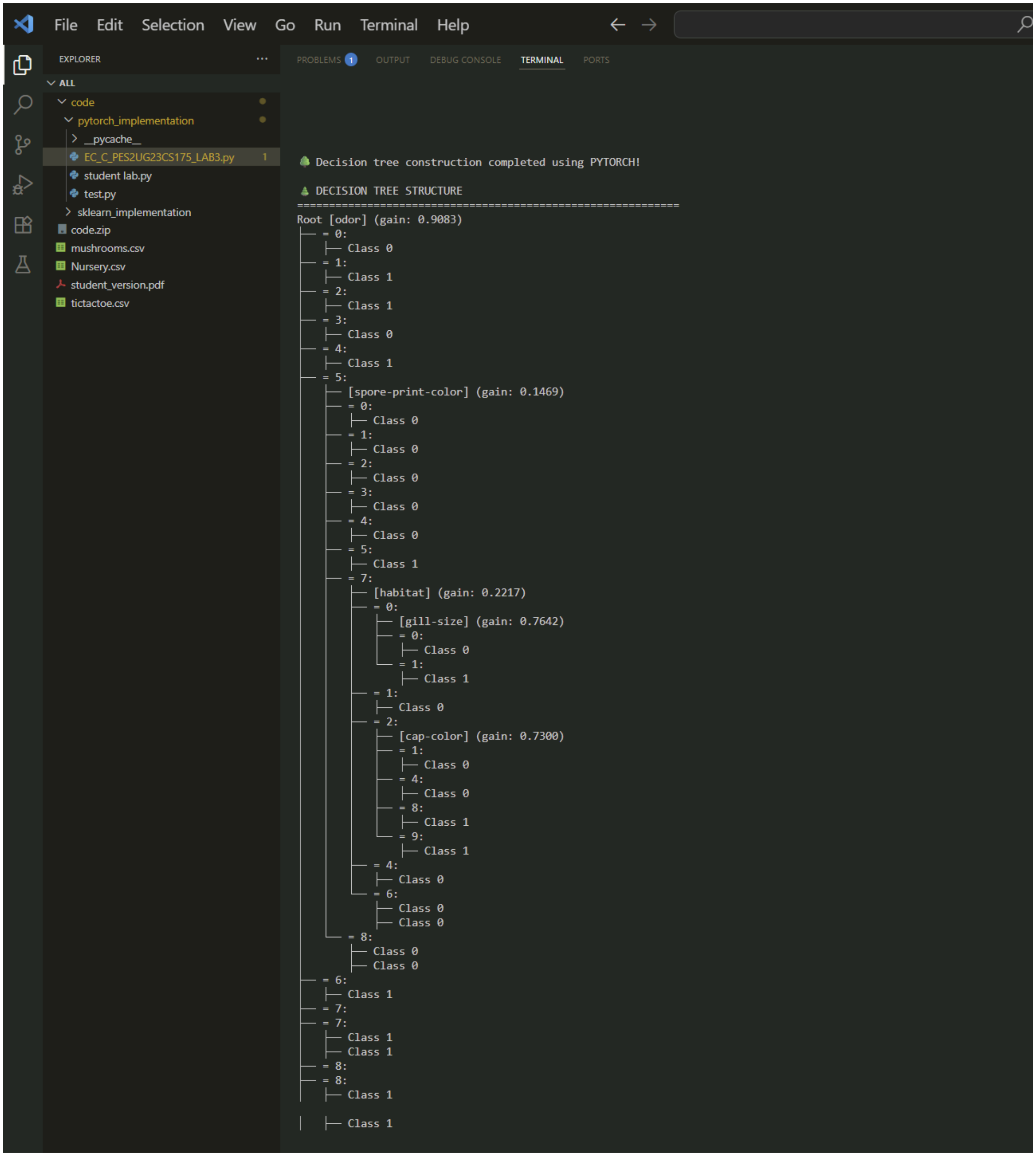
```
PS C:\Users\Dhruv Thakur\Downloads\all> python code\pytorch_implementation\test.py --ID EC_C_PES2UG23CS175_LAB3 --data mushrooms.csv --framework pytorch --print-tree --print-construc
tion
Running tests with PYTORCH framework
============================================================
 target column: 'class' (last column)
Original dataset info:
Shape: (8124, 23)
Columns: ['cap-shape', 'cap-surface', 'cap-color', 'bruises', 'odor', 'gill-attachment', 'gill-spacing', 'gill-size', 'gill-color', 'stalk-shape', 'stalk-root', 'stalk-surface-above-
ring', 'stalk-surface-below-ring', 'stalk-color-above-ring', 'stalk-color-below-ring', 'veil-type', 'veil-color', 'ring-number', 'ring-type', 'spore-print-color', 'population', 'habi
tat', 'class']

First few rows:

cap-shape: ['x' 'b' 's' 'f' 'k'] -> [5 0 4 2 3]

cap-surface: ['s' 'y' 'f' 'g'] -> [2 3 0 1]

cap-color: ['n' 'y' 'w' 'g' 'e'] -> [4 9 8 3 2]

class: ['p' 'e'] -> [1 0]

Processed dataset shape: torch.Size([8124, 23])
Number of features: 22
Features: ['cap-shape', 'cap-surface', 'cap-color', 'bruises', 'odor', 'gill-attachment', 'gill-spacing', 'gill-size', 'gill-color', 'stalk-shape', 'stalk-root', 'stalk-surface-above
-ring', 'stalk-surface-below-ring', 'stalk-color-above-ring', 'stalk-color-below-ring', 'veil-type', 'veil-color', 'ring-number', 'ring-type', 'spore-print-color', 'population', 'hab
itat']
Target: class
Framework: PYTORCH
Data type: <class 'torch.Tensor'>

============================================================
DECISION TREE CONSTRUCTION DEMO
============================================================
Total samples: 8124
Training samples: 6499
Testing samples: 1625
```



```
Constructing decision tree using training data...
Level 0: Node Info - Entropy = 0.9985
Level 0: Node Info - Selected Attribute: odor (gain: 0.9083)
Level 0: Node Info - Branch odor = 0
Level 1: Node Info - |  Entropy = -0.0000
Level 1: Node Info - |  Hypothesis: Class 0
Level 0: Node Info - Branch odor = 1
Level 1: Node Info - |  Entropy = -0.0000
Level 1: Node Info - |  Hypothesis: Class 1
Level 0: Node Info - Branch odor = 2
Level 1: Node Info - |  Entropy = -0.0000
Level 1: Node Info - |  Hypothesis: Class 1
Level 0: Node Info - Branch odor = 3
Level 1: Node Info - |  Entropy = -0.0000
Level 1: Node Info - |  Hypothesis: Class 0
Level 0: Node Info - Branch odor = 4
Level 1: Node Info - |  Entropy = -0.0000
Level 1: Node Info - |  Hypothesis: Class 1
Level 0: Node Info - Branch odor = 5
Level 1: Node Info - |  Entropy = 0.2059
Level 1: Node Info - |  Selected Attribute: spore-print-color (gain: 0.1469)
Level 1: Node Info - |  Branch spore-print-color = 0
Level 2: Node Info - |  |  Entropy = -0.0000
Level 2: Node Info - |  |  Hypothesis: Class 0
Level 1: Node Info - |  Branch spore-print-color = 1
Level 2: Node Info - |  |  Entropy = -0.0000
Level 2: Node Info - |  |  Hypothesis: Class 0
Level 1: Node Info - |  Branch spore-print-color = 2
Level 2: Node Info - |  |  Entropy = -0.0000
Level 2: Node Info - |  |  Hypothesis: Class 0
Level 1: Node Info - |  Branch spore-print-color = 3
Level 2: Node Info - |  |  Entropy = -0.0000
Level 2: Node Info - |  |  Hypothesis: Class 0
Level 1: Node Info - |  Branch spore-print-color = 4
Level 2: Node Info - |  |  Entropy = -0.0000
Level 2: Node Info - |  |  Hypothesis: Class 0
Level 1: Node Info - |  Branch spore-print-color = 5
Level 2: Node Info - |  |  Entropy = -0.0000
Level 2: Node Info - |  |  Hypothesis: Class 1
Level 1: Node Info - |  Branch spore-print-color = 7
Level 2: Node Info - |  |  Entropy = 0.3339
Level 2: Node Info - |  |  Selected Attribute: habitat (gain: 0.2217)
Level 2: Node Info - |  |  Branch habitat = 0
Level 3: Node Info - |  |  |  Entropy = 0.7642
Level 3: Node Info - |  |  |  Selected Attribute: gill-size (gain: 0.7642)
Level 3: Node Info - |  |  |  Branch gill-size = 0
Level 4: Node Info - |  |  |  |  Entropy = -0.0000
Level 4: Node Info - |  |  |  |  Hypothesis: Class 0
Level 3: Node Info - |  |  |  Branch gill-size = 1
Level 4: Node Info - |  |  |  |  Entropy = -0.0000
Level 4: Node Info - |  |  |  |  Hypothesis: Class 1
Level 2: Node Info - |  |  Branch habitat = 1
Level 3: Node Info - |  |  |  Entropy = -0.0000
Level 3: Node Info - |  |  |  Hypothesis: Class 0
Level 2: Node Info - |  |  Branch habitat = 2
```



```
Level 2: Node Info - |  |  Entropy = -0.0000
Level 2: Node Info - |  |  Hypothesis: Class 1
Level 1: Node Info - |  Branch spore-print-color = 7
Level 2: Node Info - |  |  Entropy = 0.3339
Level 2: Node Info - |  |  Selected Attribute: habitat (gain: 0.2217)
Level 2: Node Info - |  |  Branch habitat = 0
Level 3: Node Info - |  |  |  Entropy = 0.7642
Level 3: Node Info - |  |  |  Selected Attribute: gill-size (gain: 0.7642)
Level 3: Node Info - |  |  |  Branch gill-size = 0
Level 4: Node Info - |  |  |  |  Entropy = -0.0000
Level 4: Node Info - |  |  |  |  Hypothesis: Class 0
Level 3: Node Info - |  |  |  Branch gill-size = 1
Level 4: Node Info - |  |  |  |  Entropy = -0.0000
Level 4: Node Info - |  |  |  |  Hypothesis: Class 1
Level 2: Node Info - |  |  Branch habitat = 1
Level 3: Node Info - |  |  |  Entropy = -0.0000
Level 3: Node Info - |  |  |  Hypothesis: Class 0
Level 2: Node Info - |  |  Branch habitat = 2
Level 3: Node Info - |  |  |  Entropy = 0.7300
Level 3: Node Info - |  |  |  Selected Attribute: cap-color (gain: 0.7300)
Level 3: Node Info - |  |  |  Branch cap-color = 1
Level 4: Node Info - |  |  |  |  Entropy = -0.0000
Level 4: Node Info - |  |  |  |  Hypothesis: Class 0
Level 3: Node Info - |  |  |  Branch cap-color = 4
Level 4: Node Info - |  |  |  |  Entropy = -0.0000
Level 4: Node Info - |  |  |  |  Hypothesis: Class 0
Level 3: Node Info - |  |  |  Branch cap-color = 8
Level 4: Node Info - |  |  |  |  Entropy = -0.0000
Level 4: Node Info - |  |  |  |  Hypothesis: Class 1
Level 3: Node Info - |  |  |  Branch cap-color = 9
Level 4: Node Info - |  |  |  |  Entropy = -0.0000
Level 4: Node Info - |  |  |  |  Hypothesis: Class 1
Level 2: Node Info - |  |  Branch habitat = 4
Level 3: Node Info - |  |  |  Entropy = -0.0000
Level 3: Node Info - |  |  |  Hypothesis: Class 0
Level 2: Node Info - |  |  Branch habitat = 6
Level 3: Node Info - |  |  |  Entropy = -0.0000
Level 3: Node Info - |  |  |  Hypothesis: Class 0
Level 1: Node Info - |  Branch spore-print-color = 8
Level 2: Node Info - |  |  Entropy = -0.0000
Level 2: Node Info - |  |  Hypothesis: Class 0
Level 0: Node Info - Branch odor = 6
Level 1: Node Info - |  Entropy = -0.0000
Level 1: Node Info - |  Hypothesis: Class 1
Level 0: Node Info - Branch odor = 7
Level 1: Node Info - |  Entropy = -0.0000
Level 1: Node Info - |  Hypothesis: Class 1
Level 0: Node Info - Branch odor = 8
Level 1: Node Info - |  Entropy = -0.0000
Level 1: Node Info - |  Hypothesis: Class 1

🌳 Decision tree construction completed using PYTORCH!

🌲 DECISION TREE STRUCTURE
============================================================
Root [odor] (gain: 0.9083)
```


File Edit Selection View Go Run Terminal Help
EXPLORER
ALL
code
pytorch_implementation
__pycache__
EC_C_PES2UG23CS175_LAB3.py
student lab.py
test.py
sklearn_implementation
code.zip
mushrooms.csv
Nursery.csv
student_version.pdf
tictactoe.csv
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Decision tree construction completed using PYTORCH!
DECISION TREE STRUCTURE
==
Root [odor] (gain: 0.9083)
├── = 0:
│ ├── Class 0
├── = 1:
│ ├── Class 1
├── = 2:
│ ├── Class 1
├── = 3:
│ ├── Class 0
├── = 4:
│ ├── Class 1
├── = 5:
│ ├── [spore-print-color] (gain: 0.1469)
│ ├── = 0:
│ │ ├── Class 0
│ ├── = 1:
│ │ ├── Class 0
│ ├── = 2:
│ │ ├── Class 0
│ ├── = 3:
│ │ ├── Class 0
│ ├── = 4:
│ │ ├── Class 0
│ ├── = 5:
│ │ ├── Class 1
│ ├── = 7:
│ │ ├── [habitat] (gain: 0.2217)
│ │ ├── = 0:
│ │ │ ├── [gill-size] (gain: 0.7642)
│ │ │ ├── = 0:
│ │ │ │ ├── Class 0
│ │ │ └── = 1:
│ │ │ ├── Class 1
│ │ ├── = 1:
│ │ │ ├── Class 0
│ │ ├── = 2:
│ │ │ ├── [cap-color] (gain: 0.7300)
│ │ │ ├── = 1:
│ │ │ │ ├── Class 0
│ │ │ ├── = 4:
│ │ │ │ ├── Class 0
│ │ │ ├── = 8:
│ │ │ │ ├── Class 1
│ │ │ └── = 9:
│ │ │ ├── Class 1
│ │ ├── = 4:
│ │ │ ├── Class 0
│ │ └── = 6:
│ │ ├── Class 0
│ │ ├── Class 0
│ └── = 8:
│ ├── Class 0
│ ├── Class 0
├── = 6:
│ ├── Class 1
├── = 7:
├── = 7:
│ ├── Class 1
│ ├── Class 1
├── = 8:
├── = 8:
│ ├── Class 1
│ ├── Class 1

File Edit Selection View Go Run Terminal Help
EXPLORER
ALL
code
pytorch_implementation
__pycache__
EC_C_PES2UG23CS175_LAB3.py 1
student lab.py
test.py
sklearn_implementation
code.zip
mushrooms.csv
Nursery.csv
student_version.pdf
tictactoe.csv
OUTLINE
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
= 8:
Class 1
= 9:
Class 1
= 4:
Class 0
= 6:
Class 0
Class 0
= 8:
Class 0
Class 0
= 6:
Class 1
= 7:
= 7:
Class 1
Class 1
= 8:
= 8:
Class 1
Class 1
OVERALL PERFORMANCE METRICS
OVERALL PERFORMANCE METRICS
==
OVERALL PERFORMANCE METRICS
==
Accuracy: 1.0000 (100.00%)
Precision (weighted): 1.0000
Recall (weighted): 1.0000
==
Accuracy: 1.0000 (100.00%)
Precision (weighted): 1.0000
Recall (weighted): 1.0000
Accuracy: 1.0000 (100.00%)
Precision (weighted): 1.0000
Recall (weighted): 1.0000
Recall (weighted): 1.0000
F1-Score (weighted): 1.0000
Precision (macro): 1.0000
Recall (macro): 1.0000
F1-Score (macro): 1.0000
Recall (macro): 1.0000
F1-Score (macro): 1.0000
F1-Score (macro): 1.0000
TREE COMPLEXITY METRICS
TREE COMPLEXITY METRICS
==
TREE COMPLEXITY METRICS
==
Maximum Depth: 4
==
Maximum Depth: 4
Maximum Depth: 4
Total Nodes: 29
Leaf Nodes: 24
Internal Nodes: 5
PS C:\Users\Dhruv Thakur\Downloads\all>

NURSERY.CSV

File Edit Selection View Go Run Terminal Help

EXPLORER
- ALL
  - code
    - pytorch_implementation
      - __pycache__
      - EC_C_PES2UG23CS175_LAB3.py 1
      - student lab.py
      - test.py
    - sklearn_implementation
    - code.zip
    - mushrooms.csv
    - Nursery.csv
    - student_version.pdf
    - tictactoe.csv
- OUTLINE
- TIMELINE

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
                    ├─ Class 1
                - 2:
                    ├─ Class 1
                - 3:
                    ├─ Class 1
            - 1:
                ├─ Class 1
            - 2:
                ├─ Class 1
            - 3:
                ├─ Class 1
        - 2:
            ├─ [children] (gain: 0.5370)
            - 0:
                ├─ [form] (gain: 0.5917)
                - 0:
                    ├─ Class 4
                - 1:
                    ├─ Class 4
                - 2:
                    ├─ Class 1
                - 3:
                    ├─ Class 4
            - 1:
                ├─ [form] (gain: 1.0000)
                - 0:
                    ├─ Class 4
                - 1:
                    ├─ Class 4
                - 2:
                    ├─ Class 1
                - 3:
                    ├─ Class 1
            - 2:
                ├─ Class 1
            - 3:
                ├─ Class 1
- 2:
    ├─ [social] (gain: 0.1711)
    - 0:
        ├─ [children] (gain: 0.1577)
        - 0:
            ├─ [form] (gain: 0.4513)
            - 0:
                ├─ Class 4
            - 1:
                ├─ [housing] (gain: 0.7219)
                - 0:
                    ├─ Class 4
                - 1:
                    ├─ Class 1
                - 2:
                    ├─ Class 4
            - 2:
                ├─ [housing] (gain: 0.3219)
                - 0:
                    ├─ Class 1
                - 1:
                    ├─ Class 1
                - 2:
                    ├─ Class 1
            - 3:
                ├─ Class 4
        - 1:
            ├─ [form] (gain: 0.3601)
            - 0:
                ├─ [housing] (gain: 1.0000)
                - 1:
                    ├─ Class 1
                - 2:
                    ├─ Class 4
            - 1:
                ├─ [housing] (gain: 0.9183)
                - 0:
                    ├─ Class 4
                - 1:
                    ├─ Class 1
                - 2:
                    ├─ Class 4
            - 2:
                ├─ Class 1
            - 3:
                ├─ Class 1
        - 2:
            ├─ [housing] (gain: 0.3215)
            - 0:
                ├─ [finance] (gain: 1.0000)
                - 0:
                    ├─ Class 4
                - 1:
                    ├─ Class 1
            - 1:
                ├─ Class 1
            - 2:
                ├─ Class 1
        - 3:
            ├─ [housing] (gain: 0.2669)
            - 0:
                ├─ [finance] (gain: 0.9852)
                - 0:
                    ├─ Class 4
                - 1:
                    ├─ Class 1
            - 1:
                ├─ Class 1
            - 2:
                ├─ Class 1
    - 1:
        ├─ Class 1
    - 2:
        ├─ [housing] (gain: 0.2669)
        - 0:
            ├─ [finance] (gain: 0.4233)
            - 0:
                ├─ [form] (gain: 0.3912)
                - 0:
                    ├─ Class 2
                - 1:
                    ├─ Class 4
                - 2:
                    ├─ Class 4
                - 3:
                    ├─ Class 4
            - 1:
                ├─ [form] (gain: 0.5710)
                - 0:
                    ├─ Class 4
                - 1:
                    ├─ Class 1
                - 2:
                    ├─ Class 1
                - 3:
                    ├─ Class 1
        - 1:
            ├─ [children] (gain: 0.1769)
            - 0:
                ├─ [form] (gain: 0.9183)
```

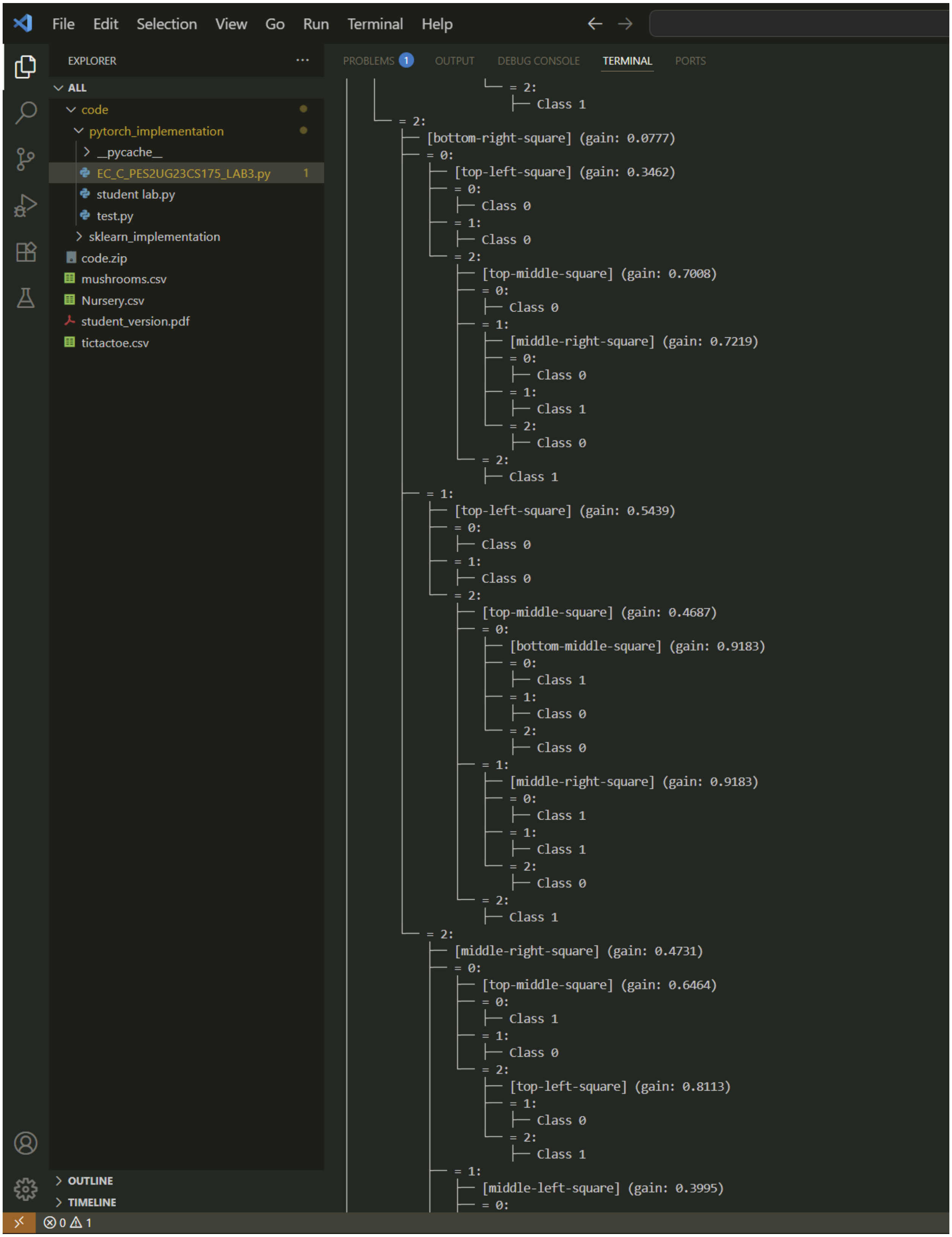
File
Edit
Selection
View
Go
Run
Terminal
Help
EXPLORER
ALL
code
pytorch_implementation
__pycache__
EC_C_PES2UG23CS175_LAB3.py
student lab.py
test.py
sklearn_implementation
code.zip
mushrooms.csv
Nursery.csv
student_version.pdf
tictactoe.csv
OUTLINE
TIMELINE
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
[social] (gain: 0.1355)
- 0:
[housing] (gain: 0.2568)
- 0:
[finance] (gain: 0.4099)
- 0:
Class 1
- 1:
[children] (gain: 0.4312)
- 0:
[form] (gain: 0.8113)
- 0:
Class 1
- 1:
Class 1
- 2:
Class 3
- 3:
Class 1
- 1:
[form] (gain: 0.9940)
- 0:
Class 1
- 1:
Class 1
- 2:
Class 3
- 3:
Class 3
- 2:
Class 3
- 3:
Class 3
- 1:
[form] (gain: 0.0887)
- 0:
[children] (gain: 0.6723)
- 0:
Class 1
- 1:
Class 3
- 2:
Class 3
- 3:
Class 3
- 1:
Class 3
- 2:
Class 3
- 3:
Class 3
- 2:
[children] (gain: 0.4779)
- 0:
[form] (gain: 0.8524)
- 0:
Class 1
- 1:
Class 1
- 2:
Class 3
- 3:
Class 1
- 1:
[form] (gain: 0.9495)
- 0:
Class 1
- 1:
Class 1
- 2:
Class 3
- 3:
Class 3
- 2:
Class 3
- 3:
Class 3
- 1:
[form] (gain: 0.0182)
- 0:
[children] (gain: 0.0748)
- 0:
[housing] (gain: 0.3060)
- 0:
[finance] (gain: 1.0000)
- 0:
Class 1
- 1:
Class 3
- 1:
Class 3
- 2:
Class 3
- 1:
Class 3
- 2:
Class 3
- 3:
Class 3
- 1:
Class 3
- 2:
Class 3
- 3:
Class 3
- 2:
[housing] (gain: 0.2060)
- 0:
[finance] (gain: 0.4994)
- 0:
Class 1
- 1:
[children] (gain: 0.4516)
- 0:
[form] (gain: 0.8113)
- 0:
Class 1
- 1:
Class 1
- 2:
Class 3
- 3:
Class 1
- 1:
[form] (gain: 0.8631)
- 0:
Class 1
- 1:
Class 1
- 2:
Class 3
- 3:
Class 3
- 2:
Class 3
- 3:
Class 3

```
📊 OVERALL PERFORMANCE METRICS
========================================
Accuracy:              0.9867 (98.67%)
Precision (weighted): 0.9876
Recall (weighted):    0.9867
F1-Score (weighted):  0.9872
Precision (macro):    0.7604
Recall (macro):       0.7654
F1-Score (macro):     0.7628

🌲 TREE COMPLEXITY METRICS
========================================
Maximum Depth:        7
Total Nodes:          952
Leaf Nodes:           680
Internal Nodes:       272
PS C:\Users\Dhruv Thakur\Downloads\all>
```

tictactoe.csv

```
🌳 Decision tree construction completed using PYTORCH!

🌲 DECISION TREE STRUCTURE
============================================================
Root [middle-middle-square] (gain: 0.0834)
├── = 0:
│   ├── [bottom-left-square] (gain: 0.1056)
│   ├── = 0:
│   │   ├── [top-right-square] (gain: 0.9024)
│   │   ├── = 1:
│   │   │   ├── Class 0
│   │   └── = 2:
│   │       ├── Class 1
│   ├── = 1:
│   │   ├── [top-right-square] (gain: 0.2782)
│   │   ├── = 0:
│   │   │   ├── Class 0
│   │   ├── = 1:
│   │   │   ├── Class 0
│   │   └── = 2:
│   │       ├── [top-left-square] (gain: 0.1767)
│   │       ├── = 0:
│   │       │   ├── [bottom-right-square] (gain: 0.9183)
│   │       │   ├── = 1:
│   │       │   │   ├── Class 0
│   │       │   └── = 2:
│   │       │       ├── Class 1
│   │       ├── = 1:
│   │       │   ├── [top-middle-square] (gain: 0.6058)
│   │       │   ├── = 0:
│   │       │   │   ├── [middle-left-square] (gain: 0.9183)
│   │       │   │   ├── = 1:
│   │       │   │   │   ├── Class 0
│   │       │   │   └── = 2:
│   │       │   │       ├── Class 1
│   │       │   ├── = 1:
│   │       │   │   ├── Class 1
│   │       │   └── = 2:
│   │       │       ├── Class 0
│   │       └── = 2:
│   │           ├── [top-middle-square] (gain: 0.3393)
│   │           ├── = 0:
│   │           │   ├── [middle-left-square] (gain: 0.9183)
│   │           │   ├── = 0:
│   │           │   │   ├── Class 0
│   │           │   ├── = 1:
│   │           │   │   ├── Class 1
│   │           │   └── = 2:
│   │           │       ├── Class 0
│   │           ├── = 1:
│   │           │   ├── [middle-left-square] (gain: 0.9183)
│   │           │   ├── = 0:
│   │           │   │   ├── Class 1
│   │           │   ├── = 1:
│   │           │   │   ├── Class 1
│   │           │   └── = 2:
│   │           │       ├── Class 0
│   │           └── = 2:
│   │               ├── Class 1
│   └── = 2:
│       ├── [top-right-square] (gain: 0.1225)
│       ├── = 0:
│       │   ├── Class 1
│       ├── = 1:
│       │   ├── [middle-right-square] (gain: 0.1682)
```

```
        └── = 2:
            ├── Class 1
─── = 1:
    ├── [top-right-square] (gain: 0.0223)
    ├── = 0:
    │   ├── [bottom-left-square] (gain: 0.2247)
    │   ├── = 0:
    │   │   ├── Class 0
    │   ├── = 1:
    │   │   ├── Class 0
    │   └── = 2:
    │       ├── [middle-right-square] (gain: 0.1159)
    │       ├── = 0:
    │       │   ├── [top-left-square] (gain: 0.1771)
    │       │   ├── = 0:
    │       │   │   ├── [middle-left-square] (gain: 0.9183)
    │       │   │   ├── = 0:
    │       │   │   │   ├── Class 1
    │       │   │   ├── = 1:
    │       │   │   │   ├── Class 1
    │       │   │   └── = 2:
    │       │   │       ├── Class 0
    │       │   ├── = 1:
    │       │   │   ├── [bottom-right-square] (gain: 0.9710)
    │       │   │   ├── = 1:
    │       │   │   │   ├── Class 0
    │       │   │   └── = 2:
    │       │   │       ├── Class 1
    │       │   └── = 2:
    │       │       ├── Class 1
    │       ├── = 1:
    │       │   ├── [middle-left-square] (gain: 0.9887)
    │       │   ├── = 0:
    │       │   │   ├── Class 1
    │       │   ├── = 1:
    │       │   │   ├── Class 0
    │       │   └── = 2:
    │       │       ├── Class 1
    │       └── = 2:
    │           ├── [bottom-middle-square] (gain: 0.2400)
    │           ├── = 0:
    │           │   ├── [top-left-square] (gain: 1.0000)
    │           │   ├── = 1:
    │           │   │   ├── Class 0
    │           │   └── = 2:
    │           │       ├── Class 1
    │           ├── = 1:
    │           │   ├── Class 0
    │           └── = 2:
    │               ├── [bottom-right-square] (gain: 0.9710)
    │               ├── = 1:
    │               │   ├── Class 0
    │               └── = 2:
    │                   ├── Class 1
    ├── = 1:
    │   ├── [bottom-left-square] (gain: 0.4759)
    │   ├── = 0:
    │   │   ├── Class 0
    │   ├── = 1:
    │   │   ├── Class 0
    │   └── = 2:
    │       ├── [top-middle-square] (gain: 0.1974)
    │       ├── = 0:
    │       │   ├── Class 1
    │       ├── = 1:
    │       │   ├── [top-left-square] (gain: 0.3436)
    │       │   ├── = 0:
```


File Edit Selection View Go Run Terminal Help
EXPLORER
ALL
code
pytorch_implementation
__pycache__
EC_C_PES2UG23CS175_LAB3.py
student lab.py
test.py
sklearn_implementation
code.zip
mushrooms.csv
Nursery.csv
student_version.pdf
tictactoe.csv
OUTLINE
TIMELINE
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
= 2:
Class 1
= 2:
[bottom-right-square] (gain: 0.0777)
= 0:
[top-left-square] (gain: 0.3462)
= 0:
Class 0
= 1:
Class 0
= 2:
[top-middle-square] (gain: 0.7008)
= 0:
Class 0
= 1:
[middle-right-square] (gain: 0.7219)
= 0:
Class 0
= 1:
Class 1
= 2:
Class 0
= 2:
Class 1
= 1:
[top-left-square] (gain: 0.5439)
= 0:
Class 0
= 1:
Class 0
= 2:
[top-middle-square] (gain: 0.4687)
= 0:
[bottom-middle-square] (gain: 0.9183)
= 0:
Class 1
= 1:
Class 0
= 2:
Class 0
= 1:
[middle-right-square] (gain: 0.9183)
= 0:
Class 1
= 1:
Class 1
= 2:
Class 0
= 2:
Class 1
= 2:
[middle-right-square] (gain: 0.4731)
= 0:
[top-middle-square] (gain: 0.6464)
= 0:
Class 1
= 1:
Class 0
= 2:
[top-left-square] (gain: 0.8113)
= 1:
Class 0
= 2:
Class 1
= 1:
[middle-left-square] (gain: 0.3995)
= 0:

```
📊 OVERALL PERFORMANCE METRICS
========================================
Accuracy:             0.8730 (87.30%)
Precision (weighted): 0.8741
Recall (weighted):    0.8730
F1-Score (weighted):  0.8734
Precision (macro):    0.8590
Recall (macro):       0.8638
F1-Score (macro):     0.8613

🌳 TREE COMPLEXITY METRICS
========================================
Maximum Depth:        7
Total Nodes:          281
Leaf Nodes:           180
Internal Nodes:       101
PS C:\Users\Dhruv Thakur\Downloads\all>
```

1. Performance Comparison (Accuracy, Precision, Recall, F1)

Dataset	Accuracy	Precision (Weighted)	Recall (Weighted)	F1 (Weighted)
Mushroom	1.0000 (100%)	1.0000	1.0000	1.0000
Nursery	0.9867 (98.67%)	0.9876	0.9867	0.9872
TicTacToe	0.8730 (87.30%)	0.8741	0.8730	0.8734

Mushroom dataset achieves perfect accuracy (100%), likely because the dataset is clean, balanced, and attributes clearly separate classes.
Nursery dataset also achieves very high performance (98.7%), but slightly lower due to more complex/multi-valued attributes.
TicTacToe is the hardest, with 87.3% accuracy, showing possible class imbalance or overlap in decision patterns.

2. Tree Characteristics Analysis

Dataset	Max Depth	Total Nodes	Leaf Nodes	Internal Nodes
Mushroom	4	29	24	5
Nursery	7	952	680	272
TicTacToe	7	281	180	101

Mushroom tree is shallow (depth=4) → very simple, interpretable, and still achieves perfect accuracy → attributes are very strong predictors.
Nursery tree is very large (952 nodes) → indicates many attributes and class combinations → dataset complexity is high.
TicTacToe tree is smaller than Nursery but deeper than Mushroom (depth=7) → reflects more balanced but tricky classification.

3. Dataset-Specific Insights

Mushroom Dataset

Feature Importance: Likely dominated by odor, gill-size, spore-print color (classic strong features).Class Distribution: Balanced (edible vs poisonous).
Decision Patterns: A few attributes are enough for separation.
Overfitting Indicators: None → simple tree with perfect accuracy.

Nursery Dataset

Feature Importance: Attributes like parent satisfaction, safety, and children number dominate early splits.
Class Distribution: Imbalanced (some classes much larger).
Decision Patterns: Complex rules, large tree size.
Overfitting Indicators: Tree size (952 nodes) suggests risk of overfitting.

TicTacToe Dataset

Feature Importance: Center and corner positions dominate.
Class Distribution: Slight imbalance between X-win and O-win/no-win.
Decision Patterns: Patterns depend on board state → complex paths.
Overfitting Indicators: Depth=7 with moderate performance suggests partial overfitting.

---

4. Comparative Analysis Report

a) Algorithm Performance

Highest Accuracy: Mushroom (100%) → dataset has clean, separable features.
Dataset Size Impact: Larger datasets (Nursery, TicTacToe) create deeper and more complex trees, but not always better accuracy.
Number of Features: More features (Nursery) → tree grows huge, but accuracy is still high.
Simpler features (Mushroom) → shallow tree, perfect accuracy.

b) Data Characteristics Impact

Class Imbalance: Nursery and TicTacToe suffer slightly (precision/recall differences).

Feature Types: Binary features (TicTacToe board positions) lead to deeper, less accurate trees. Multi-valued categorical features (Mushroom, Nursery) capture more information quickly.

c) Practical Applications

Mushroom Dataset: Food safety → easily interpretable, real-world application in toxicology.
Nursery Dataset: Resource allocation for childcare → complex decisions, needs pruning or ensembles.
TicTacToe Dataset: Game AI → less practical but shows handling of strategic states.

d) Improvements

Apply tree pruning (reduces overfitting).
Use Random Forest / Ensemble methods for TicTacToe and Nursery.
Apply feature engineering to reduce complexity in Nursery dataset.

a) Algorithm Performance

a. Which dataset achieved the highest accuracy and why?

The Mushroom dataset achieved the highest accuracy (100%).
Reason: Its features are highly discriminative (e.g., odor, gill size, spore print color), making class boundaries very clear.
The classes (edible vs poisonous) are well-separated with minimal overlap, so even a shallow decision tree (depth=4) achieves perfect performance.

b. How does dataset size affect performance?

Mushroom: Moderate size, shallow tree, but still perfect accuracy → dataset simplicity matters more than size.

Nursery: Very large dataset, leading to a deep and very complex tree (952 nodes). High accuracy (98.67%) but near overfitting.
TicTacToe: Medium size dataset, but binary board positions cause ambiguity → accuracy only 87.3% despite tree depth 7.
Larger datasets tend to create deeper/more complex trees, but performance depends

more on data separability than size alone.

c. What role does the number of features play?

More features increase tree depth and node count, especially if multi-valued.
Mushroom (multi-valued features): Fewer splits needed, tree remains small.
Nursery (many features, multi-valued): Explodes in size (952 nodes).

TicTacToe (binary features): Needs more depth to capture combinations, but still struggles to classify perfectly.
Multi-valued categorical features often lead to better early splits than purely binary features.

b) Data Characteristics Impact

How does class imbalance affect tree construction?

In Nursery, some classes are dominant (e.g., "not recommended"), making the tree biased toward majority classes.
This leads to lower macro precision/recall (~0.76) despite high weighted accuracy.
In TicTacToe, imbalance between X-win, O-win, and draw states causes the tree to favor certain outcomes.

Which types of features work better (binary vs multi-valued)?

Multi-valued features (Mushroom, Nursery): Provide strong splits early, leading to higher accuracy.
Binary features (TicTacToe): Require more splits, create deeper trees, and result in lower accuracy.
Multi-valued categorical attributes are more efficient for decision tree construction.

c) Practical Applications

For which real-world scenarios is each dataset type most relevant?

Mushroom: Food safety & toxicology (classifying edible vs poisonous mushrooms). High accuracy & interpretability make it reliable for life-critical tasks.
Nursery: Resource allocation, social services, childcare recommendation systems.

Handles complex multi-criteria decision-making.
TicTacToe: Game strategy modeling → useful as a teaching dataset for AI/game theory, but less real-world critical.

What are the interpretability advantages for each domain?

Mushroom: Shallow tree (depth=4), very interpretable → rules can be directly explained (e.g., *if odor=bad → poisonous*).
Nursery: Tree is large, interpretability is lower, but still can highlight key factors (e.g.safety, parent satisfaction).
TicTacToe: Medium interpretability → paths represent board strategies, useful for teaching but less human-readable due to many binary states.

How would you improve performance for each dataset?

Mushroom: Already perfect; no major improvement needed.
Nursery: Apply pruning or Random Forests to reduce overfitting and simplify rules.
TicTacToe: Use ensemble methods (Bagging/Boosting) or feature engineering (grouping board positions) to improve accuracy beyond 87%.