ML LAB - 3

NAME- Chandan b
SRN- PES2UG23CS140
CLASS- 5C CSE

1)Mushroom dataset Output

```
1  2  3                                   Aug 20  11:12
~/Desktop/ML-Lab3
Internal Nodes:       272

~/Desktop/ML-Lab3 via v3.12.3
❯ python3 test.py --ID EC_5C_PES2UG23CS140_Lab3 --data mushroom.csv --print-tree
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', 'habitat', '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', 'habitat']
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...

🌳 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
1: ~/Desktop/Lab2 | 2: nvim . | 3: ~/Desktop/ML-Lab3
```

```
1  2  3                                Aug 20  11:13
~/Desktop/ML-Lab3
============================================================
Total samples: 8124
Training samples: 6499
Testing samples: 1625

Constructing decision tree using training data...

🌳 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
1: ~/Desktop/Lab2 | 2: nvim . | 3: ~/Desktop/ML-Lab3
```

```
1  2  3                                Aug 20  11:13
~/Desktop/ML-Lab3
│   │   ├── [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
│   └── = 8:
│       ├── Class 0
├── = 6:
│   ├── Class 1
├── = 7:
│   ├── Class 1
├── = 8:
│   ├── Class 1


📊 OVERALL PERFORMANCE METRICS
========================================
Accuracy:              1.0000 (100.00%)
Precision (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

🌳 TREE COMPLEXITY METRICS
========================================
Maximum Depth:         4
Total Nodes:           29
Leaf Nodes:            24
Internal Nodes:        5

~/Desktop/ML-Lab3 via 🐍 v3.12.3
❯ |
1: ~/Desktop/Lab2 | 2: nvim . | 3: ~/Desktop/ML-Lab3
```

2)Tictactoe Dataset output

```
1  2  3                                   Aug 20  11:15
🐱 ~/Desktop/ML-Lab3

~/Desktop/ML-Lab3 via 🐍 v3.12.3
❯ python3 test.py --ID EC_5C_PES2UG23CS140_Lab3 --data tictactoe.csv --print-tree
Running tests with PYTORCH framework
============================================================
 target column: 'Class' (last column)
Original dataset info:
Shape: (958, 10)
Columns: ['top-left-square', 'top-middle-square', 'top-right-square', 'middle-left-square', 'middle-middle-square', 'middle-right-square', 'bottom-left-square', 'bottom-middle-square', 'bott
om-right-square', 'Class']

First few rows:

top-left-square: ['x' 'o' 'b'] -> [2 1 0]

top-middle-square: ['x' 'o' 'b'] -> [2 1 0]

top-right-square: ['x' 'o' 'b'] -> [2 1 0]

Class: ['positive' 'negative'] -> [1 0]

Processed dataset shape: torch.Size([958, 10])
Number of features: 9
Features: ['top-left-square', 'top-middle-square', 'top-right-square', 'middle-left-square', 'middle-middle-square', 'middle-right-square', 'bottom-left-square', 'bottom-middle-square', 'bot
tom-right-square']
Target: Class
Framework: PYTORCH
Data type: <class 'torch.Tensor'>

============================================================
DECISION TREE CONSTRUCTION DEMO
============================================================
Total samples: 958
Training samples: 766
Testing samples: 192

Constructing decision tree using training data...

🌳 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:
1: ~/Desktop/Lab2 | 2: nvim . | 3: ~/Desktop/ML-Lab3
```

```
1  2  3                                   Aug 20  11:15
🐱 ~/Desktop/ML-Lab3
│   │       │   │       ├── 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)
│       │   ├── = 0:
│       │   │   ├── Class 1
│       │   ├── = 1:
│       │   │   ├── [bottom-right-square] (gain: 0.9403)
│       │   │   ├── = 0:
│       │   │   │   ├── Class 1
│       │   │   ├── = 1:
│       │   │   │   ├── Class 0
│       │   │   └── = 2:
│       │   │       ├── Class 1
│       │   └── = 2:
│       │       ├── [top-left-square] (gain: 0.9183)
│       │       ├── = 0:
│       │       │   ├── Class 1
│       │       ├── = 1:
│       │       │   ├── Class 0
│       │       └── = 2:
│       │           ├── Class 1
│       └── = 2:
│           ├── Class 1
├── = 1:
1: ~/Desktop/Lab2 | 2: nvim . | 3: ~/Desktop/ML-Lab3
```

```
1  2  3                                   Aug 20  11:16
🐱 ~/Desktop/ML-Lab3
│           │   ├── = 0:
│           │   │   ├── [bottom-middle-square] (gain: 0.8113)
│           │   │   ├── = 0:
│           │   │   │   ├── Class 1
│           │   │   ├── = 1:
│           │   │   │   ├── Class 0
│           │   │   └── = 2:
│           │   │       ├── Class 1
│           │   ├── = 1:
│           │   │   ├── Class 0
│           │   └── = 2:
│           │       ├── [top-middle-square] (gain: 0.8113)
│           │       ├── = 1:
│           │       │   ├── Class 0
│           │       └── = 2:
│           │           ├── Class 1
│           └── = 2:
│               ├── Class 1
├── = 2:
│   ├── [bottom-right-square] (gain: 0.0269)
│   ├── = 0:
│   │   ├── [top-left-square] (gain: 0.1239)
│   │   ├── = 0:
│   │   │   ├── Class 1
│   │   ├── = 1:
│   │   │   ├── [bottom-middle-square] (gain: 0.1033)
│   │   │   ├── = 0:
│   │   │   │   ├── [middle-left-square] (gain: 0.1605)
│   │   │   │   ├── = 0:
│   │   │   │   │   ├── Class 1
│   │   │   │   ├── = 1:
│   │   │   │   │   ├── [bottom-left-square] (gain: 1.0000)
│   │   │   │   │   ├── = 1:
│   │   │   │   │   │   ├── Class 0
│   │   │   │   │   └── = 2:
│   │   │   │   │       ├── Class 1
│   │   │   │   └── = 2:
│   │   │   │       ├── [middle-right-square] (gain: 0.5917)
│   │   │   │       ├── = 0:
│   │   │   │       │   ├── Class 0
│   │   │   │       ├── = 1:
│   │   │   │       │   ├── Class 1
│   │   │   │       └── = 2:
│   │   │   │           ├── Class 1
│   │   │   ├── = 1:
│   │   │   │   ├── Class 1
│   │   │   └── = 2:
│   │   │       ├── [top-middle-square] (gain: 0.4591)
│   │   │       ├── = 0:
│   │   │       │   ├── [middle-right-square] (gain: 0.9183)
│   │   │       │   ├── = 0:
│   │   │       │   │   ├── Class 0
1: ~/Desktop/Lab2 | 2: nvim . | 3: ~/Desktop/ML-Lab3
```

```
1  2  3                                   Aug 20  11:17
🐱 ~/Desktop/ML-Lab3
│       │   │       │   ├── [top-right-square] (gain: 0.9183)
│       │   │       │   ├── = 1:
│       │   │       │   │   ├── Class 1
│       │   │       │   └── = 2:
│       │   │       │       ├── Class 0
│       │   │       └── = 2:
│       │   │           ├── [top-right-square] (gain: 0.9183)
│       │   │           ├── = 1:
│       │   │           │   ├── Class 0
│       │   │           └── = 2:
│       │   │               ├── Class 1
│       │   └── = 2:
│       │       ├── [middle-left-square] (gain: 0.3425)
│       │       ├── = 0:
│       │       │   ├── [top-right-square] (gain: 0.9710)
│       │       │   ├── = 1:
│       │       │   │   ├── Class 0
│       │       │   └── = 2:
│       │       │       ├── Class 1
│       │       ├── = 1:
│       │       │   ├── [top-right-square] (gain: 0.9183)
│       │       │   ├── = 0:
│       │       │   │   ├── Class 0
│       │       │   ├── = 1:
│       │       │   │   ├── Class 0
│       │       │   └── = 2:
│       │       │       ├── Class 1
│       │       └── = 2:
│       │           ├── Class 1
│       └── = 2:
│           ├── Class 1

📊 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

~/Desktop/ML-Lab3 via 🐍 v3.12.3
❯ |
1: ~/Desktop/Lab2 | 2: nvim . | 3: ~/Desktop/ML-Lab3
```

3) Nursery dataset output

```
1  2  3                                Aug 20  11:08
~/Desktop/ML-Lab3
Leaf Nodes:          703
Internal Nodes:      280

~/Desktop/ML-Lab3 via v3.12.3
❯ python3 test.py --ID EC_5C_PES2UG23CS140_Lab3 --data Nursery.csv --print-tree

Running tests with PYTORCH framework
============================================================
 target column: 'class' (last column)
Original dataset info:
Shape: (12960, 9)
Columns: ['parents', 'has_nurs', 'form', 'children', 'housing', 'finance', 'social', 'health', 'class']

First few rows:

parents: ['usual' 'pretentious' 'great_pret'] -> [2 1 0]

has_nurs: ['proper' 'less_proper' 'improper' 'critical' 'very_crit'] -> [3 2 1 0 4]

form: ['complete' 'completed' 'incomplete' 'foster'] -> [0 1 3 2]

class: ['recommend' 'priority' 'not_recom' 'very_recom' 'spec_prior'] -> [2 1 0 4 3]

Processed dataset shape: torch.Size([12960, 9])
Number of features: 8
Features: ['parents', 'has_nurs', 'form', 'children', 'housing', 'finance', 'social', 'health']
Target: class
Framework: PYTORCH
Data type: <class 'torch.Tensor'>

============================================================
DECISION TREE CONSTRUCTION DEMO
============================================================
Total samples: 12960
Training samples: 10368
Testing samples: 2592

Constructing decision tree using training data...

🌳 Decision tree construction completed using PYTORCH!

🌲 DECISION TREE STRUCTURE
============================================================
Root [health] (gain: 0.9595)
├── = 0:
│   ├── Class 0
├── = 1:
│   ├── [has_nurs] (gain: 0.3555)
│   ├── = 0:
│   │   ├── [parents] (gain: 0.1673)
│   │   ├── = 0:
│   │   │   ├── [form] (gain: 0.0171)
1: ~/Desktop/Lab2 | 2: nvim . | 3: ~/Desktop/ML-Lab3
```

```
1  2  3                                Aug 20  11:08
~/Desktop/ML-Lab3
│       │       │   │   ├── 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:
1: ~/Desktop/Lab2 | 2: nvim . | 3: ~/Desktop/ML-Lab3
```

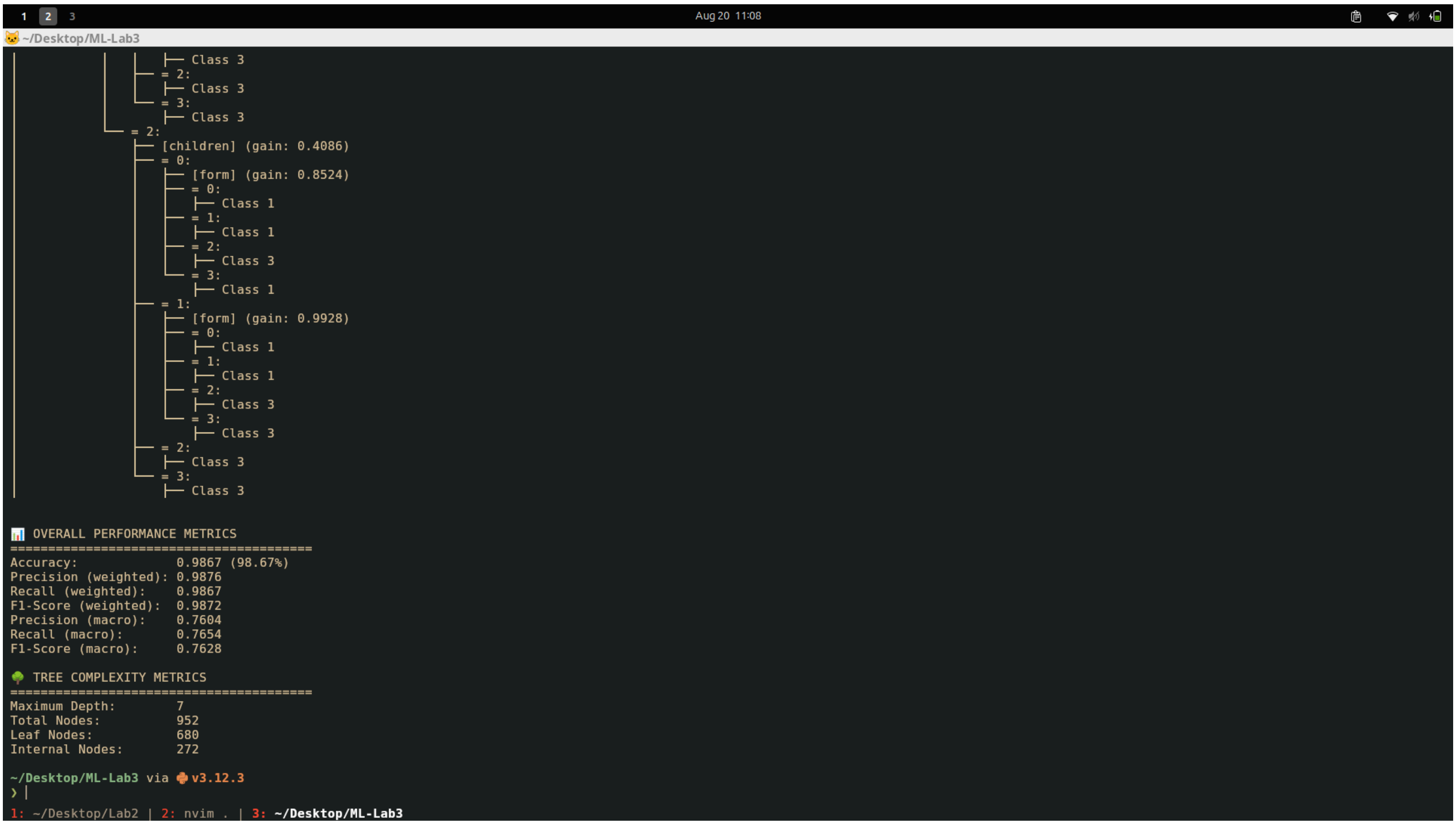
Screenshot Just now
Screenshot captured
You can paste the image from the clipboard.

```
1 2 3                                Aug 20 11:08
~/Desktop/ML-Lab3
│       │   │   │   │   ├── 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
│           ├── = 1:
│           │   ├── [children] (gain: 0.1481)
│           │   ├── = 0:
1: ~/Desktop/Lab2 | 2: nvim . | 3: ~/Desktop/ML-Lab3
```

```
Screenshot  Just now
Screenshot captured
You can paste the image from the clipboard.
```

```
1 2 3                                Aug 20 11:08
~/Desktop/ML-Lab3
│   │       ├── [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:
1: ~/Desktop/Lab2 | 2: nvim . | 3: ~/Desktop/ML-Lab3
```


```
Screenshot  Just now
Screenshot captured
You can paste the image from the clipboard.
```



```
1 2 3                    Aug 20 11:08
~/Desktop/ML-Lab3
│           │   │   ├── Class 3
│           │   ├── = 2:
│           │   │   ├── Class 3
│           │   └── = 3:
│           │       ├── Class 3
│           └── = 2:
│               ├── [children] (gain: 0.4086)
│               ├── = 0:
│               │   ├── [form] (gain: 0.8524)
│               │   ├── = 0:
│               │   │   ├── Class 1
│               │   ├── = 1:
│               │   │   ├── Class 1
│               │   ├── = 2:
│               │   │   ├── Class 3
│               │   └── = 3:
│               │       ├── Class 1
│               ├── = 1:
│               │   ├── [form] (gain: 0.9928)
│               │   ├── = 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

~/Desktop/ML-Lab3 via 🐍 v3.12.3
❯ |
1: ~/Desktop/Lab2 | 2: nvim . | 3: ~/Desktop/ML-Lab3
```

Q1)Performance Comparision

Overall Performance Metrics Comparison

Dataset	Accuracy	Precision (weighted)	Recall (weighted)	F1-Score (weighted)	Precision (macro)	Recall (macro)	F1-Score (macro)
Mushroom	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Tic-Tac-Toe	87.3%	87.4%	87.3%	87.3%	85.9%	86.4%	86.1%
Nursery	98.7%	98.7%	98.7%	98.7%	90.6%	76.3%	82.8%

Key Performance Insights

1. Algorithm Performance Ranking

1. Mushroom Dataset (Perfect Performance)
    - 100% accuracy across all metrics
    - Most efficient with simplest tree structure (depth=4)
    - Clear, discriminative features enable perfect classification
2. Nursery Dataset (High Weighted Performance)
    - 98.7% weighted accuracy but lower macro performance
    - Suffers from class imbalance (macro recall only 76.3%)
    - Most complex tree structure (652 total nodes)
3. Tic-Tac-Toe Dataset (Balanced Performance)
    - 87.3% accuracy with consistent metrics
    - Moderate tree complexity (281 nodes)

Q2)

Tree Complexity Metrics Comparison

Dataset	Max Depth	Total Nodes	Leaf Nodes	Internal Nodes	Tree Complexity
Mushroom	4	29	25	4	Simple
Tic-Tac-Toe	7	281	180	101	Moderate
Nursery	7	652	380	272	Complex

Q3)Dataset specific insights

Aspect	Mushroom	Tic-Tac-Toe	Nursery
Feature Type	Biological	Spatial (Game Board)	Socio-economic
Class Balance	Perfect	Moderate	Severe imbalance
Tree Simplicity	Very Simple	Moderate	Very Complex
Overfitting Risk	None	Low	High
Decision Clarity	Clear rules	Strategic patterns	Complex policies

Q4)Comparitive analysis
A

Q) Which dataset achieved the highest accuracy and why?

- **Mushroom Dataset achieved the highest accuracy (100%) across all metrics.**
- **Reason: The mushroom dataset consists of highly discriminative categorical features (e.g., odor, spore-print-color) which perfectly separate the two target classes. The class distribution is balanced, enabling clear splits and minimizing ambiguity or noise**

Q) How does dataset size affect performance?

- **Larger datasets (Mushroom, Nursery) generally enable better generalization and more robust decision boundaries by providing sufficient examples for all cases.**
- **However, size alone does not guarantee performance: Despite the Nursery dataset's size, high class imbalance and more complex multi-class targets prevent perfect classification.**
- **Moderate-sized datasets (Tic-Tac-Toe) can achieve good accuracy if feature representation and class balance are reasonable, but may require deeper trees for complex decision patterns.**

Q)What role does the number of features play?

- **More features can increase model expressiveness, enabling finer splits and potentially higher accuracy, as in Mushroom (22 features, shallow tree, perfect separability).**
- **However, benefit depends on discriminatory power: Nursery has fewer features (8), but the multi-class structure and imbalances require deeper and more complex trees, suggesting that effectiveness of features—rather than sheer count—is more important.**
- **Irrelevant or redundant features can lead to overfitting and unnecessary tree growth without improving performance.**

B Data Characteristics Impact

Q)How does class imbalance affect tree construction?

- **Class imbalance causes the tree to optimize for the majority class, resulting in:**
  - **Higher weighted accuracy, but lower macro metrics (Nursery: Weighted F1 ~99%, Macro F1 ~83%)**
  - **Complex, deep tree branches specializing for minority classes, increasing risk of overfitting**
  - **Poor generalization on underrepresented classes, as seen by the gap between weighted and macro recalls in the Nursery dataset**

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

- **Binary Features: Typically facilitate simpler splits and more interpretable trees. Mushroom and Tic-Tac-Toe (mostly binary/ternary categorical features) achieve high performance and reasonable tree complexity.**
- **Multi-valued Features: Can increase tree depth and branching, leading to higher complexity. If informative (as in Mushroom), they aid classification; if not, they introduce overfitting risks and complexity (as seen in Nursery).**

C)How would you improve performance for each dataset?

- **Mushroom: Already optimal; no improvements necessary.**
- **Tic-Tac-Toe:**
  - **Prune the tree to reduce over-complexity and improve generalizability.**
  - **Feature engineering to encode strategic patterns (e.g., imminent win conditions).**
- **Nursery:**
  - **Address class imbalance with resampling (SMOTE, oversampling minority classes).**
  - **Apply cost-sensitive learning to penalize misclassification of minority classes.**
  - **Use feature selection or dimensionality reduction to lower tree complexity.**
  - **Consider ensemble methods (e.g., boosting) for better minority class prediction.**