

ML Lab 3

Name: Cheruku Manas Ram

SRN: PES2UG23CS147

Date: 19/08/2025

Q. Compare the following metrics across all three datasets:

- Accuracy: Overall classification accuracy
- Precision: True positives / (True positives + False positives)
- Recall: True positives / (True positives + False negatives)
- F1-Score: Harmonic mean of precision and recall

A.

Metric		Mushroom	Nursery	Tic-Tac-Toe
Accuracy	Weighted	1	0.9887	0.8836
Precision	Weighted	1	0.9888	0.8827
	Macro	1	0.9577	0.8784
Recall	Weighted	1	0.9887	0.8836
	Macro	1	0.9576	0.8600
F1-Score	Weighted	1	0.9887	0.8822
	Macro	1	0.9576	0.8680

Mushroom dataset:

The model correctly classified all mushrooms in the test set. There were zero false positives and false negatives since precision and recall are “1”. Hence, F1-score is also “1”.

Nursery dataset:

The model was correct on 98.9% (approx) of its predictions. False positives and false negatives are very rare since precision and recall are very high (nearly 99%). The high F1-score also indicates that the model is strong in both precision and recall.

Tic-tac-toe dataset:

The model correctly predicts the output about 88.4% (approx) of the time. The model incorrectly predicted about 11.7% of the time (False positives derived from precision) and the model failed to identify a win for “X” about 11.6% of the time. (False negatives derived from recall). The F1-score indicates a decent balance between precision and recall.

Q. Tree Characteristics Analysis

Analyze and compare:

- Tree Depth: Maximum depth of the constructed trees
- Number of Nodes: Total nodes in each tree
- Tree Complexity: Relationship between tree size and dataset characteristics

A.

Characteristic	Mushroom	Nursery	Tic-Tac-Toe
Max Depth	4	7	7
Total Nodes	29	983	260

Root feature	odor	health	middle-middle-square
--------------	------	--------	----------------------

The mushroom dataset is very shallow. This means that a mushroom can be classified with at most 4 questions. Whereas the other two datasets are deeper, indicating that they might require a few more questions (at most 7 questions) to come to a conclusion.

mushroom uses odor as the root node and then splits into spore-print-color and habitat.

Nursery uses health as the root node.

tic tac toe uses middle-middle-square as the root node.

Mushroom is very easy to interpret since it has only 29 nodes. Tic tac toe is moderate with 260 nodes, whereas nursery has almost a 1000 nodes which makes the tree very complex with a lot of combinations.

The complexity is proportional to the tree size. Therefore, complexity = Mushroom < tic tac toe < Nursery.

Q. Dataset-Specific Insights

A. Mushroom dataset:

- a. Odor is the most important attribute. The class distribution is very balanced (since entropy is very high). Therefore, no rare classes. There are no signs of overfitting. It is a simple tree with a depth of only 4.

B. Nursery:

- a. There are almost a 1000 attributes with health being on the root of the tree. The decision paths are long, involving at most 7 conditions to arrive at the conclusion. There might be a potential overfitting, since there are almost a 1000 nodes, the model probably learns a lot of patterns specific to the training data that may not generalize well. Pruning might need to be done.

C. Tic tac toe

- a. The middle-middle-square is taken as the root node. There are a little over 200 nodes and has a depth of 7. This dataset however has an accuracy of less than 90% indicating that more training might need to be done.

• Which dataset achieved the highest accuracy and why?

Mushroom, the feature odor is strongly correlated with mushroom's class and can classify most samples correctly, indicating that the classes are easily separable (poisonous vs edible).

• How does dataset size affect performance?

Larger the dataset, better the performance. But, data quality is better than size, as seen in mushroom compared to nursery which is a larger dataset. Also, the tree might be too deep, reducing performance with a larger dataset.

• What role does the number of features play?

Too many multi-valued features make it prone to overfitting.

• How does class imbalance affect tree construction?

Makes the tree biased towards majority class.

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

Binary since it produces simpler splits and shallow trees, on the other hand, multi-valued makes more complex trees that might also increase the risk of overfitting.

- For which real-world scenarios is each dataset type most relevant?
 - Mushroom food safety.
 - Strategy learning
 - Nursery admissions
- What are the interpretability advantages for each domain?
 - Mushroom edibility with high clarity. Accountability in nursery admissions. Outcomes by board positions.
- How would you improve performance for each dataset?
 - Mushrooms -> Pruning
 - Nursery -> Pruning, resampling
 - Tic tac toe -> Feature engineering and pruning

Output Screenshots

Mushroom

```

(MLVenv) manas@vivoBook:~/Desktop/ML/ML_C_PES2UG23CS147_CherukuManasRao$ python test.py --ID EC_C_PES2UG23CS147_Lab3 --data mushrooms.csv --framework sklearn
Running tests with SKLEARN 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: (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: SKLEARN
Data type: <class 'numpy.ndarray'>

=====
DECISION TREE CONSTRUCTION DEMO
=====
Total samples: 8124
Training samples: 6499
Testing samples: 1625
Constructing decision tree using training data...
🟢 Decision tree construction completed using SKLEARN!

📊 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

```



DECISION TREE STRUCTURE

```
=====
Root [odor] (gain: 0.9048)
├── = 0:
│   └── Class 0
├── = 1:
│   └── Class 1
├── = 2:
│   └── Class 1
├── = 3:
│   └── Class 0
├── = 4:
│   └── Class 1
├── = 5:
│   ├── [spore-print-color] (gain: 0.1487)
│   ├── = 0:
│   │   └── Class 0
│   ├── = 1:
│   │   └── Class 0
│   ├── = 2:
│   │   └── Class 0
│   ├── = 3:
│   │   └── Class 0
│   ├── = 4:
│   │   └── Class 0
│   ├── = 5:
│   │   └── Class 1
│   └── = 7:
│       ├── [habitat] (gain: 0.2767)
│       ├── = 0:
│       │   ├── [gill-size] (gain: 0.6374)
│       │   ├── = 0:
│       │   │   └── Class 0
│       │   └── = 1:
│       │       └── Class 1
│       ├── = 1:
│       │   └── Class 0
│       ├── = 2:
│       │   ├── [cap-color] (gain: 0.8267)
│       │   ├── = 1:
│       │   │   └── Class 0
│       │   ├── = 4:
│       │   │   └── Class 0
│       │   ├── = 8:
│       │   │   └── Class 1
│       │   └── = 9:
│       │       └── Class 1
│       ├── = 4:
│       │   └── Class 0
│       ├── = 6:
│       │   └── Class 0
│       └── = 8:
│           └── Class 0
└── = 6:
    └── Class 1
```

```
Constructing decision tree using training data...
Level 0: Node Info - Entropy = 0.9992
Level 0: Node Info - Selected Attribute: odor (gain: 0.9048)
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 = 0.2186
Level 1: Node Info - | Selected Attribute: spore-print-color (gain: 0.1487)
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 = 0.4019
Level 2: Node Info - | | Selected Attribute: habitat (gain: 0.2767)
Level 2: Node Info - | | Branch habitat = 0
Level 3: Node Info - | | | Entropy = 0.6374
Level 3: Node Info - | | | Selected Attribute: gill-size (gain: 0.6374)
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
```

Nursery

```
(MLenv) manas@vivobook:~/Desktop/ML/ML_C_PES2UG23CS147_CherukuManasRam$ python test.py --ID EC_C_PES2UG23CS147_Lab3 --data Nursery.csv --framework sklearn
Running tests with SKLEARN 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: (12960, 9)
Number of features: 8
Features: ['parents', 'has_nurs', 'form', 'children', 'housing', 'finance', 'social', 'health']
Target: class
Framework: SKLEARN
Data type: <class 'numpy.ndarray'>

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

Constructing decision tree using training data...

🟢 Decision tree construction completed using SKLEARN!

📊 OVERALL PERFORMANCE METRICS
=====
Accuracy:      0.9887 (98.87%)
Precision (weighted): 0.9888
Recall (weighted):  0.9887
F1-Score (weighted): 0.9887
Precision (macro):  0.9877
Recall (macro):     0.9576
F1-Score (macro):   0.9576

🌳 TREE COMPLEXITY METRICS
=====
Maximum Depth: 7
Total Nodes: 983
Leaf Nodes: 703
Internal Nodes: 280
```

DECISION TREE STRUCTURE

```
=====
Root [health] (gain: 0.9597)
├── = 0:
│   └── Class 0
├── = 1:
│   ├── [has_nurs] (gain: 0.3648)
│   └── = 0:
│       ├── [parents] (gain: 0.1629)
│       └── = 0:
│           ├── [children] (gain: 0.0288)
│           └── = 0:
│               ├── [form] (gain: 0.1164)
│               └── = 0:
│                   ├── [housing] (gain: 0.4028)
│                   └── = 0:
│                       ├── [finance] (gain: 0.9710)
│                       └── = 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
│       └── = 1:
│           ├── [form] (gain: 0.0277)
│           └── = 0:
│               ├── [children] (gain: 0.1152)
│               └── = 0:
│                   ├── [housing] (gain: 0.4028)
│                   └── = 0:
│                       ├── [finance] (gain: 0.9710)
│                       └── = 0:
│                           ├── Class 1
│                           └── = 1:
│                               └── Class 3
```

```

Constructing decision tree using training data...
Level 0: Node Info - Entropy = 1.7226
Level 0: Node Info - Selected Attribute: health (gain: 0.9597)
Level 0: Node Info - Branch health = 0
Level 1: Node Info - | Entropy = -0.0000
Level 1: Node Info - | Hypothesis: Class 0
Level 0: Node Info - Branch health = 1
Level 1: Node Info - | Entropy = 0.9864
Level 1: Node Info - | Selected Attribute: has_nurs (gain: 0.3648)
Level 1: Node Info - | Branch has_nurs = 0
Level 2: Node Info - | | Entropy = 0.5436
Level 2: Node Info - | | Selected Attribute: parents (gain: 0.1629)
Level 2: Node Info - | | Branch parents = 0
Level 3: Node Info - | | | Entropy = 0.1037
Level 3: Node Info - | | | Selected Attribute: children (gain: 0.0288)
Level 3: Node Info - | | | Branch children = 0
Level 4: Node Info - | | | | Entropy = 0.3182
Level 4: Node Info - | | | | Selected Attribute: form (gain: 0.1164)
Level 4: Node Info - | | | | Branch form = 0
Level 5: Node Info - | | | | | Entropy = 0.7496
Level 5: Node Info - | | | | | Selected Attribute: housing (gain: 0.4028)
Level 5: Node Info - | | | | | Branch housing = 0
Level 6: Node Info - | | | | | | Entropy = 0.9710
Level 6: Node Info - | | | | | | Selected Attribute: finance (gain: 0.9710)
Level 6: Node Info - | | | | | | Branch finance = 0
Level 7: Node Info - | | | | | | | Entropy = -0.0000
Level 7: Node Info - | | | | | | | Hypothesis: Class 1
Level 6: Node Info - | | | | | | | Branch finance = 1
Level 7: Node Info - | | | | | | | | Entropy = -0.0000
Level 7: Node Info - | | | | | | | | Hypothesis: Class 3
Level 5: Node Info - | | | | | | | Branch housing = 1
Level 6: Node Info - | | | | | | | | Entropy = -0.0000
Level 6: Node Info - | | | | | | | | Hypothesis: Class 3
Level 5: Node Info - | | | | | | | | Branch housing = 2
Level 6: Node Info - | | | | | | | | | Entropy = -0.0000
Level 6: Node Info - | | | | | | | | | Hypothesis: Class 3
Level 4: Node Info - | | | | | | | | Branch form = 1
Level 5: Node Info - | | | | | | | | | Entropy = -0.0000
Level 5: Node Info - | | | | | | | | | Hypothesis: Class 3
Level 4: Node Info - | | | | | | | | | Branch form = 2
Level 5: Node Info - | | | | | | | | | | Entropy = -0.0000
Level 5: Node Info - | | | | | | | | | | Hypothesis: Class 3
Level 4: Node Info - | | | | | | | | | | Branch form = 3
Level 5: Node Info - | | | | | | | | | | | Entropy = -0.0000
Level 5: Node Info - | | | | | | | | | | | Hypothesis: Class 3
Level 3: Node Info - | | | | | | | | | | Branch children = 1
Level 4: Node Info - | | | | | | | | | | | Entropy = -0.0000
Level 4: Node Info - | | | | | | | | | | | Hypothesis: Class 3
Level 3: Node Info - | | | | | | | | | | | Branch children = 2
Level 4: Node Info - | | | | | | | | | | | | Entropy = -0.0000
Level 4: Node Info - | | | | | | | | | | | | Hypothesis: Class 3

```

Tic Tac Toe


```
(MLenv) manas@vivo:~/Desktop/ML_EC_PES2UG23CS147_CherryManasRao$ python test.py --ID_EC_PES2UG23CS147_Lab3 --data tictactoe.csv --framework sklearn
Running tests with SKLEARN 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', 'bottom-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: (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', 'bottom-right-square']
Target: class
Framework: SKLEARN
Data type: <class 'numpy.ndarray'>
=====
DECISION TREE CONSTRUCTION DEMO
=====
Total samples: 958
Training samples: 766
Testing samples: 192

Constructing decision tree using training data...

🌱 Decision tree construction completed using SKLEARN!

📊 OVERALL PERFORMANCE METRICS
=====
Accuracy: 0.8836 (88.36%)
Precision (weighted): 0.8827
Recall (weighted): 0.8836
F1-Score (weighted): 0.8822
Precision (macro): 0.8784
Recall (macro): 0.8680
F1-Score (macro): 0.8680

🌱 TREE COMPLEXITY METRICS
=====
Maximum Depth: 7
Total Nodes: 260
Leaf Nodes: 165
Internal Nodes: 95
```



DECISION TREE STRUCTURE

```
=====
Root [middle-middle-square] (gain: 0.0909)
├── = 0:
│   ├── [bottom-left-square] (gain: 0.0922)
│   │   └── = 0:
│   │       ├── [top-right-square] (gain: 0.8281)
│   │       │   ├── = 1:
│   │       │   │   └── Class 0
│   │       │   └── = 2:
│   │       │       └── Class 1
│   │   └── = 1:
│   │       ├── [top-right-square] (gain: 0.3120)
│   │       │   ├── = 0:
│   │       │   │   └── Class 0
│   │       │   ├── = 1:
│   │       │   │   └── Class 0
│   │       │   └── = 2:
│   │       │       ├── [bottom-right-square] (gain: 0.1399)
│   │       │       │   ├── = 0:
│   │       │       │   │   ├── [top-left-square] (gain: 0.9183)
│   │       │       │   │   │   ├── = 1:
│   │       │       │   │   │   │   └── Class 0
│   │       │       │   │   │   └── = 2:
│   │       │       │   │   │       └── Class 1
│   │       │       │   └── = 1:
│   │       │       │       ├── [bottom-middle-square] (gain: 0.6952)
│   │       │       │       │   ├── = 0:
│   │       │       │       │   │   └── Class 1
│   │       │       │       │   ├── = 1:
│   │       │       │       │   │   └── Class 0
│   │       │       │       │   └── = 2:
│   │       │       │       │       ├── [top-left-square] (gain: 0.9183)
│   │       │       │       │       │   ├── = 1:
│   │       │       │       │       │   │   └── Class 0
│   │       │       │       │       │   └── = 2:
│   │       │       │       │       │       └── Class 1
│   │       │       │       └── = 2:
│   │       │       │           ├── [middle-right-square] (gain: 0.4834)
│   │       │       │           │   ├── = 0:
│   │       │       │           │   │   ├── [top-left-square] (gain: 1.0000)
│   │       │       │           │   │   │   ├── = 1:
│   │       │       │           │   │   │   │   └── Class 0
│   │       │       │           │   │   │   └── = 2:
│   │       │       │           │   │   │       └── Class 1
│   │       │       │           │   └── = 1:
│   │       │       │           │       └── Class 0
│   │       │       │           └── = 2:
│   │       │       │               └── Class 1
│   │       └── = 2:
│   │           └── [top-right-square] (gain: 0.1815)
│   │               ├── = 0:
│   │               │   └── Class 1
```

```

Constructing decision tree using training data...
Level 0: Node Info - Entropy = 0.9316
Level 0: Node Info - Selected Attribute: middle-middle-square (gain: 0.0909)
Level 0: Node Info - Branch middle-middle-square = 0
Level 1: Node Info - | Entropy = 0.9020
Level 1: Node Info - | Selected Attribute: bottom-left-square (gain: 0.0922)
Level 1: Node Info - | Branch bottom-left-square = 0
Level 2: Node Info - | | Entropy = 0.8281
Level 2: Node Info - | | Selected Attribute: top-right-square (gain: 0.8281)
Level 2: Node Info - | | Branch top-right-square = 1
Level 3: Node Info - | | | Entropy = -0.0000
Level 3: Node Info - | | | Hypothesis: Class 0
Level 2: Node Info - | | Branch top-right-square = 2
Level 3: Node Info - | | | Entropy = -0.0000
Level 3: Node Info - | | | Hypothesis: Class 1
Level 1: Node Info - | Branch bottom-left-square = 1
Level 2: Node Info - | | Entropy = 0.9892
Level 2: Node Info - | | Selected Attribute: top-right-square (gain: 0.3120)
Level 2: Node Info - | | Branch top-right-square = 0
Level 3: Node Info - | | | Entropy = -0.0000
Level 3: Node Info - | | | Hypothesis: Class 0
Level 2: Node Info - | | Branch top-right-square = 1
Level 3: Node Info - | | | Entropy = -0.0000
Level 3: Node Info - | | | Hypothesis: Class 0
Level 2: Node Info - | | Branch top-right-square = 2
Level 3: Node Info - | | | Entropy = 0.9576
Level 3: Node Info - | | | Selected Attribute: bottom-right-square (gain: 0.1399)
Level 3: Node Info - | | | Branch bottom-right-square = 0
Level 4: Node Info - | | | | Entropy = 0.9183
Level 4: Node Info - | | | | Selected Attribute: top-left-square (gain: 0.9183)
Level 4: Node Info - | | | | Branch top-left-square = 1
Level 5: Node Info - | | | | | Entropy = -0.0000
Level 5: Node Info - | | | | | Hypothesis: Class 0
Level 4: Node Info - | | | | Branch top-left-square = 2
Level 5: Node Info - | | | | | Entropy = -0.0000
Level 5: Node Info - | | | | | Hypothesis: Class 1
Level 3: Node Info - | | Branch bottom-right-square = 1
Level 4: Node Info - | | | Entropy = 0.9457
Level 4: Node Info - | | | Selected Attribute: bottom-middle-square (gain: 0.6952)
Level 4: Node Info - | | | Branch bottom-middle-square = 0
Level 5: Node Info - | | | | Entropy = -0.0000
Level 5: Node Info - | | | | Hypothesis: Class 1
Level 4: Node Info - | | | Branch bottom-middle-square = 1
Level 5: Node Info - | | | | Entropy = -0.0000
Level 5: Node Info - | | | | Hypothesis: Class 0
Level 4: Node Info - | | | Branch bottom-middle-square = 2
Level 5: Node Info - | | | | Entropy = 0.9183
Level 5: Node Info - | | | | Selected Attribute: top-left-square (gain: 0.9183)
Level 5: Node Info - | | | | Branch top-left-square = 1
Level 6: Node Info - | | | | | Entropy = -0.0000
Level 6: Node Info - | | | | | Hypothesis: Class 0
Level 5: Node Info - | | | | Branch top-left-square = 2
Level 6: Node Info - | | | | | Entropy = -0.0000
Level 6: Node Info - | | | | | Hypothesis: Class 1
Level 3: Node Info - | | Branch bottom-right-square = 2

```