PES UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MACHINE LEARNING

LAB 3

NAME : CHINMAYI SHRAVANI YELLANKI
SRN : PES2UG23CS152

CLASS AND SEC : SEM 5 - C
DATE : 20.08.2025

OUTPUT SCREENSHOTS:

1.python test.py --ID EC_C_PES2UG23CS152_Lab3 --data mushroom.csv

```
🌳 Decision tree construction completed using PYTORCH!

📊 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
```

2. python test.py --ID EC_C_PES2UG23CS152_Lab3 --data tictactoe.csv

```
🌳 Decision tree construction completed using PYTORCH!

📊 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
```

3. python test.py --ID EC_C_PES2UG23CS152_Lab3 --data nursery.csv

```
🌳 Decision tree construction completed using PYTORCH!

📊 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
```

Algorithm Performance:

1. Which dataset achieved the highest accuracy and why?

Ans: Mushroom dataset (Dataset 1) had the best accuracy, at 1.0000 (100%).
Reason:
The mushroom data has categorical attributes that are highly indicative of the target ("edible"/"poisonous"). There are deterministic, observable relationships between feature sets and class, allowing the decision tree to distinguish classes perfectly with a small tree depth (maximum depth: 4, few nodes). There is minimal overlap or noise in feature values in terms of edibility, thus the perfect scores.

2. How does dataset size affect performance?

Ans:
Mushroom: 29 nodes, 24 leaf nodes, depth 4, 100% accuracy.
Tic-Tac-Toe: 281 nodes, 180 leaf nodes, depth 7, 87.3% accuracy.
Nursery: 952 nodes, 680 leaf nodes, depth 7, 98.67% accuracy.
Interpretation:
With increasing dataset size and number of classes/features, tree complexity increases (depth and nodes), but accuracy might not increase proportionately. Noise and overlapping values introduced by large datasets make perfect classification more difficult, such as with Tic-Tac-Toe. But more data gives more scope for generalization. If the data is clean and features are descriptive, as in Nursery, precision can remain high

3. What role does the number of features play?

Ans:
Mushroom: Numerous categorical features.
Tic-Tac-Toe: 9 board positions, all multi-valued.
Nursery: 8 family/social features, all categorical.
Impact:
More features translate into more axes for splits, but only those features that are informative are useful; non-informative features can make the tree larger without improving accuracy.
In Mushrooms dataset, features are very discriminative, thus the tree is compact but efficient.
In Tic-Tac-Toe dataset, board positions introduce complexity but also redundant/irrelevant
information (board symmetries, irrelevant to outcome).
In Nursery dataset, combination complexity of features demands a large and deep tree but discrimination is robust enough to preserve high accuracy

Data Characteristics Impact

1. How does class imbalance affect tree construction?

Ans: Class imbalance in a data set leads the decision tree to bias towards the majority class, which tends to create additional splits and leaves for the majority class. This makes the recall and precision decrease for minority classes and even results in a less equitable or less generalizable model, particularly when tested on imbalanced real-world data sets.

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

Ans: For feature kinds, binary features (such as "yes"/"no" or two-class features) tend to produce simpler, shallower, and easier-to-explain trees. Multi-valued features, though potent and versatile in being able to model intricate structures, can grow the tree very quickly, making it deeper and more difficult to understand, and in some cases grow the risk of overfitting.

Practical Applications

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

Ans:
Mushroom dataset: Most appropriate for agriculture and food safety, helping individuals or systems decide with absolute certainty based on physical characteristics observable whether a mushroom is edible or not.
Tic-Tac-Toe dataset: Most appropriate for game AI coding, strategy analysis, and machine learning-based predictions of game outcomes from board positions. It is helpful in teaching software or in reinforcement learning research.
Nursery dataset: Relevant to education policy, school admissions, and social work. It permits recommendation systems that consider family and social factors when ranking applications.

2. What are the interpretability advantages for each domain?

Ans:
Mushroom: Shallow tree with basic and explicit rules provides high interpretability. This is valuable in safety-critical settings where the foundation of a classification must be made clear.
Tic-Tac-Toe: Interpretability is relatively moderate. A tree can be used to represent game strategies, yet the more complex the board configurations, the higher the complexity and some decision paths become harder for humans to follow entirely.
Nursery: Lower interpretability. The tree is very big and the decisions are made based on lots of factors, so it is difficult to get simple explanations. Nevertheless, the tree is able to identify the most influential drivers in complicated real-world situations.
.

3. How would you improveperformanceforeachdataset?

Ans:
Mushroom: Ideal accuracy so no major enhancements are required. As additional confirmation, cross-validation or introducing noise can be used for robustness testing.
Tic-Tac-Toe: Prune the tree to minimize overfitting. Solve class imbalance if it occurs, and think about encoding board symmetries to make the model easier and more generalizable.
Nursery: Use feature selection to eliminate unimportant variables. Starting with rare target classes if feasible, thin out the tree to make it easier to interpret, or employ ensemble methods like random forests to make more intelligent predictions and less overfitting.