ML LAB WEEK 3 - ANALYSIS REPORT

1. Performance comparison :
   Mushroom dataset -

- Accuracy: 100%
- Precision: 1.0000
- Recall: 1.0000
- F1-Score: 1.0000

Nursery dataset -

- Accuracy: 98.87%
- Precision (weighted): 0.9888
- Recall (weighted): 0.9887
- F1-Score (weighted): 0.9887

TicTacToe dataset -

- Accuracy: 88.36%
- Precision (weighted): 0.8827
- Recall (weighted): 0.8836
- F1-Score (weighted): 0.8822

The decision tree models showed the best performance on the Mushroom dataset, achieving perfect accuracy, precision, recall, and F1-score, indicating clear feature-label relationships and no ambiguity in classification. The Nursery dataset also performed very well with near-perfect metrics (~99%), though its deeper and more complex tree suggests more variability in features and class labels. In contrast, the TicTacToe dataset had comparatively lower performance (~88%), reflecting the higher complexity and less deterministic nature of game states, where patterns are harder to separate with simple splits. Overall, dataset characteristics such as feature clarity, balance, and noise strongly influenced performance.

2. Tree character analysis :

Mushroom dataset -

- Depth: 4
- Nodes: 29
- Leaf nodes: 24
- Early splits: Odor, spore-print-color dominate.
- Complexity: Shallow and simple tree that means high interpretability, no overfitting.

Nursery dataset -

- Depth: 7
- Nodes: 983
- Leaf nodes: 703
- Early splits: Parents, financial, social, and health attributes.
- Complexity: Large and complex tree due to multi-valued attributes and many output classes. Higher chance of overfitting.

TicTacToe dataset -

- Depth: 7
- Nodes: 260
- Leaf nodes: 165
- Early splits: Specific board positions (center, corners).
- Complexity: Medium complexity tree. Game patterns create overlapping rules, causing imperfect generalization.

The Mushroom dataset produced a shallow and highly interpretable tree, dominated by a few decisive features like odor and spore-print-color, showing no signs of overfitting. In contrast, the Nursery dataset generated a very large and deep tree with nearly a thousand nodes, reflecting its multi-valued attributes and diverse class labels, which increases complexity and overfitting risk. The TicTacToe dataset resulted in a moderately complex tree, where board positions drive early splits, but overlapping game patterns limit generalization. Overall, tree depth and size strongly correlate with dataset complexity and feature diversity.

3. Dataset - specific insights :

Mushroom dataset -

- Feature Importance: Odor and spore-print-color are the most critical features, dominating early splits.
- Class Distribution: Balanced between edible and poisonous classes.
- Decision Patterns: Very clear rules such as “odor = foul → poisonous” lead to perfect classification.
- Overfitting Indicators: None, as the tree is shallow, simple, and achieves perfect generalization.

Nursery dataset -

- Feature Importance: Parents, financial, social, and health conditions dominate splits.
- Class Distribution: Imbalanced with certain classes (like 'recommend' or 'priority') occurring more frequently than others.
- Decision Patterns: Complex rules combining multiple attributes like financial stability and parent's occupation.
- Overfitting Indicators: Large depth (7) and nearly 1000 nodes suggest overfitting risk due to high feature diversity and class imbalance.

TicTacToe dataset -

- Feature Importance: Central position and corners on the board drive classification strongly.
- Class Distribution: Some outcomes (win/loss) are more frequent, while draws are less represented, creating imbalance.
- Decision Patterns: Typical winning strategies are captured, but overlapping patterns reduce clarity.
- Overfitting Indicators: Moderate tree depth and many nodes; overfitting occurs as the tree tries to memorize game-specific board configurations rather than generalize.

4a. Algorithm performance

i. Which dataset achieved the highest accuracy and why?

The mushroom dataset achieved 100% accuracy because a few highly discriminative features (odor, spore-print-color) perfectly separate the classes.

ii. How does dataset size affect performance?

Larger datasets like Nursery increase complexity and tree size, making the model prone to overfitting but still maintaining high performance due to sufficient training data. Smaller structured datasets like TicTacToe show lower accuracy as patterns are less distinct.

iii. What role does the number of features play?

Fewer but highly informative features (Mushroom) lead to simple, accurate trees. Many multi-valued features (Nursery) increase complexity. Binary features (TicTacToe) generate many overlapping patterns, reducing generalization.

4b. Data characteristics impact

i. How does class imbalance affect tree construction?

In Nursery and TicTacToe, imbalance skews tree construction toward dominant classes, reducing macro precision/recall. Balanced classes (Mushroom) allow perfect splits.

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

Binary features (TicTacToe) can create rigid, less generalizable splits. Multi-valued features (Nursery) provide richer information but also make trees large and complex. Highly discriminative categorical features (Mushroom) work best for decision trees.

4c. Practical applications

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

- ★ Mushroom Dataset: Useful for food safety and biological classification.
- ★ Nursery Dataset: Relevant for decision support in child-care admission or welfare systems.
- ★ TicTacToe Dataset: Relevant for game AI and pattern recognition.

ii. What are the interpretability advantages for each domain?

- Mushroom: Highly interpretable with simple rules based on key features like odor.
- Nursery: Complex but mirrors human decision factors (parents, financials, health).
- TicTacToe: Shows clear game strategies through board position importance.

iii. How would you improve performance for each dataset?

- ➔ Mushroom: Already optimal, no improvement needed.
- ➔ Nursery: Apply pruning or ensemble methods (Random Forest, Gradient Boosting) to reduce overfitting and improve generalization.
- ➔ TicTacToe: Use feature engineering (e.g., grouping board states) or advanced models (SVM, Neural Networks) to better capture overlapping patterns.