Comparative Analysis Report - ID3 Decision Tree

Q1. Algorithm Performance

Q1.a) Which dataset achieved the highest accuracy and why?

Answer: The Mushroom dataset achieved the highest accuracy because it has strong discriminative features such as odor, spore-print-color, and gill-size that almost perfectly separate edible from poisonous mushrooms. The Tic-Tac-Toe dataset achieved moderate accuracy (~70–80%) due to the complexity of board configurations, while the Nursery dataset had the lowest accuracy (~65–75%) because of multiple classes and class imbalance.

Q1.b) How does dataset size affect performance?

Answer: Dataset size affects robustness and generalization. The Mushroom dataset (~8,000 samples) is large enough to learn clear rules without overfitting. The Tic-Tac-Toe dataset is smaller (~958 samples), making it prone to overfitting. The Nursery dataset (~12,000 samples) is larger, but due to many class labels and imbalance, complexity increases and minority classes suffer in precision and recall.

Q1.c) What role does the number of features play?

Answer: The number of features influences tree complexity. The Mushroom dataset has many features, but a few dominant ones make the tree shallow and accurate. Tic-Tac-Toe has 9 board position features that must all be checked, leading to deeper trees. Nursery has 8 multi-valued features that cause branching explosions and complex trees that are harder to interpret.

Q2. Data Characteristics Impact

Q2.a) How does class imbalance affect tree construction?

Answer: Class imbalance skews decision-making. The Mushroom dataset is balanced, so splits are fair. Tic-Tac-Toe has slight imbalance between winning and non-winning states, so the tree may favor the majority. Nursery is highly imbalanced, with most instances in 'not_recom', leading to biased trees that poorly classify minority classes like 'spec_prior' and 'very_recom'.

Q2.b) Which types of features (binary vs multi-valued) work better?

Answer: Binary features generally work better because they produce simpler, cleaner splits. Mushroom and Tic-Tac-Toe datasets (binary or small categorical features) result in shallow trees with good accuracy. Nursery dataset features are multi-valued, which leads to branching explosions, complex trees, and higher risk of overfitting.

Q3. Practical Applications & Improvements

Q3.a) For which real-world scenarios is each dataset type most relevant?

Answer:
• Mushroom Classification: Relevant to food safety, biology, and agriculture.
• Tic-Tac-Toe Endgame: Relevant to game AI, strategy modeling, and decision-making in finite state problems.
• Nursery School: Relevant to education and administrative decision-making based on family and social factors.

Q3.b) What are the interpretability advantages for each domain?

Answer: Decision trees are interpretable. For the Mushroom dataset, rules like 'if odor=foul → poisonous' are clear. For Tic-Tac-Toe, rules resemble actual strategies (e.g., 'if center=X and corner=X → likely win'). For Nursery, rules are more complex but still traceable, allowing policymakers to understand how features affect recommendations.

Q3.c) How would you improve performance for each dataset?

Answer:
• Mushroom: Already highly accurate; minimal improvements needed.
• Tic-Tac-Toe: Apply pruning to reduce overfitting or use ensemble methods like bagging/boosting.
• Nursery: Handle class imbalance with oversampling/undersampling, prune the tree to simplify structure, and use ensemble methods like Random Forest or Gradient Boosting for better performance.

Q4. Comparative Analysis Report

Answer: Overall, the Mushroom dataset performed best due to its strong categorical features and balanced classes. The Tic-Tac-Toe dataset performed moderately well, with deeper trees required to represent board states. The Nursery dataset had the most challenges due to multiple classes, imbalanced distribution, and multi-valued attributes, leading to complex trees and reduced performance. In practice, decision trees are useful for interpretable rule-based predictions, but improvements like pruning, balancing, and ensembles are necessary to handle dataset complexity and imbalance effectively.