3. Dataset Specific Insights:

Mushroom Dataset

- **Feature Importance:** The most important feature is ‘odor’ with an information gain of 0.9077, indicating it is highly effective at splitting the data and will likely be the root of the decision tree. The next most important features are ‘ population’, ’gill-size’, and ‘spore-print-color’, all with high information gain.
- **Class Distribution:** The dataset is nearly balanced, with 51.79% edible (e) and 48.20% poisonous (p) mushrooms.
- **Decision Patterns:** Given the high information gain of a single attribute like odor, the decision tree will be very shallow. A common decision path might be odor is foul→ mushroom is poisonous. The tree can quickly and accurately classify most instances with just a few splits.
- **Overfitting Indicators:** Due to the strong predictive power of a few key features, the resulting tree will be simple and have a low number of nodes. This simplicity suggests a low risk of overfitting to the training data. The clear and simple decision patterns are signs of a model that generalizes well.

TicTacToe Dataset

- **Feature Importance:** The most important feature is ‘middle-middle square’ with an information gain of 0.0989. However the information gains for all features are relatively low and similar to one another, suggesting that no single square's state is a strong predictor of the outcome on its own.
- **Class Distribution:** The dataset has a moderate imbalance, with 65.28% positive outcomes and 34.62% negative outcomes.
- **Decision Patterns:** Since individual features have low information gain, the decision tree must be much deeper and more complex to classify outcomes. The model will need to find decision paths that rely on combinations of features (multiple squares) to determine if a player has won. A common decision path might involve checking multiple board positions, such as, "If the top-left, top-middle, and top-right squares are all 'x', then the outcome is positive."
- **Overfitting Indicators:** The tree will be more complex and have a higher number of nodes due to the need for multiple splits to reach a conclusion. This complexity increases the risk of overfitting, as the model may create rules that are specific to the training examples rather than generalizable to all possible board configurations.

Nursery Dataset

- **Feature Importance:** The most important feature is ‘health’ with a high information gain of 0.9599, indicating it is a very strong predictor of the recommendation level.
- **Class Distribution:** This dataset has a severe class imbalance. The not_recom (33.33%), priority (32.91%), and spec_prior (31.20%) classes are well-represented, but the very_recom(2.53%) and recommend (0.02%) classes are extremely rare. This imbalance poses a significant challenge for the algorithm.
- **Decision Patterns:** The decision tree will likely have a dominant split at the root on ‘health’. From there, the paths for the less-frequent classes will be much deeper and more complex. The tree will learn specific rules to correctly classify the majority classes while struggling to correctly identify instances of the rare classes, which may lead to misclassifications.
- **Overfitting Indicators:** The presence of a dominant feature and a heavily imbalanced class distribution are clear indicators of potential overfitting. The high accuracy of 98.7% for this complex, imbalanced dataset suggests the model is highly optimized for the training data and may not perform as well on new, unseen data.

4. Comparative Analysis Report:

a. Which dataset achieved the highest accuracy and why?

The Mushroom dataset typically achieves the highest accuracy, reaching 100%. This is because it contains a small number of features with very high predictive power, such as odor and spore-print-color. The dataset is also nearly balanced, which prevents the algorithm from being biased toward a single class.

b. How does dataset size affect performance?

For decision trees, a larger dataset generally leads to a more complex tree with more nodes and greater depth. While a larger dataset can help a model generalize better and reduce the risk of overfitting by exposing it to a more comprehensive representation of the data, it can also lead to an overly complex tree that simply memorizes the training data. In some cases, increasing the dataset size beyond a certain point may not significantly improve accuracy, but it can cause the tree size to grow linearly, which is a sign of overfitting.

c. What role does the number of features play?

The number of features directly impacts the complexity of the decision tree. A higher number of features provides more options for splitting the data, which can increase the tree's complexity and lead to a deeper structure. A large number of features can increase computation time and, if many features are irrelevant, can confuse the algorithm and introduce noise.

b) Data Characteristics Impact

• How does class imbalance affect tree construction?

Class imbalance can significantly affect how a decision tree is constructed. The algorithm's goal is to maximize information gain by creating the purest possible child nodes. When one class vastly outnumbers the others, the tree-building process can become biased toward it, as splits that favor the majority class may seem to reduce overall error more effectively. This can result in a model that performs very well on the majority class but poorly on the minority class, leading to high overall accuracy but low recall for the underrepresented class. The Nursery dataset is a good example of this, where the ‘recommend’ class is severely underrepresented.

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

Decision trees can handle both binary and multi-valued features . Multi-valued features can be particularly effective because a single split can create multiple branches, leading to a quicker reduction in entropy.

c) Practical Applications

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

Mushroom: This type of dataset is relevant for real-world scenarios where a few key features are highly predictive of an outcome. Example: email spam filtering (e.g., classifying an email as spam if a specific keyword is present).

TicTacToe: This dataset is a classic example of a game-theory problem. It's relevant to scenarios where an outcome depends on a combination of actions or states rather than a single attribute. Real-world applications include strategic planning in business or logistics.

Nursery: This dataset, with its complexity and class imbalance, is a good representation of decision support systems in fields like admissions or credit scoring. It models a real-world scenario where a decision must be made based on multiple, often subjective, factors. For example, a bank might use a similar model to approve a loan application based on an applicant's financial history, credit score, and income.

• What are the interpretability advantages for each domain?

Mushroom: The interpretability advantage is the creation of simple, direct rules that can be easily communicated.

TicTacToe: The tree's interpretability helps understand the optimal sequence of moves in a game. By analyzing the tree, you can see the logical combinations of board positions that lead to a win or a loss.

Nursery: A decision tree can reveal exactly which criteria (e.g., family circumstances, health status) contributed most to an applicant's recommendation level.

How would you improve performance for each dataset?

Mushroom: Performance is already high. However, pruning could be applied to trim unnecessary branches.

TicTacToe: Performance can be improved by using better methods that can handle the complex combinations of board positions.

Nursery: The primary challenge here is class imbalance. Performance can be significantly improved by using sampling techniques.