<u>REPORT WEEK-3 ML LAB</u>

Comparative Analysis of ID3 Decision Tree on 3 datasets — Mushrooms, Tic-Tac-Toe, and Nursery

---

Analysis

Performance Comparison (from Colab runs)

MUSHROOM

- Accuracy: 100.00%
- Weighted precision: 1.0000, weighted recall: 1.0000, weighted F1-score: 1.0000
- Macro precision: 1.0000, macro recall: 1.0000, macro F1-score: 1.0000
- Comments: The attribute odor almost completely determines the class, producing a very shallow decision tree (~3–5 levels) and perfect classification on the dataset as run in Colab.

TIC-TAC-TOE

- Accuracy: 87.30%
- Weighted precision: 0.8741, weighted recall: 0.8730, weighted F1-score: 0.8734
- Macro precision: 0.8590, macro recall: 0.8638, macro F1-score: 0.8613
- Comments: The problem is harder because all nine board cells matter. The tree is deeper (~7–9 levels) and exhibits lower accuracy than Mushroom, with per-class metrics close to one another but lower than Mushroom's perfect values.

NURSERY

- Accuracy: 98.67%
- Weighted precision: 0.9876, weighted recall: 0.9867, weighted F1-score: 0.9872
- Macro precision: 0.7604, macro recall: 0.7654, macro F1-score: 0.7628
- Comments: High overall accuracy, but macro metrics are much lower than weighted metrics — indicating the classifier performs well on majority classes and less well on minority classes. Trees tend to be deep (often >10 levels) due to many-valued categorical attributes.

Tree Characteristics Analysis

- MUSHROOM: Very shallow (≈3–5 levels). Few nodes because a single highly discriminative feature (odor) divides classes cleanly. Little to no overfitting observed in the notebook runs.
- TIC-TAC-TOE: Moderately deep trees (≈7–9 levels). Many binary features (nine cells) create long decision paths. The central cell often appears as an early split because it strongly influences outcomes. Signs of overfitting appear in memorized board states.
- NURSERY: Deepest and largest trees (>10 levels). Many categorical attributes with many values increase branching factor and tree size. Early splits often select parents/finance features.

Dataset-Specific Insights

- MUSHROOM: Most informative feature is odor. Dataset is effectively separable; simple rules (e.g., *if odor = foul → poisonous*) are sufficient. No significant generalization issues per the notebook runs.
- TIC-TAC-TOE: Central and diagonal board positions are important. Because many board states are possible, the tree sometimes memorizes patterns rather than general strategies — pruning/feature engineering recommended.
- NURSERY: parents and finance are dominant features. Dataset is imbalanced (majority class: not_recom), so the model achieves high weighted scores but lower macro scores (shows weakness on minority classes). Consider imbalance handling.

Comparative Analysis

a) Algorithm Performance — Which dataset achieved the highest accuracy and why?

Ranking (from Colab runs):

1. Mushroom — 100.00%
2. Nursery — 98.67%
3. Tic-Tac-Toe — 87.30%

Why:

- *Mushroom* has a single extremely discriminative attribute (odor) which nearly perfectly splits the classes — therefore ID3 learns rules that generalize perfectly on the dataset used.
- *Nursery* benefits from large sample size and informative features, giving a very high accuracy; however, multiple classes and imbalance cause weaker macro metrics.

- *Tic-Tac-Toe* has many distinct board-state patterns and limited simple global rules, so ID3 produces deeper trees and lower generalization performance.

b) How does dataset size affect performance?

- Larger datasets (like Nursery) provide more examples per rule/combination, improving split statistics and generalization for majority classes, but they also create deeper, more complex trees (higher variance in structure).
- Smaller or highly combinatorial datasets (Tic-Tac-Toe) can lead to overfitting: many unique states combined with limited repetition encourage the tree to memorize training instances instead of learning generalizable rules.

c) What role does the number of features play?

- Few, highly informative features (Mushroom): Shallow trees, high interpretability, high accuracy.
- Many features or many-valued categorical features (Nursery): Increase branching (depth and node count), reduce interpretability, but with enough data maintain high accuracy for common classes.
- Many binary features across positions (Tic-Tac-Toe): Yield medium-depth trees and complex decision paths; interactions between positions are important and not easily captured by single splits.

Data Characteristics Impact

How does class imbalance affect tree construction?

- In Nursery, imbalance (majority = not_recom) causes early splits to favor majority-class improvement of overall accuracy but hurts minority-class precision/recall (macro scores drop). ID3 greedily optimizes impurity reduction and can therefore create branches that primarily separate the dominant class — leading to inflated global accuracy but poor minority detection.

Which feature types work better (binary vs multi-valued)?

- Binary features (Tic-Tac-Toe cells): provide simple splits but many such features cause combinatorial state explosion. Good when the problem truly depends on binary conditions and data is abundant.
- Multi-valued categorical features (Nursery): increase tree branching which raises complexity and reduces interpretability; they need many samples to reliably estimate split benefit for each value.
- Single highly discriminative categorical feature (Mushroom odor) is ideal for ID3: few splits, simple rules, high accuracy.

Practical Applications & Interpretability

Which real-world scenarios match each dataset type?

- Mushroom: High-stakes, interpretable classification (food safety, toxicity checks) where a small number of features dominate the decision and transparent rules are required.
- Tic-Tac-Toe: Educational or demonstration datasets for teaching decision trees, pruning and overfitting; also simple game AI prototypes.
- Nursery: Decision support systems with many stakeholder criteria (school/admission recommendation engines) — many classes and categorical features.

Interpretability advantages by domain:

- Mushroom: Highest interpretability — shallow tree and direct rules like "if odor = foul → poisonous."
- Tic-Tac-Toe: Moderate interpretability — you can trace important positions (center/diagonals) but decision paths are longer.
- Nursery: Lowest interpretability — deep, multi-valued splits make global logic harder to read; local rules exist but many branches complicate human understanding.

How to improve performance for each dataset (practical, Colab-applicable steps)

Mushroom (already perfect in runs):

- No urgent changes needed. For robustness, confirm with cross-validation and test on held-out or slightly perturbed data. If target domain needs generalization under noise, consider small pruning for stability.

Tic-Tac-Toe:

- Apply pruning (pre-pruning: max_depth, min_samples_leaf; post-pruning: cost-complexity/ccp_alpha) to reduce memorization.
- Feature engineering: add derived features that capture tactical patterns (e.g., "two in a row with empty third cell", "center control"), so tree splits on game tactics rather than raw cell combinations.
- Use k-fold cross-validation (stratified if classes imbalanced) to measure generalization and tune pruning hyperparameters.
- Consider ensemble models (RandomForest/GradientBoosting) to reduce variance, or use smoothed counts for rare patterns.

Nursery:

- Handle imbalance: resampling (SMOTE/oversampling, undersampling) or class weights in tree learning (if implementation supports it) to boost minority class performance.
- Pruning (cost complexity) to reduce overfitting to rare combinations.
- Group rare categorical values (merge infrequent categories into an "other" bucket) to reduce branching factor.
- Evaluate per-class metrics (precision/recall/confusion matrix) and focus on classes with low recall/precision (macro metrics indicate minority weakness).
- Consider using a tree ensemble or calibrated classifier if interpretability is less critical than minority performance.

Implementation & evaluation checklist (practical experiments to run next)

1. Run stratified k-fold cross-validation (5 or 10 folds) and report mean ± std of accuracy and macro F1.
2. Print confusion matrices and per-class precision/recall for Nursery and Tic-Tac-Toe to identify specific weak classes.
3. Tune pre-pruning (max_depth, min_samples_split, min_samples_leaf) via grid search on validation folds.
4. Test post-pruning using cost-complexity pruning (ccp_alpha) and find alpha that improves macro F1 for Nursery or Tic-Tac-Toe.
5. For Nursery, test resampling strategies (SMOTE, random oversample) and class weights to improve macro metrics.
6. For Tic-Tac-Toe, create engineered features capturing game tactics and compare results.

7. Optionally compare with RandomForest and XGBoost/LightGBM to measure improvement in overall and per-class metrics (while noting decreased interpretability).

Final summary (explicitly anchored to Colab metrics)

- Mushroom: Perfect classification in your Colab run — 100.00% accuracy and 1.000 macro/weighted scores; shallow tree with a dominant feature (odor). Minimal action required besides robust cross-validation for confirmation.
- Nursery: Very high overall accuracy (98.67%) but lower macro metrics (~0.76) revealing weak performance on minority classes. Address class imbalance and prune/regularize tree depth for better minority recall.
- Tic-Tac-Toe: Moderate accuracy (87.30%) and balanced but lower macro scores compared to Mushroom. Overfitting to board states is present — prune and engineer tactical features to generalize better.