Group 5 Final Project Report

1. Introduction / Executive Summary

This project focuses on the classification of mushrooms as either edible or poisonous using machine learning techniques. The dataset, sourced from the UCI Machine Learning Repository, provides categorical features describing mushroom characteristics. The objective is to build interpretable and effective models that not only predict edibility with high accuracy but also offer visual insights into feature importance and decision pathways.

The core approaches include:

- Decision Tree
- Random Forest
- SuperTree (Enhanced Decision Tree Visualization)
- Sunburst Chart (Multivariate Visual Insight)

The models demonstrated excellent classification accuracy, with the Random Forest achieving the best predictive performance. The system prioritizes visual interpretability and baseline benchmarking of different classifiers.

Improvements After the Last Presentation

- Introduced **SuperTree** for more intuitive visual analysis
- Integrated **Sunburst Chart** to assist non-technical users in understanding feature relationships

- Calculated **Information Gain** for feature importance
- Refactored codebase to follow modular structure
- Updated baseline.py to include comparative metrics output and runtime tracking
- Ensured all visuals are saved to source_code/visuals/ for documentation

2. Problem Statement & Data Description

The problem originates from the mushroom classification challenge: determine if a mushroom is poisonous or edible based on visual and structural features. The classification problem is purely categorical and ideal for tree-based models.

Dataset Source:

- UCI Mushroom Dataset: https://archive.ics.uci.edu/ml/datasets/Mushroom
- Benchmark target: Class column (p = poisonous, e = edible)
- Features include cap shape, odor, gill spacing, spore print color, etc.
- 8,124 samples and 22 categorical attributes (after cleaning)

3. Approaches, Algorithms, and Code Structure

Algorithms Used

- **Decision Tree (Scikit-learn)**: Visual and interpretable baseline model.
- **Random Forest (Scikit-learn)**: Ensemble method for increased robustness.
- **SuperTree**: A wrapper for scikit-learn tree classifiers with better visual interpretation.
- **Sunburst Chart (Plotly)**: Categorical data visualization to support non-technical users.

- **Information Gain (Manual Calculation)**: Quantifies the most discriminative features.

Tools and Libraries

- Python 3.11
- scikit-learn
- matplotlib
- pandas
- plotly
- numpy
- os / pathlib for modular I/O

4. Metrics and Evaluation Setup

All classifiers were evaluated using:

- **Accuracy Score**
- **Precision, Recall, F1-Score**
- **Confusion Matrix**
- **Runtime/Scalability Tracking**
- Visual outputs for interpretability

Each model was trained on an 80/20 train-test split with a fixed random_state=42.

5. Analysis and Comparisons

Model	Accuracy	AUC	Improvement
Decision Tree	0.973	0.997	0.344
SuperTree	1.000	1.000	0.371
Random Forest	1.000	1.000	0.371
Majority Class (baseline)	0.629	-	-

6. Conclusion: Challenges & Lessons Learned

Challenges:

- Ensuring clean and accurate encoding of categorical data
- Designing visuals that are informative and not redundant

- Modularizing the code while preserving interdependencies
- Managing feature overlap across models without redundancy

Lessons Learned:

- Tree-based classifiers excel on discrete categorical data
- Visual interpretability is critical when building models for educational or non-technical audiences
- Modular architecture improves scalability and testing across models
- Information gain can pre-validate which features the models will favor