

1. Dataset Overview

The core of this project is the **mushrooms_decoded.csv** dataset, sourced from a well-known collection available on Kaggle

(<https://www.kaggle.com/datasets/uciml/mushroom-classification>).

It contains **comprehensive attributes of mushroom species**, with the goal of classifying them as **edible or poisonous**. Each row in the dataset represents a mushroom specimen described by multiple features.

Key Features Include:

- **Cap Characteristics:** Shape, surface, color
- **Bruises:** Binary indicator
- **Odor:** Distinct smell categories
- **Gill Details:** Attachment, spacing, size, and color
- **Target Variable:** Edibility (edible or poisonous)

The dataset is **categorical in nature**, making it highly suitable for supervised classification tasks.

2. Data Preprocessing

Before training, the raw dataset undergoes essential preprocessing to ensure it is clean, consistent, and model-ready. This process is automated via `preprocess.py` and includes:

- **Missing Value Handling:** Rows/columns with excessive missing data are removed or imputed (e.g., mode-based imputation).
- **Categorical Encoding:** All categorical features are label encoded into numerical form, making them compatible with machine learning algorithms.
- **Normalization:** Although most features are categorical, any numerical data is standardized to ensure fair treatment during training.

These steps enhance model performance and ensure data integrity.

3. Model Training

With a clean dataset, model development is executed using `train_model.py`, which trains a **Random Forest Classifier**, chosen for its robustness and ability to handle categorical data well.

Training Pipeline:

- The data is split into **training and test sets**.
- The model learns patterns linking input features to mushroom edibility.
- Evaluation metrics such as **accuracy, precision, and recall** are computed to assess model performance.

GUI Integration:

The trained model is embedded into a **Streamlit-based GUI (`gui_app.py`)** that allows users to input mushroom features and receive real-time predictions, complete with confidence scores.