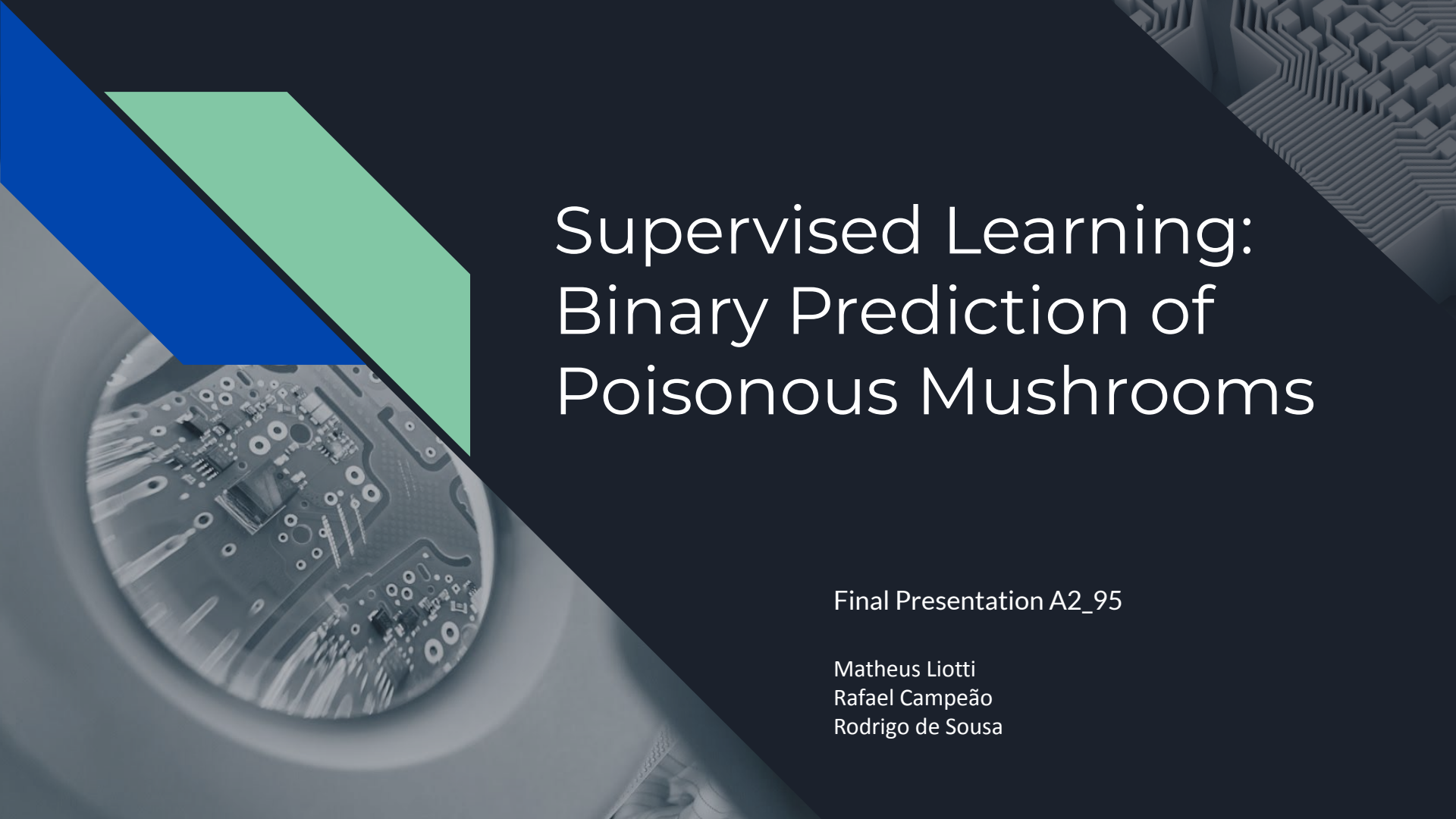


Supervised Learning: Binary Prediction of Poisonous Mushrooms

Final Presentation A2_95

Matheus Liotti
Rafael Campeão
Rodrigo de Sousa

Problem Definition

We are solving a binary classification problem to predict whether a mushroom is **poisonous** or **edible** based on its **physical characteristics**. The target variable is “Class”, which can be either “p” (poisonous) or “e” (edible).

The dataset is from the Kaggle Competition “Playground Series S4E8” :

https://www.kaggle.com/competitions/playground-series-s4e8

Related Work

01 The Dataset From the Kaggle Competition:

https://www.kaggle.com/competitions/playground-series-s4e8

02 This Notebook by Althariq Fairuz:
https://www.kaggle.com/code/althariqfairuz/binary-prediction-of-poisonous-mushrooms

03 UCI Mushroom Dataset:
https://archive.ics.uci.edu/dataset/73/mushroom

Methodology

Tools: Python (Pandas, NumPy, Scikit-learn, Seaborn, Matplotlib)

Algorithms planned:

- Decision Trees
- Random Forest Classification
- Support Vector Machines (SVM)

Algorithms Developed:

- Decision Trees, Random Forest Classification
- Gaussian Naive Bayes
- Gradient, Categorical Boosting Classifier
- K Nearest Neighbour

Evaluation Metrics:

- Accuracy, Precision, Recall, F1-score, Support, Confusion Matrix, Training/Testing Time.

Work Done - Final

Data Preprocessing:

- Removed Unnecessary Columns and Filled Missing Values
- Determined Strong Correlations
- Balanced the Data

Supervised Learning:

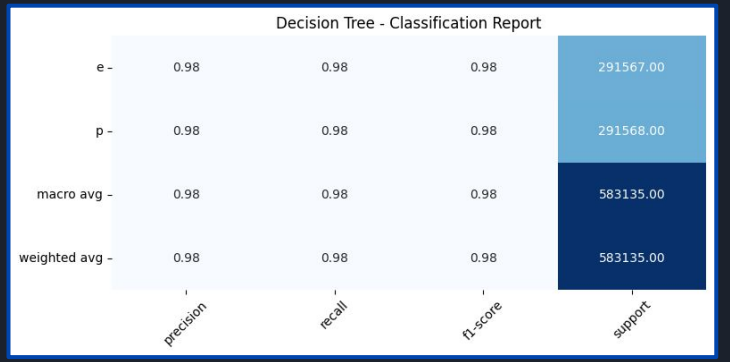
- 6 Models
- Classification Reports, Confusion Matrices and Feature Importance
- Comparison Results

Data Preprocessing

To ensure effective model training, we performed several essential preprocessing steps on the dataset. Since must features in the mushroom dataset are categorical, we applied **ordinal encoding** to convert each categorical variable into numerical format, enabling compatibility with scikit-learn algorithms.

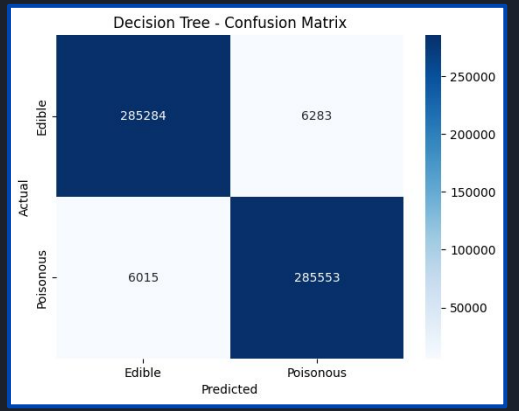
We also checked for **missing values**, confirming that the dataset was clean and complete. The data was then **split into training and testing sets** using an 75/25 stratified split to preserve class distribution. A **pipeline** was used to streamline the preprocessing and modeling workflow, ensuring consistency and reproducibility across all experiments. These preprocessing steps were crucial for maximizing model performance and maintaining a robust and generalizable learning process.

Algorithm and Evaluation
- Decision Trees



Justification:

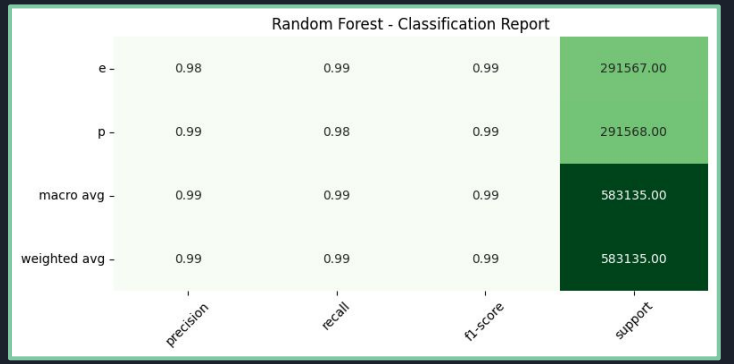
- Handles categorical features well (especially with OrdinalEncoder).
- Fast training and prediction.
- First Algorithm we learnt



Evaluation:

- Training Time: 57.636 seconds
- Testing Time: 2.254 seconds
- Accuracy: 97.89%

Algorithm and Evaluation
- Random Forest

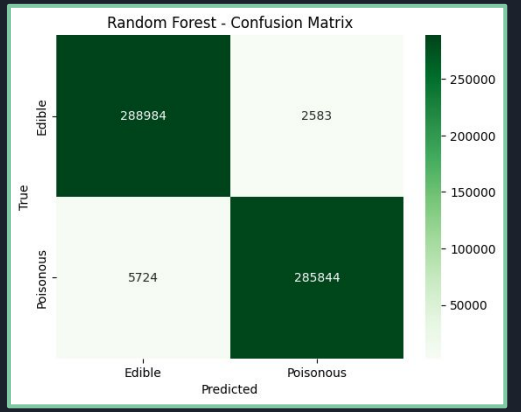


Justification:

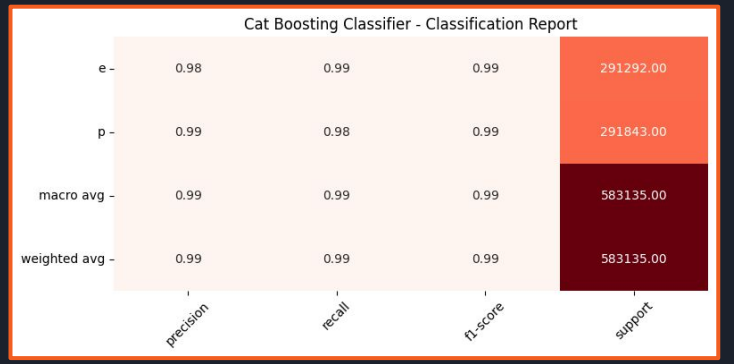
- Combines the predictions of multiple Decision Trees to increase accuracy and robustness.
- Unlike a single Decision Tree, it generalizes better on unseen data.

Evaluation:

- Training Time: 98.11 seconds
- Testing Time: 3.086 seconds
- Accuracy: 98.58%



Algorithm and Evaluation
- Categorical Boosting

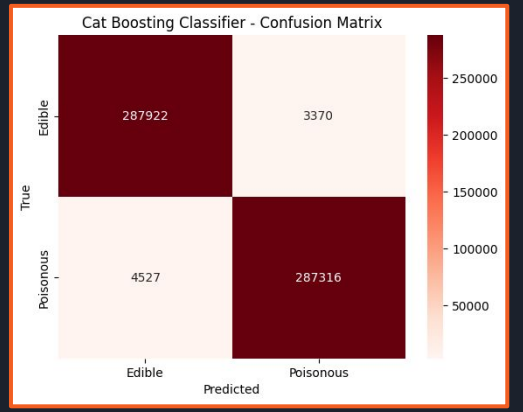


Justification:

- Supports both categorical and numerical features natively
- Is optimized for categorical data

Evaluation:

- Training Time: 6 minutes and 38 seconds
- Testing Time: 6.05 seconds
- Accuracy: 98.65%



Conclusions 🍄

In this project, we successfully applied supervised learning techniques to classify mushrooms as poisonous or edible using a balanced dataset. Through a complete machine learning pipeline, we implemented and compared six different algorithms: Decision Tree, Random Forest, Gaussian Naive Bayes, Gradient Boosting Classifier, Categorical Boosting Classifier and K Nearest Neighbors. Each model was assessed using standard classification metrics and timing benchmarks.

Among these, the Random Forest classifier demonstrated the best overall performance, offering high accuracy and robustness while mitigating overfitting. However, The Categorical Boosting gave us the highest accuracy results. The results confirm the importance of model selection and preprocessing in achieving reliable predictions.

This assignment not only reinforced our understanding of supervised learning concepts but also provided hands-on experience in building scalable and interpretable classification systems using Python and Scikit-learn.