CSE422 Lab Project Report

Course: CSE422 - Artificial Intelligence Lab

Project Title: Mushroom Classification Using Machine Learning

Submitted by:

Name: Faria Hoque Tazree

ID: 22299382

Name: Masrura Nahian Nodi

ID: 22201421

Total Pages: 16

Table of Contents

Dataset pre-processing

- Faults
- Solutions

Dataset splitting

- Random/Stratified (as required)
- Train set (70%)
- Test set (30%)

Model training & testing

- Neural Network
- Decision Tree
- Naive Bayes

Model selection/Comparison analysis

- Bar chart showcasing prediction accuracy of all models
- Precision and recall comparison of each model
- Confusion Matrix
- AUC score, ROC curve

Conclusion

- Summary of findings and future recommendations.

1. Introduction

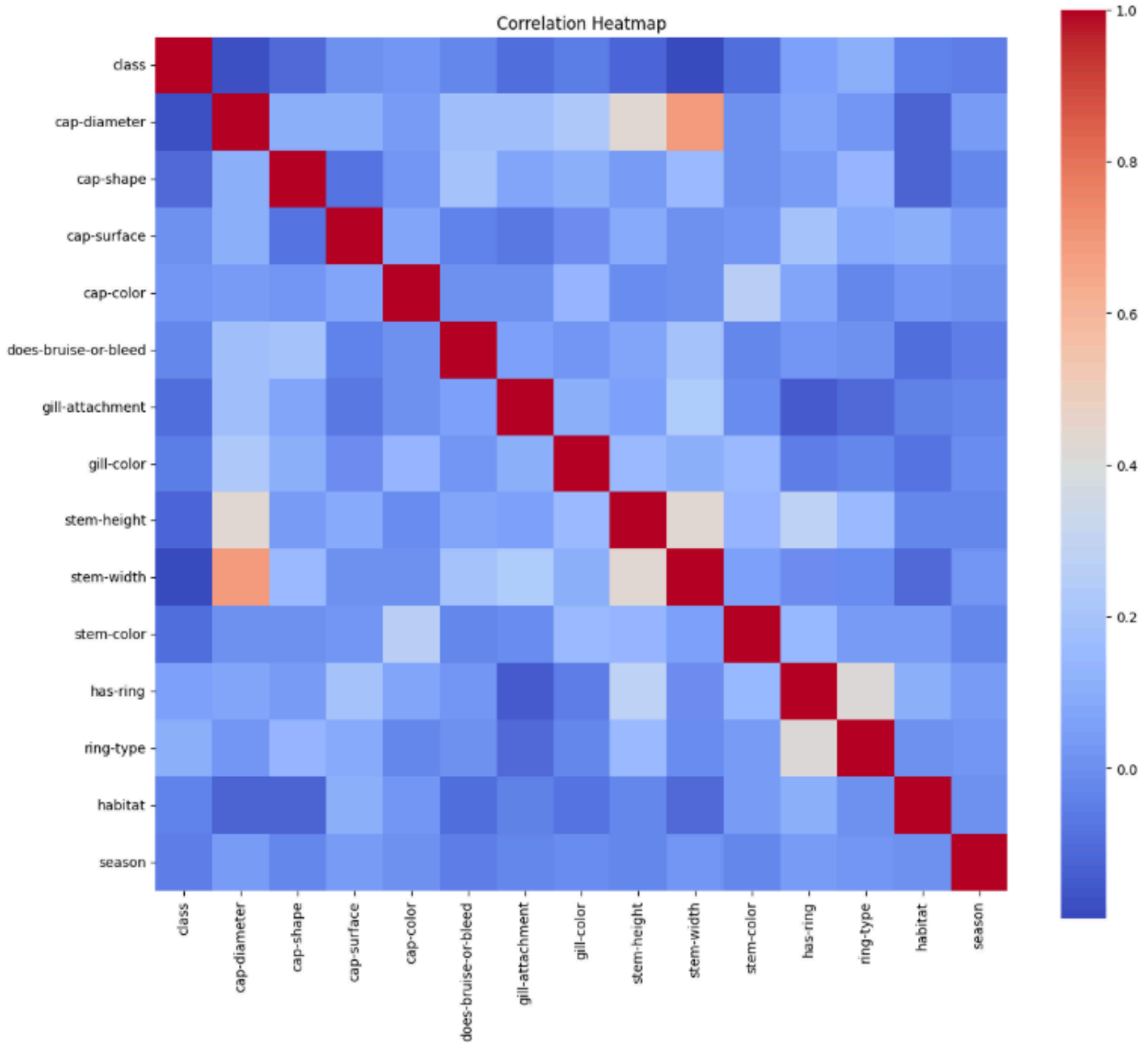
This project aims to classify mushrooms as either edible or poisonous using machine learning. Mushroom classification is vital in real-life applications, particularly for foragers and food safety, where misidentification can have serious health risks. We applied supervised learning models, including a Neural Network, Decision Tree, and Naive Bayes to solve this binary classification problem.

2. Dataset Description

- **Number of Features**: Initially 21; reduced to 15 after preprocessing
- **Type:** This is a classification problem because the target variable is categorical (edible or poisonous).
- **Number of Data Points**: 61,069 entries
- **Feature Types**:
    - Quantitative: cap-diameter, stem-height, stem-width
    - Categorical: cap-shape, cap-color, odor, gill-size etc.

Correlation of Features (with Heatmap)

A correlation heatmap was generated using the Seaborn library to observe how numeric features relate to each other. It showed low to moderate correlation, suggesting that each numeric feature provides relatively independent information.



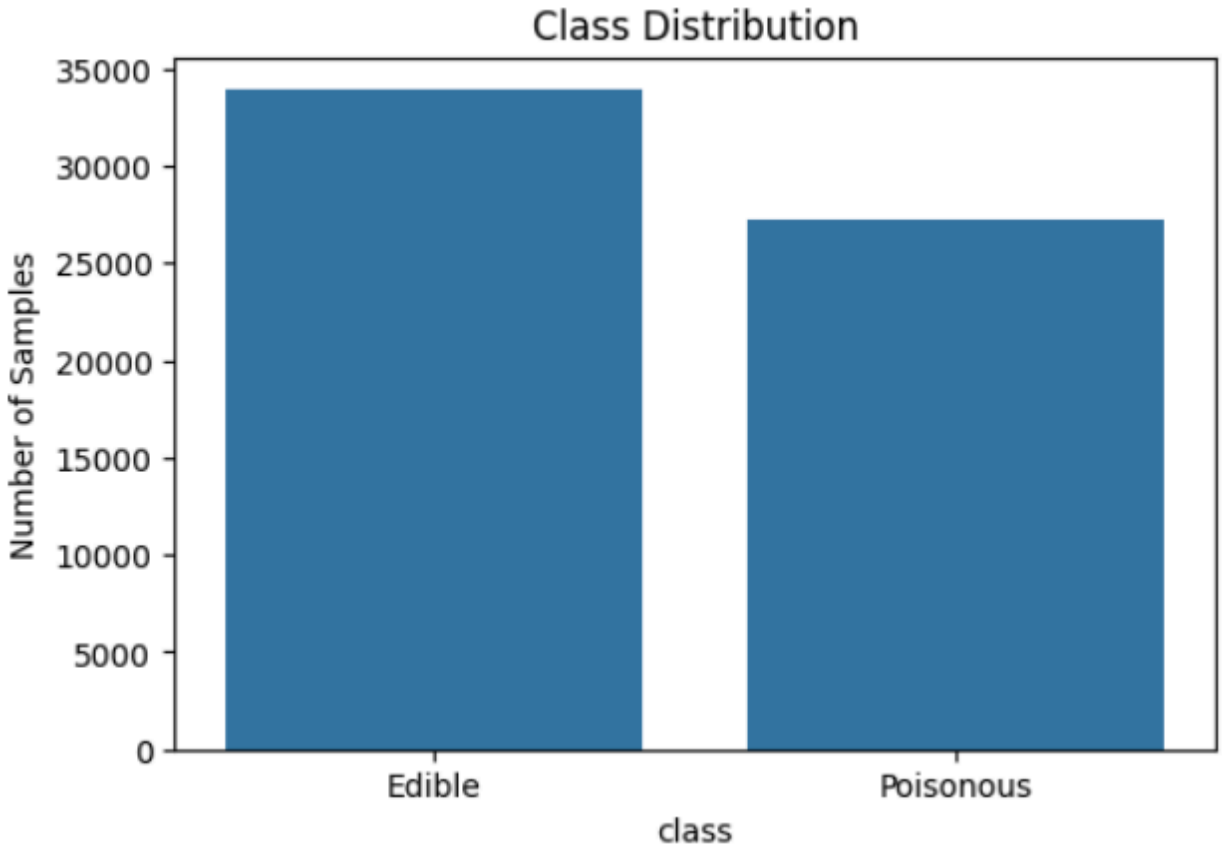
What We Understand from the Correlation Test

Most features are weakly correlated, meaning no strong multicollinearity is present. This supports retaining all numeric features, as they each provide unique signals that may help model learning.

Imbalanced Dataset

- Edible: 33888 samples
- Poisonous: 27181 samples

These values were calculated using value_counts() on the class column before preprocessing. A bar chart of class distribution was generated to visualize this.



Exploratory Data Analysis

EDA revealed that certain features (like odor, spore color, and gill-color) are highly indicative of mushroom class. For example, certain odors are exclusive to poisonous mushrooms.

3. Dataset Pre-processing

1. Null / Missing Values

Problem:

Several columns contained missing values. For example, stem-root, veil-type, and spore-print-color had many missing entries, making them unreliable.

Solution:

- Columns with **e**xcessive missing values were dropped to avoid data distortion.

- For remaining missing values in categorical columns, we imputed with the string "Missing" to retain samples without injecting bias.

2. Categorical Values

Problem:

Many features were categorical (e.g., cap-shape, odor, gill-attachment). Machine learning models require numeric inputs, so this format was not compatible.

Solution:

- We used One-Hot Encoding to convert these categorical features into binary vectors.

- This method was chosen because it preserves the non-ordinal nature of categorical values and avoids false numeric relationships.

3. Feature Scaling

Problem:

Quantitative features like cap-diameter, stem-height, and stem-width had varying scales and units. This could bias models like Neural Networks.

Solution:

- We applied StandardScaler to normalize these numeric features to a mean of 0 and standard deviation of 1.

- This ensures consistent weighting and better convergence during model training.

4. Dataset Splitting

- **Strategy**: Stratified sampling
    - **Training Set**: 70%
    - **Test Set**: 30%

Stratified sampling was used to maintain the class balance in both splits.

5. Model Training & Testing

Models Used:

1. Decision Tree:

- Parameters: max_depth=12, min_samples_split=5

- Chosen for its interpretability and speed.

2. Neural Network:

- Parameters: hidden_layer_sizes=(20,), alpha=0.01, max_iter=200

- Chosen for its ability to learn complex non-linear relationships.

3. Naive Bayes:

- Assumes feature independence
- Very fast but sensitive to this assumption.

All models were trained on the preprocessed training data and evaluated on the test set.

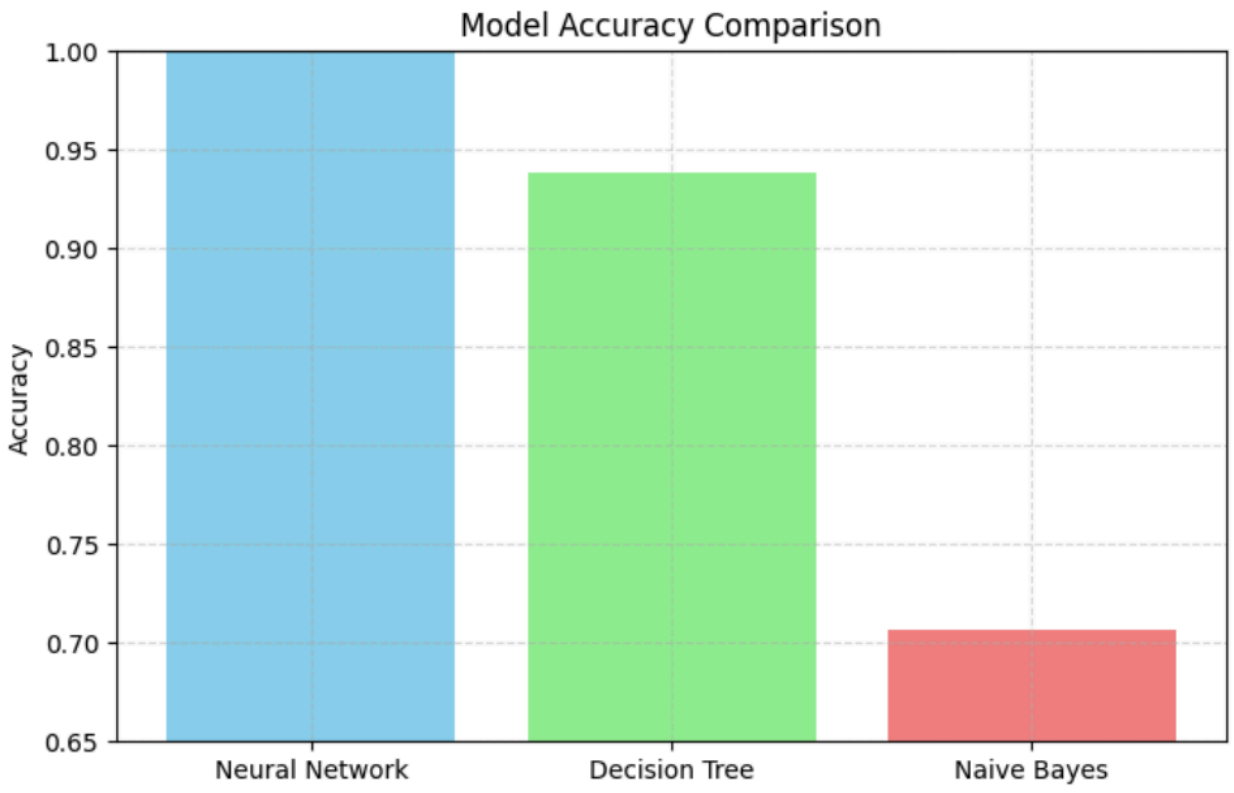
6. Model Selection/Comparison Analysis

Accuracy, Precision, Recall

Model	Accuracy	Precision	Recall
Decision Tree	0.939	0.973	0.914
Neural Network	0.999	0.999	0.9997
Naive Bayes	0.707	0.698	0.830

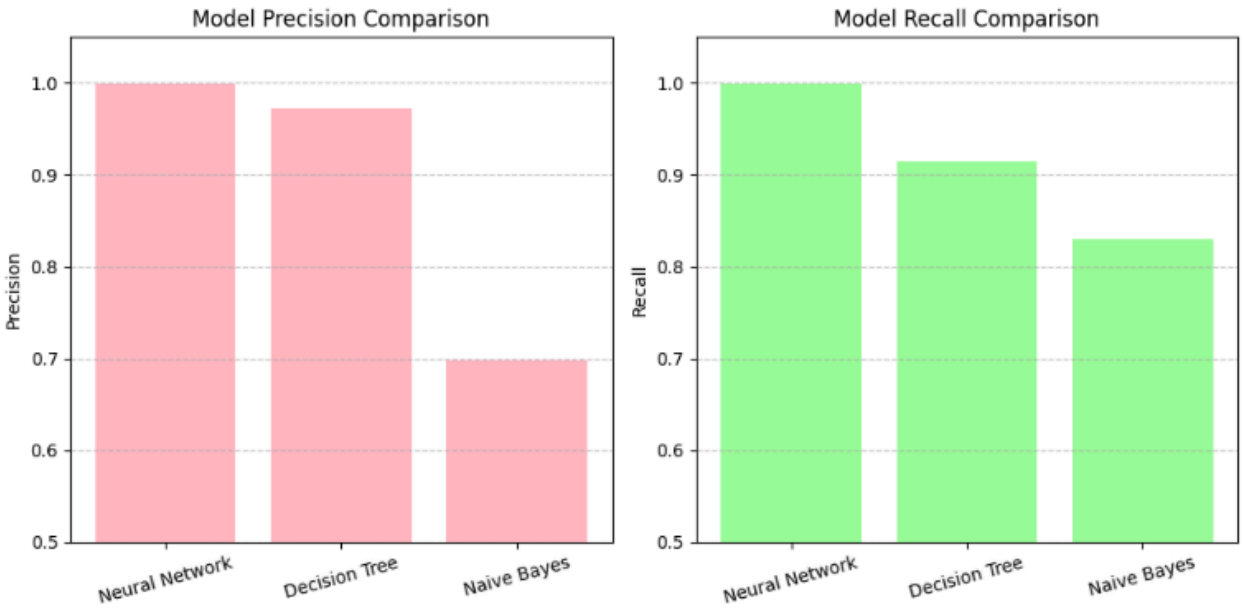
Bar Chart of Accuracies

A bar chart was plotted to visually compare the accuracy of all three models using Matplotlib.



Precision and Recall Comparison

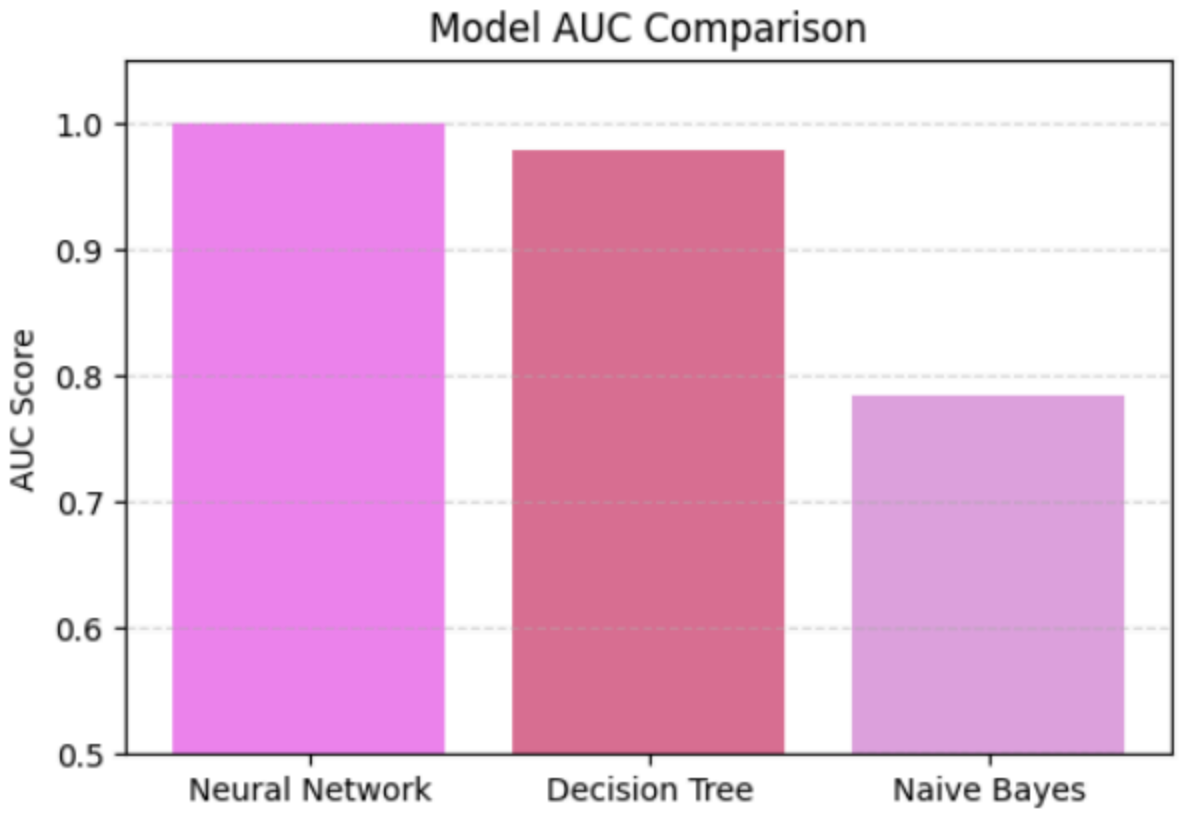
Two separate bar charts were generated side by side to compare precision and recall across the models. These charts showed that the Neural Network had the highest values, followed closely by the Decision Tree.



AUC Score Bar Chart

An additional bar chart was included to directly compare AUC scores of all models:

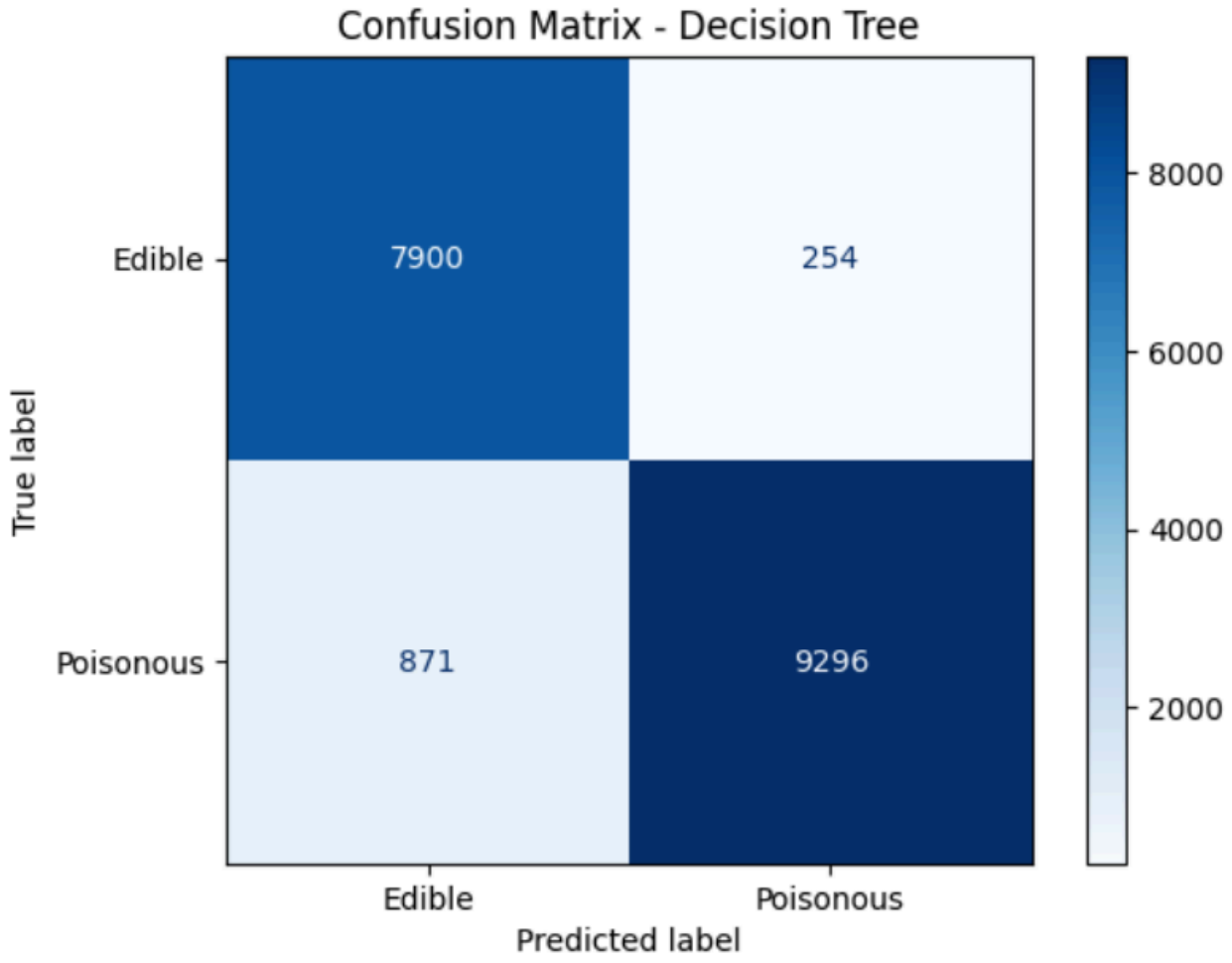
- **Neural Network**: AUC = 0.999999
- **Decision Tree**: AUC = 0.98
- **Naive Bayes**: AUC = 0.78



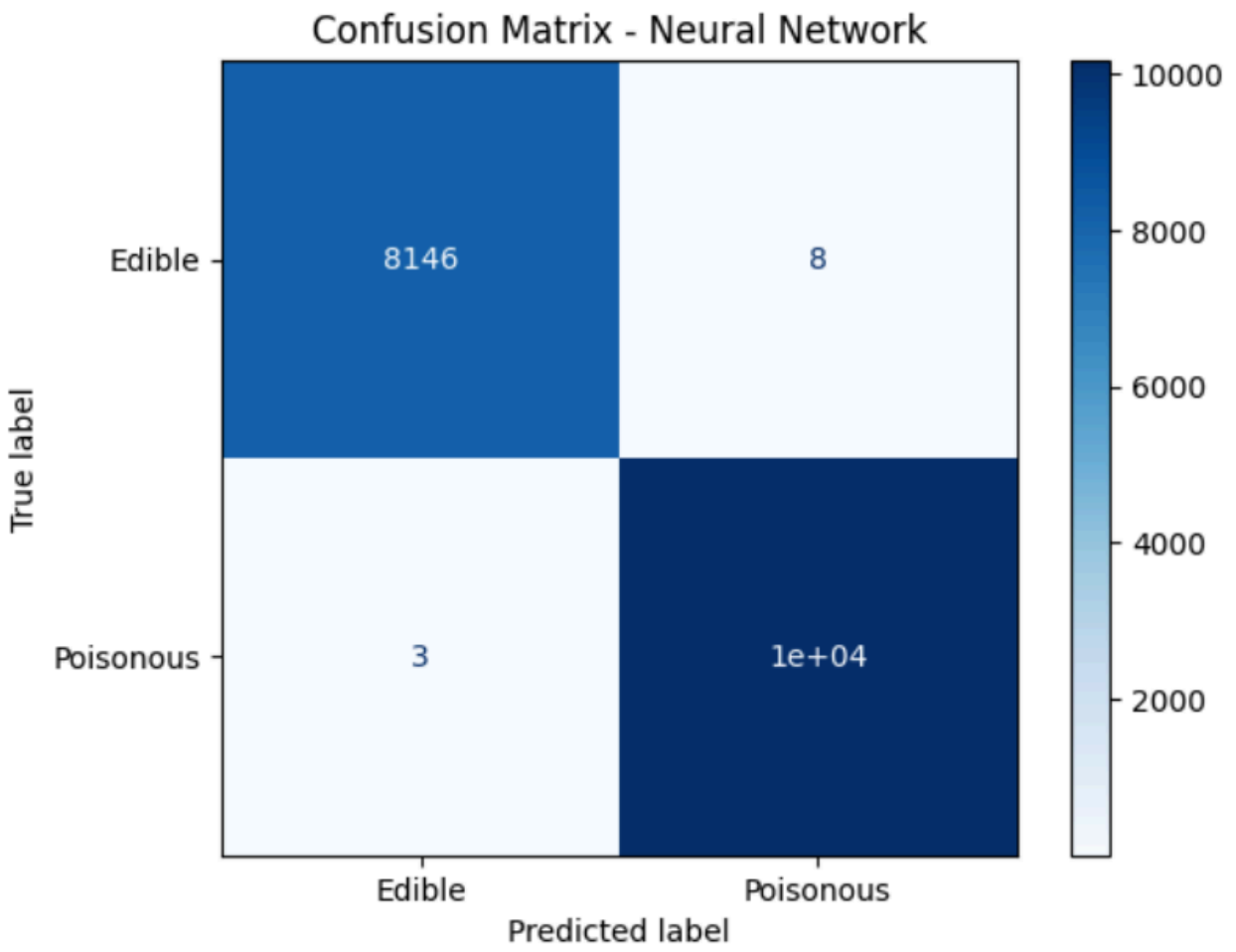
Confusion Matrices

Confusion matrices were plotted for each model to understand their prediction patterns on the test set:

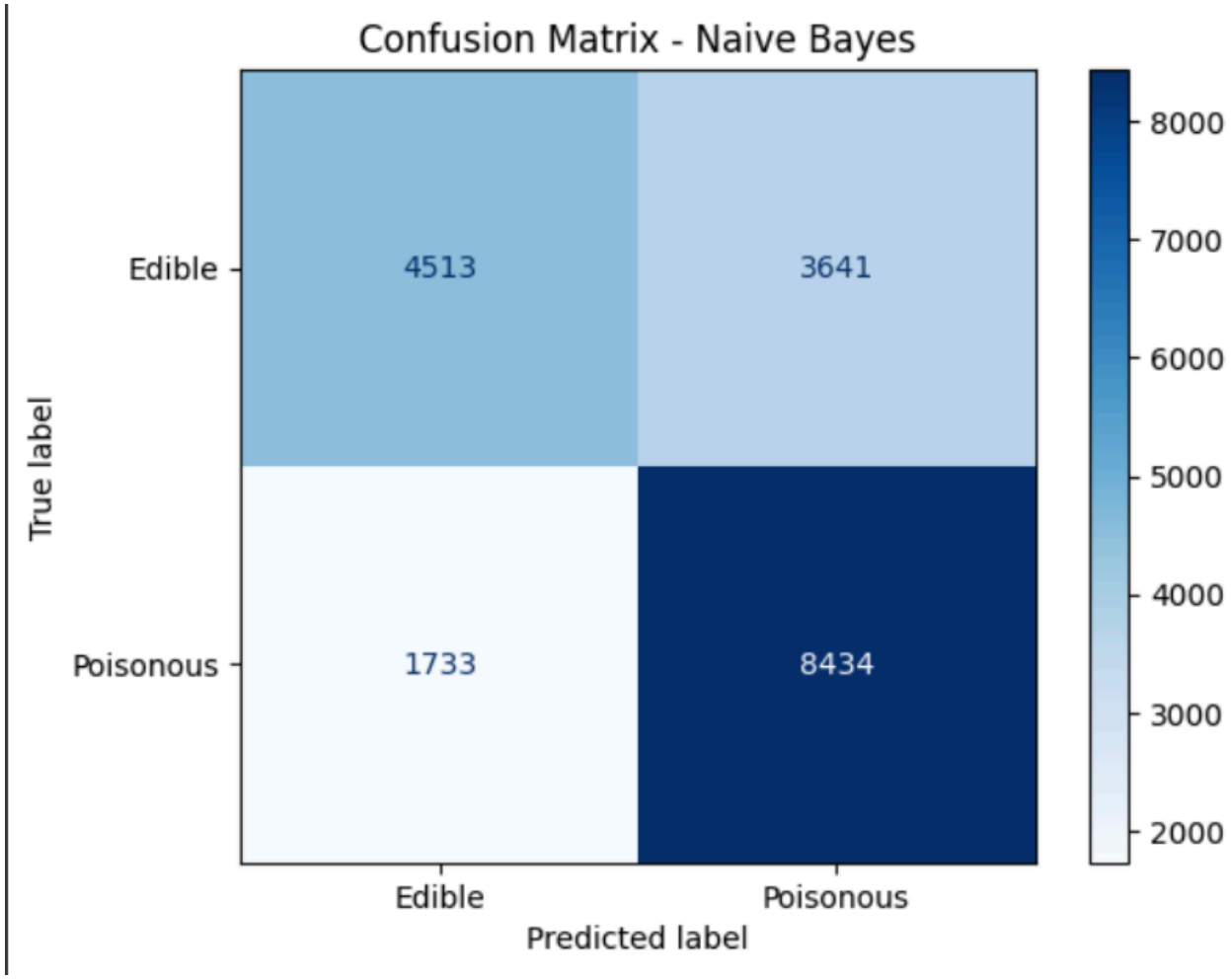
- **Decision Tree**:



- **Neural Network**:

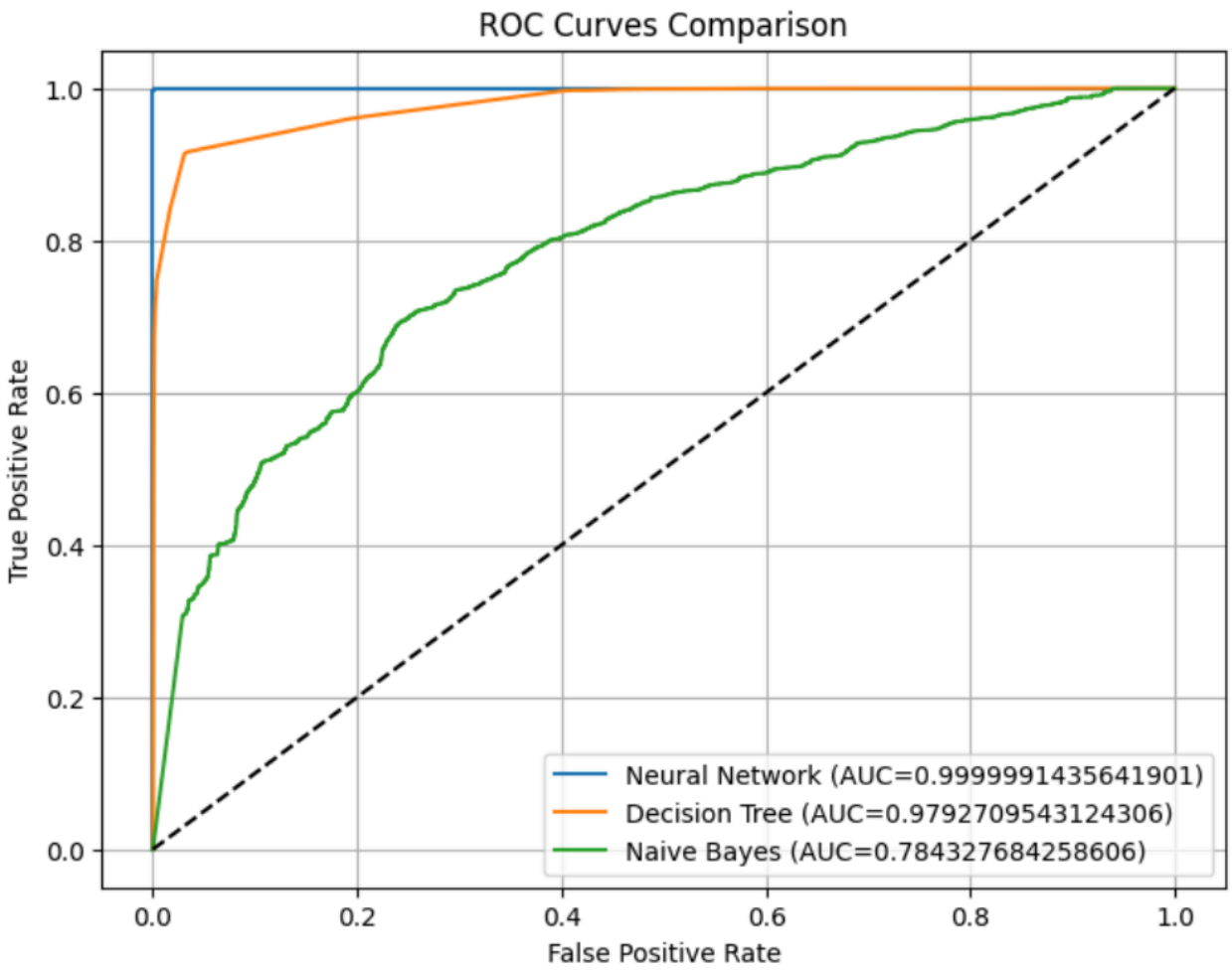

Confusion Matrix - Neural Network
Edible
Poisonous
True label
8146
8
3
1e+04
Edible
Poisonous
Predicted label
10000
8000
6000
4000
2000

- **Naive Bayes**:



ROC Curve and AUC Scores

ROC curves were plotted to visualize the trade-off between true positive and false positive rates across thresholds. The AUC (Area Under Curve) score was labeled for each model:



Summary of Results

- **Neural Network**: Best performance, near-perfect accuracy and recall
- **Decision Tree**: Reliable and interpretable with high precision
- **Naive Bayes**: Reasonable recall but poor precision and accuracy

Insights

- Neural Network excels due to its ability to learn complex feature interactions

- Decision Trees are fast, interpretable, and still perform strongly
- Naive Bayes is not suitable when feature independence is violated

Challenges

- Tuning neural networks to avoid overfitting
- Encoding and scaling mixed feature types
- Dealing with missing or inconsistent data entries
- Understanding class distribution early helped with modeling decisions

Recommendation

We should use Neural Network when performance is the highest priority and Decision Tree when interpretability is needed or for real-time classification with limited compute.