Lab Report

on

“Decision Tree and Random Forest Classifiers on the Mushroom Dataset.”

Course Title: Machine Learning.
Course Code: CSE475
Section: 02

Submitted by:
Md Tariqul Islam
ID:2021-1-60-054

Submitted to:
Dr.Raihan ul Islam(DRUI)
Associate Professor
Department of Computer Science and Engineering
East West University

Submission Date: 12/7/2024

Introduction

The Mushroom dataset from the UCI Machine Learning Repository describes various types of mushrooms. It includes 23 species of gilled mushrooms in the Agaricus and Lepiota families. Each species is labeled as definitely edible, definitely poisonous, or of unknown edibility and not recommended. The goal is to classify these mushrooms as either edible or poisonous based on their features.

Decision Tree Classifier

We used a Decision Tree classifier to classify the mushrooms. Decision Trees are straightforward yet powerful models that split the data into subsets based on the most important features, creating a tree-like structure.

First, we loaded the dataset, handled any missing values, and converted categorical features to numerical values using one-hot encoding. Then, we split the data into training and test sets. We trained the Decision Tree classifier on the training data, and then used it to predict labels for the test data. We evaluated the model's performance using accuracy and a classification report. The Decision Tree classifier achieved perfect accuracy, effectively classifying both edible and poisonous mushrooms.

Here's the classification report presented in table form:

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	**0.98**	**0.93**	**0.95**	**843**
1	**0.93**	**0.98**	**0.95**	**782**
Accuracy	**0.95**	**0.95**	**0.95**	**1625**
Macro Avg	**0.95**	**0.95**	**0.95**	**1625**
Weighted Avg	**0.95**	**0.95**	**0.95**	**1625**

Random Forest Classifier

We also used a Random Forest classifier, which is an ensemble of Decision Trees. Random Forests improve classification performance by reducing overfitting and providing more robust predictions.

For this model, we followed the same steps as with the Decision Tree: loading and preprocessing the data, converting categorical features to numerical values, and splitting the data into training and test sets. We then trained the Random Forest classifier with 100 trees on the training data and used it to predict labels for the test data. The Random Forest classifier also achieved perfect accuracy, providing excellent precision and recall values for each class.

Both classifiers performed exceptionally well on the Mushroom dataset, demonstrating their effectiveness in distinguishing between edible and poisonous mushrooms.

Here's the classification report presented in table form:

Class	Precision	Recall	F1-Score	Support
False	**1.00**	**1.00**	**1.00**	**424**
True	**1.00**	**1.00**	**1.00**	**705**
Accuracy	**1.00**	**1.00**	**1.00**	**1129**
Macro Avg	**1.00**	**1.00**	**1.00**	**1129**
Weighted Avg	**1.00**	**1.00**	**1.00**	**1129**

#Random Forest Accuracy for different n_estimators values:
n_estimators = 1: Accuracy = 1.0000
n_estimators = 50: Accuracy = 1.0000
n_estimators = 100: Accuracy = 1.0000
n_estimators = 150: Accuracy = 1.0000
n_estimators = 200: Accuracy = 1.0000
n_estimators = 250: Accuracy = 1.0000

Comparing Performances

Both classifiers, Decision Tree and Random Forest, achieved perfect accuracy on the Mushroom dataset. This indicates that the dataset is highly separable and the models can easily distinguish between edible and poisonous mushrooms.

Differences:

- **Model Complexity:** The Decision Tree is simpler and easier to interpret, whereas the Random Forest is an ensemble model, making it more complex but generally more robust.
- **Overfitting:** Decision Trees are prone to overfitting, especially with complex datasets. Random Forests mitigate this by averaging multiple Decision Trees, leading to better generalization.
- **Interpretability:** Decision Trees provide clear, interpretable rules for classification, while Random Forests, being ensembles, are less interpretable.

Although both the Decision Tree and Random Forest classifiers achieved perfect accuracy on the Mushroom dataset, here are some steps to improve their performance:

Decision Tree: To improve a Decision Tree's performance,we need to focus on feature engineering by creating new features or selecting the most relevant ones. Tuning hyperparameters like maximum depth, minimum samples for splits and leaves, and the number of features considered can also help. Pruning the tree, either post-creation or during its growth, can reduce overfitting. Using cross-validation ensures that the model generalizes well to new data.

Random Forest: For Random Forests, similar feature engineering techniques apply. Hyperparameter tuning, such as adjusting the number of trees, maximum features, and other tree-specific parameters, can boost performance. Increasing diversity within the forest by using different subsets of features and varying random states can make the model more robust.

By applying these techniques, we can significantly enhance the performance and robustness of Decision Tree and Random Forest classifiers in various datasets.

Conclusion

Both the Decision Tree and Random Forest classifiers worked incredibly well on the Mushroom dataset, each achieving perfect accuracy. The Decision Tree is easy to understand and interpret, making it simple to see how decisions are made. On the other hand, the Random Forest is more robust and helps to avoid overfitting by combining multiple trees.

In real-world applications, the choice between these two models depends on what we need: if we want something easy to interpret,we should go with the Decision Tree. If we need more robust predictions, the Random Forest is the better option.

Overall, both these models are very effective for classifying data when the differences between classes are clear, like in the Mushroom dataset.

“END”