Machine Learning CS379: Unit 5 - Individual Project

Jacob Marquez

9/30/23

Colorado Technical University

Overview

- The Iris dataset is loaded, and then it is split into training and testing datasets.
- The CART algorithm is used for the Decision Tree Classifier.
- The classifier is then fitted with the training data and labels.
- Predictions are made on the test data.
- The accuracy of the model is calculated by comparing the predicted labels with the actual labels of the test data.
- The decision tree is plotted, showing the splits and the homogeneous branches.

Detailed Description

The Decision Tree Classifier decides the species of Iris flowers based on their features. The Iris dataset is a popular dataset in machine learning and pattern recognition, and it is included in the `sklearn` library for Python. It includes 150 samples of iris flowers, with 4 features each:

- sepal length (cm)
- sepal width (cm)
- petal length (cm)
- petal width (cm)

Each sample belongs to one of three classes (**Setosa**, **Versicolor**, or **Virginica**), which represent the species of the iris flowers.

Here's how the decision tree works in this example:

1. **Training**: During the training phase, the decision tree algorithm examines the features and their corresponding labels (species) and creates a model by finding the best questions to ask (split points) to segregate the data into one of the three classes. It continues this process, finding the best questions for the "child" nodes and so forth, until it can perfectly categorize the training data or until it reaches a predefined depth to avoid overfitting.

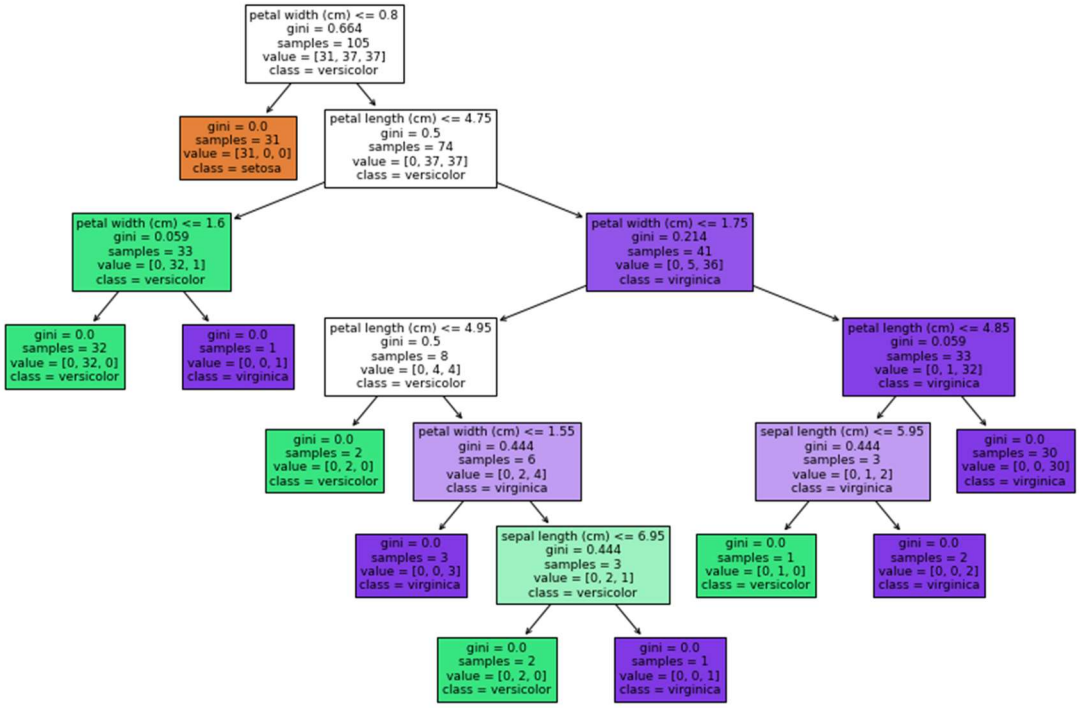
2. **Splitting Criteria**: The CART algorithm uses a metric called "Gini impurity" to find the best split at each node. It evaluates all the possible splits across all features and chooses the one that results in the lowest impurity in the child nodes.

3. **Prediction**: During the prediction phase, the algorithm uses the model (tree) it built during the training phase to classify new, unseen data. It takes the features of a new sample, traverses the tree by answering the questions at each node based on the sample's features, and ultimately assigns a class to the sample based on the leaf node it ends up in.

4. **Output**: The output in this case is the species prediction for the test set samples, and the decision tree plotted, which visually represents the decision-making process of the classifier.

In essence, the decision tree is making decisions on the species of iris flowers based on the four features mentioned above. The tree makes these decisions by asking a series of questions, each one intended to narrow down the possibilities until it can make a confident prediction.

Initial Results




```
In [2]: runfile('C:/Users/Jacob/Documents/School
CTU1/Classes/Machine Learning/Unit 5/
DecisionTreeClassifier.py', wdir='C:/Users/Jacob/
Documents/School CTU1/Classes/Machine Learning/
Unit 5')
Accuracy: 1.00
```

If a model achieves a perfect accuracy score of 1.00 on the test data, it could potentially be a sign of overfitting, especially if the model is complex and the dataset is small. Overfitting occurs when a model learns the training data too well, including its noise and outliers, leading to a loss in the model's ability to generalize from the training data to unseen data.

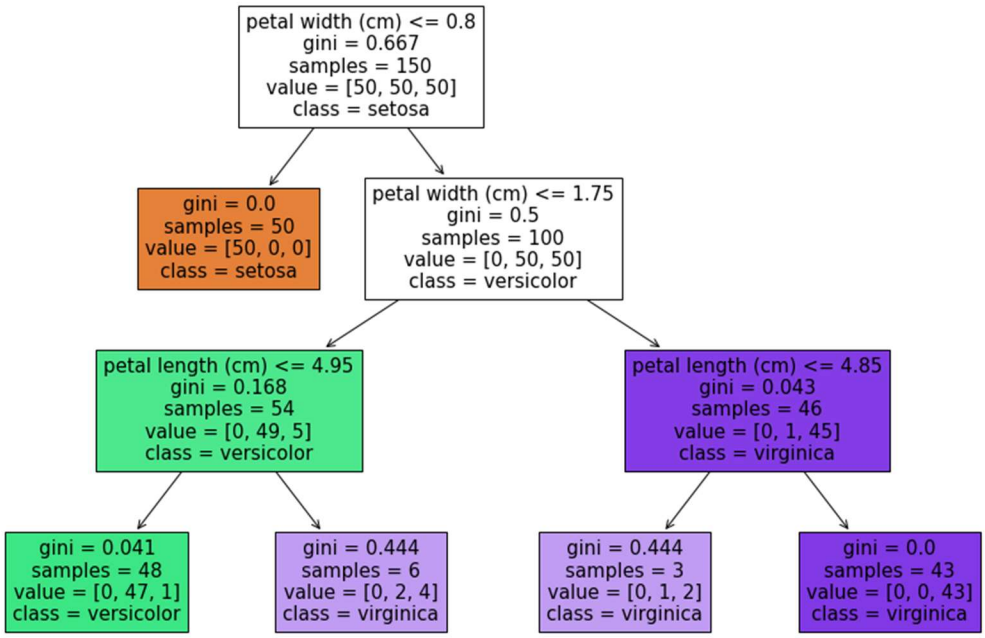
In the context of decision trees, a deep tree might capture noise in the training data, leading to overfitting. The tree perfectly classifies the training data but fails to generalize to new data, thus performing poorly on unseen data.

However, in the case of the Iris dataset, it is relatively small and clean, and the classes are quite separable by the features, so a high accuracy on the test set might not necessarily indicate overfitting. It's always a good idea to test the model on a separate validation dataset or use cross-validation to ensure the model's robustness.

Addressing Potential Overfitting

To enhance the model's performance on the Iris dataset and mitigate overfitting risks, cross-validation and tree pruning were implemented. Cross-validation ensures the model’s accuracy is consistent across different dataset subsets, providing a more reliable performance estimate. Limiting the tree's depth prevents the model from becoming overly complex and capturing dataset noise as patterns, improving its generalization to new data. These strategies collectively bolster the model’s predictive capability and real-world applicability.

Final Results




```
In [4]: runfile('C:/Users/Jacob/Documents/School
CTU1/Classes/Machine Learning/Unit 5/
DecisionTreeClassifier.py', wdir='C:/Users/Jacob/
Documents/School CTU1/Classes/Machine Learning/
Unit 5')
Accuracy: 0.96 (+/- 0.05)
```

This is a strong result! An accuracy of 0.96 with a small standard deviation +/- 0.05 indicates that the model is performing well and is relatively stable across different subsets of the dataset.

It's important to note that no model is perfect, and a 1.00 accuracy should be approached with caution. However, in this case, given the context of the provided dataset, the slightly less-than-perfect accuracy of 0.96 and a reasonable standard deviation suggest that the model is likely not overfitting and is generalizing well to unseen data.

The model's performance can be considered robust, especially with a dataset like Iris which is relatively simple. The decision tree, with a limited depth, can achieve high accuracy without being overly complex. Despite these results, it's always beneficial to try different algorithms and techniques to ensure that you are getting the best possible performance and not missing any important insights. Other techniques like random forests or gradient boosting machines (like XGBoost) could be explored as well to verify and compare the performance with the decision tree classifier.

References

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and Regression Trees. CRC press.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.

Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer.

Provost, F., & Fawcett, T. (2013). Data Science for Business. O'Reilly Media, Inc.

Tan, P.-N., Steinbach, M., & Kumar, V. (2006). Introduction to Data Mining. Pearson Addison Wesley.