

Documentation: Implementing and Testing a Decision Tree Classifier in Prolog for the IRIS Dataset

12.01.2024

Overview

This documentation covers the second part of the homework, which involves implementing a decision tree classifier in Prolog for the IRIS dataset. The classifier is designed to identify the species of an iris flower based on its sepal length, sepal width, petal length, and petal width.

The Prolog code implements a decision tree based on specific rules derived from the IRIS dataset. The decision tree's logic is translated into Prolog predicates, with each rule in the tree represented as a conditional in the **classify** predicate.

How to Run the Code

1. **Environment Setup:** Ensure you have a Prolog interpreter installed, such as SWI-Prolog.
2. **Code Loading:**
 - Start the Prolog interpreter.
 - Load the Prolog code containing the **classify** predicate. This can typically be done using the **[second_part.pl]** command in Prolog, where **second_part.pl** is the name of the file containing the code.
3. **Running Queries:**
 - To classify an iris flower, use the query format: **?- classify(SepalLength, SepalWidth, PetalLength, PetalWidth, Class).**
 - Replace **SepalLength**, **SepalWidth**, **PetalLength**, and **PetalWidth** with the respective measurements of the iris you want to classify.
 - The variable **Class** will be bound to the species name if the classification is successful.

Range Set of Each Argument

The typical ranges for the measurements in the IRIS dataset are as follows:

- **Sepal Length:** 4.3 to 7.9 cm
- **Sepal Width:** 2.0 to 4.4 cm
- **Petal Length:** 1.0 to 6.9 cm
- **Petal Width:** 0.1 to 2.5 cm

Test Cases

Here are some example test cases to validate the classifier:

1. Test Case for Iris Setosa:

- Query: `?- classify(5.0, 3.6, 1.4, 0.2, Class).`
- Expected Result: `Class = 'Iris-setosa'`

2. Test Case for Iris Versicolor:

- Query: `?- classify(6.0, 2.9, 4.5, 1.5, Class).`
- Expected Result: `Class = 'Iris-versicolor'`

3. Test Case for Iris Virginica:

- Query: `?- classify(6.3, 3.3, 6.0, 2.5, Class).`
- Expected Result: `Class = 'Iris-virginica'`

Conclusion

The implemented Prolog code provides a functional approach to classifying iris species based on the provided decision tree. It is important to test the code with various input ranges to ensure its robustness and accuracy in classification.

Salih Karagöllü

210104004069