Relatório sobre executar uma árvore de decisão em um microcontrolador

Metodologia:

O código é dividido em duas partes principais: o terminal serial feito com python e o código em C executado pelo microcontrolador. Abaixo explicarei em mais detalhes como cada um funciona.

O firmware do microcontrolador recebe via interrupção serial as 4 características do dataset Iris (comprimento e altura da sépala e pétala) e utiliza uma árvore de decisão que foi treinada previamente e convertida para C utilizando a biblioteca *emlearn*. A função eml_trees_predict_tree é chamada para executar a árvore e determinar a classe da flor através das características e o resultado é devolvido via serial.

Já o código em Python é responsável por conversar com o microcontrolador utilizando uma porta serial. O script envia os dados do dataset e aguarda 0.5 segundos pela resposta, registrando a classe resultante. O código então repete esse processo para as 150 amostras e compara a resposta do microcontrolador com o resultado do dataset.

Após executar o código foi obtida uma taxa de acerto de 94,67%:

```
PROBLEMS (5)
        TERMINAL
                PORTS
                                 OUTPUT DEBUG CONSOLE
        Sending inputs with index 149
        - feature[0]: 5.9
        - feature[1]: 3.0
        - feature[2]: 5.1
        - feature[3]: 1.8
(Cópia ...
               2.0
        All database checked
.py
        Checking accuracy
.py~
        Accuracy (train set): 94.666667
        _accur...
         1. 1. 1. 1. 2. 1. 1. 1. 1. 1. 2. 1. 1. 1. 1. 1. 1. 1. 1. 1.
eric
                                                2. 2. 2. 2. 2. 2.
l.py
         2. 2. 2. 1. 2. 2. 1. 1. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 1. 2. 2. 2. 2. 2.
         2. 2. 2. 2. 2. 2.]
        18:46:05 in material_de_aula_uc_tweber_revised_2024/python_terminals/pyt
```

O resultado foi bem satisfatório quando comparado aos 96,7% de taxa de acerto que obtive ao treinar uma árvore de decisão diretamente com python.