CSC459 Database Management Systems (Spring 2024)

Programming Assignment 3: Decision Trees

Goals for learning

In this assignment, we will:

1. Take a close look at building decision trees for classification problems.
2. Practice using the Pandas and NumPy libraries for data analysis.
    - Emphasize practice with the array-oriented paradigm.

Submission details

- When is it due: **Monday 05/6, 11:59 PM**
    - Late assignments will not be accepted after 11:59 pm Thursday 5/9
- What is provided to you:

    1. This assignment description.
    2. A training dataset (all_electronics_training.db).
    3. A test/evaluation dataset (all_electronics_test.db).
    4. The template for your script (decision_tree.py).

- What to submit:

    1. Your implementation of the decision tree algorithm (decision_tree.py).
    2. Text files containing the output from running your completed decision_tree.py.
        - "decision_tree_structure.txt" – Displays the structure of the trained decision tree.
        - "decision_tree_results.txt" – The results of evaluating the test data with your tree.
        - Both of these files are generated by the script template under your current working directory.

- Where to submit: Upload to the associated assignment in **Brightspace**
- As specified in the syllabus, a 5% penalty will be applied per day late.

About the data

- We will be using the "all electronics" customer data example from "Data Mining: Concepts and Techniques" (table 8.1) as our training set (all_electronics_training.db)
- Some additional rows/tuples are provided for the test set as a separate database file (all_electronics_test.db).

Instructions

1. Download the script template (decision_tree.py).
2. Download the dataset files (all_electronics_training.db and all_electronics_test.db)
    - By default, the script template looks in its current directory for the dataset files.

3. Modify the template with your solution:
    - See the section “About decision_tree.py” below for details.
    - For the following functions, **do not use recursion or iteration**. Instead, consider the Pandas and NumPy documentation for array-oriented alternatives:
        - CalculateGiniImpurity
        - CalculateGini
        - CalculatePerformance
4. Run the program: “python3 decision_tree.py”
    - Upload the modified “decision_tree.py” file to Brightspace.
    - Copy the output files to Brightspace.
        - “decision_tree_structure.txt”
        - “decision_tree_results.txt”
        - “agnes_complete_link.txt”

About decision_tree.py

- What you need to do:
    - Implement any steps marked with a “TODO” comment.
    - Do not use recursion or iteration where indicated (3 functions)
    - Make sure that your output files look reasonable to you.
- Implementation Hints:
    - You are not obligated to use the existing function or class outlines.
        - The template is provided as a suggestion. You are not required to use all provided parameters for each function.
        - Feel free to make modifications to function names and/or parameters.
        - The format of the output files should not be changed.
    - I have included unit tests (derived from the text book) to help you validate your Gini index calculations, which are the building blocks of this algorithm.
    - You will need to search the Pandas library documentation, especially the DataFrame documentation, to see what functions are available to you.
    - I found the following Pandas features to be especially helpful:
        - https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.isin.html
        - The tilde (‘~’) negation operator