CS 470: Data Mining

Homework III - Classification

Prof. Kai Shu
Due by March 24, 2025, at 11:59 PM

Submission Instructions: Submit your assignment through the Canvas system. Upload a single ZIP archive file named `hw3_<your first name>_<your last name>.zip`, containing:

1. A **code folder** including the executable source code.
2. A **README.txt** file including a description of input parameters, language, software, environment, and required libraries. Users should be able to run your code following the guidance in the README file. The Emory Honor Code should also be included:

   ```
   /* THIS CODE IS MY OWN WORK, IT WAS WRITTEN WITHOUT
   CONSULTING CODE WRITTEN BY OTHER STUDENTS. Your Name Here */
   ```

3. A **report** (in PDF or Word format) containing experimental results and discussions.

No email submissions are accepted. Include a section named **Collaboration Statement** at the top of your solution, acknowledging any collaboration, help, or resources used.

1 Part 1: Decision Tree Classifier for Heart Disease Prediction (70 points)

1.1 Problem Setup

The file `data.csv` contains a dataset of anonymized patient information with 14 attributes, where the last attribute, *"Has heart disease?"*, is the prediction target. All other attributes except *"person ID"* can be used as input attributes. Implement a decision tree classifier to predict heart disease, following the tasks below.

1.2 Coding (30 points)

Implement the Decision Tree Induction algorithm for classifier training and prediction in Python (`.py` or `.ipynb` format). The program should accept four string-type parameters:

- **Parameter 1:** Path to the input dataset file (`data.csv`).
- **Parameter 2:** Path to `para2_file.txt`, listing training set person IDs (one per line).
- **Parameter 3:** Path to `para3_file.txt`, listing test set person IDs (one per line).

- **Parameter 4:** Path to `para4_file.txt`, listing test set person IDs and corresponding prediction results (`yes` or `no`, space-separated).

Example files for Parameters 2, 3, and 4 are provided for format reference.

Hint: Start early. While studying existing decision tree implementations is allowed, copying them is not. An automatic plagiarism checker will be used.

2 Part 2: Writing (40 points)

2.1 Experiment Design

- Choose an impurity measure (e.g., Information Gain, Gini, Gain Ratio) and justify your selection.
- Decide how to split categorical and continuous attributes.
- Determine data structures (e.g., tree, dictionary, class-based implementation).
- Optimize your implementation where possible.

In the report, discuss:

1. Justification for impurity measure selection.
2. Attribute splitting strategy.
3. Data structure choice.
4. Additional techniques used, with references.

2.2 Model Evaluation

Select a model evaluation scenario (e.g., hold-out, cross-validation, bootstrap) and justify your choice. Evaluate your model using:

- Accuracy
- Precision
- Recall
- F-measure
- Specificity
- Sensitivity

Analyze these metrics and discuss any insights gained from the assignment.

Note: The `README.txt` file contributes 10 points to the total 40.

3 Part 3: Clustering Algorithms (30 points)

Write the pseudocode for the following clustering methods and explain the issues or limitations each algorithm addresses:

1. K-Means++
2. K-Medoids

4 Grading Criteria

- **30 points** for code implementation. TAs will run 10-fold cross-validation to evaluate accuracy and runtime.
  - -10 points for minor mistakes.
  - -20 points for substantial mistakes.
  - -30 points for major errors reducing accuracy below 60%.
  - **Zero points** if:
    - The code does not compile as per the `README.txt`.
    - The model achieves accuracy below 50%.
    - The runtime exceeds 15 minutes on a 2.3 GHz 8-Core Intel Core i9 CPU.
    - Directly using an existing decision tree library.
- **10 points** for `README.txt`.
  - -3 points if key statements are missing, making execution unclear.
  - -2 points for incorrect filenames.
  - -5 points for missing explanations.
- **30 points** for a well-organized report.
  - 15 points for correctness and reasonableness of results.
  - 15 points for sufficient discussion of results and their implications.
- **30 points** for pseudocode of clustering algorithms (15 points each).

Note: *10 points will be deducted for missing the Honor Code or Collaboration Statement (-20 if both are missing).*