CSCI316 – Big Data Mining Techniques and Implementation
Assignment 1
Autumn 2023

10 Marks
Deadline: Refer to the submission link of this assignment on Moodle

Two tasks are included in this laboratory. The specification of each task starts in a separate page.

You must implement and run all your Python code in Jupyter Notebook. ***The deliverables include one Jupyter Notebook source file (with .ipybn extension) and one PDF document for each task.***

To generate a PDF file for a notebook source file, you can either (i) use the Web browser's PDF printing function, or (ii) click "File" on top of the notebook, choose "Download as" and then "PDF via LaTex".

The submitted source file(s) and PDF document(s) must show that all of your code has been executed successfully. Otherwise, they will not be assessed.

This is an individual assessment. Plagiarism of any part of this assessment will result in having 0 mark for this assessment and for all students involved.

The correctness of your implementation and the clearness of your explanations will be assessed.

Task 1

(5 marks)
Dataset: **Dataset**: The Superconductivty data
Source: https://archive.ics.uci.edu/ml/datasets/Superconductivty+Data

There are two files in this dataset: (1) train.csv contains 81 features extracted from 21263 superconductors along with the critical temperature in the last column, (2) unique_m.csv contains the chemical formula broken up for all the 21263 superconductors from the train.csv file. This task uses *the first file*.

Objective
The objective of this task is to develop an end-to-end data mining project by using the Python machine learning library ***Scikit-Learn***. (Note. Scikit-Learning is the only machine-learning library allowed to use in this task. However, other non-ML libraries (e.g., SciPy) can be used.)

Requirements
(1) This is a *regression* problem. The task is to *predict critical temperature in the superconductivity data*.
(2) Use a stratified sampling method to sample (around) 2/3 and 1/3 records in the file train.csv for training and test, respectively.
(3) Main steps of the project should be (a) "discover and visualise the data", (b) "prepare the data for machine learning algorithms", (c) "select and train models", (d) "fine-tune the models" and (e) "evaluate the outcomes". You can structure the project in your own way. Some steps can be performed more than once.
(4) In the steps (c) and (d), you can choose any regression algorithm.
(5) Explanation of each step together with the Python codes must be included.
(6) A comparison of the regression models' performance must be included.

Deliverables
- A Jupiter Notebook source file named `your_name_task1.ipybn` which contains your implementation source code in Python
- A PDF document named `your_name_task1.pdf` which is generated from your Jupiter Notebook source file, and which includes clear and accurate explanation of your implementation and results.

Task 2

(5 marks)
Dataset: The Nursery Data Set
Source: https://archive.ics.uci.edu/ml/datasets/nursery

Nursery Database was derived from a hierarchical decision model originally developed to rank applications for nursery schools. There are 8 attributes and 1 target variable, all of which are categorical. The last target variable (i.e., the application ranking) includes 5 classes: "not_recom", "recommend", "very_recom", "priority" and "spec_prior".

Objective
The objective of this task is to implement *from scratch* a Decision Tree (DT) classifier to predict whether the application ranking. You cannot use any ML library for this task.

Requirements
(1) Implement three DT models by the split criteria of Information Gain, Gain Ratio and Gini Index. You can use either binary-split or multiple-split.
(2) Use ~60% data as training data and ~40% as test data
(3) It is recommended that your implementation includes a "tree induction function", a "classification function" and other functions (which are up to you).
(4) After implementing the three DT models, use a simple ensemble method to build a new classifier. This new classifier just utilises a voting function on the prediction outcomes of the three DT models.
(5) Present clear and accurate explanation of your implementation and results (in the Markdown format). In particular, report the accuracy of the models, and report whether an improvement the ensemble method can achieve.
(6) Note. You can (but not must) use any suitable pre-processing method. You also can (but not must) use any reasonable early stopping criteria (pre-pruned parameters such as number of splits, minimum data set size, and split threshold) to improve the training speed. If you do so, you must explain the criteria clearly.

Deliverables
- A Jupiter Notebook source file named `<your_name>_task2.ipybn` which contains your implementation source code in Python
- A PDF document named `<your_name>_task2.pdf` which is generated from your Jupiter Notebook source file.