CSCI316 – Big Data Mining Techniques and Implementation
Individual Assignment 2
2025 Session 3 (SIM)

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

Two (2) tasks are included in this assignment. 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.***

Note: 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".

All results of your implementation must be reproducible from your submitted Jupyter notebook source files. In addition, the submission must include all execution outputs as well as clear explanation of your implementation algorithms (e.g., in the Markdown format or as comments in your Python codes).

Submission must be done online by using the submission link associated with assignment 1 for this subject on MOODLE. The size limit for all submitted materials is 20MB. DO NOT submit a zip file.

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

Task 1

(7 marks)
Dataset: Weather Type Classification
Source: https://www.kaggle.com/datasets/nikhil7280/weather-type-classification/data

Objective
The objective of this task is to implement a Naïve Bayes classifier in Scikit-Learn to predict the weather type.

Task requirements

(1) Use *stratified sampling* to select ~80% for training and ~20% for test.
(2) Use one-hot encoding to transform the non-class categorical features into numbers.
(3) The Naïve Bayes API must be from Scikit-Learn. But you can use Numpy, Pandas and other non-ML libraries for preprocessing and visualisation purposes.

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 presents clear and accurate explanation of your implementation and results.

Task 2

Dataset: Weather Type Classification
Source: https://www.kaggle.com/datasets/nikhil7280/weather-type-classification/data

Objective
Develop an Artificial Neural Network (ANN) in TensorFlow/Keras to predict the weather type.

Requirements

(1) You can (but not must) use Scikit-Learn or other Python libraries to pre-process (e.g., scaling/normalisation). However, the ANN must be implemented within TensorFlow.
(2) You can use any ANN architecture (e.g., feedforward, CNN, etc.).
(3) The training process includes a hyperparameter fine-tunning step. Define a grid including at least three hyperparameters: (a) the number of hidden layers, (b) the number neurons in each layer, and (c) the regularization parameter for L1 and L2. Each hyperparameter has at least two candidate values. All other hyperparameters (e.g., activation functions and learning rates) are up to you.
(4) Use ~80% data for training and ~20% for test. Report the loss function values and classification accuracy for training and test.
(5) Present clear and accurate explanation of your ANN architecture and results.

Deliverables

- A Jupiter Notebook source file named `<your_name>_task_2.ipybn` which contains your implementation source code in Python
- A PDF document named `<your_name>_task_2.pdf` which is generated from your Jupiter Notebook source file.