# CSCI316 – Big Data Mining Techniques and Implementation

## Assignment 1

**Total Marks:** 15
**Due by:** 11:30pm Sunday April 20, 2025

This assignment consists of two independent tasks.

### General Instructions:

- Implement and run all Python code in a Jupyter Notebook.
- For each task, submit **one** Jupyter Notebook source file (with the extension .ipynb) and **one** PDF document generated from the notebook.
- Ensure all submitted code cells are executed, and outputs are visible in the PDF.
- This is an **individual assignment**. Plagiarism of any part of the work will result in a zero mark for all involved.
- Your work will be evaluated based on the correctness of your implementation, clarity of explanations, and quality of your analysis and visualizations.

### Task 1 – Classification with Custom Decision Trees (7.5 Marks)

**Dataset:**

- Name: Nursery Data Set
- Source: UCI Machine Learning Repository – Nursery Data Set
- Description: This dataset consists of 12,960 instances with 8 categorical attributes. The target variable represents an application ranking with five classes: “not_recom”, “recommend”, “very_recom”, “priority”, and “spec_prior.”

**Objective:**
Implement a decision tree classifier from scratch (without using any external machine learning libraries) to predict the nursery application ranking.

**Requirements:**

1. **Custom Decision Tree Implementation:**
   - Develop two decision tree models using different split criteria: one based on Information Gain and the other based on Gini Index.
   - Your implementation should include at least:
     - A tree induction function to build the decision tree.
     - A classification function to predict the class of new samples.
     - (Optional) Additional helper functions to calculate split metrics and manage stopping criteria.
2. **Data Splitting:**
   - Randomly split the dataset into 65% training data and 35% testing data.
3. **Ensemble Approach:**
   - After implementing the two decision tree models, build an ensemble classifier that combines their predictions using a weighted majority voting scheme.
   - Experiment with different weighting strategies and report on any observed improvements in overall classification accuracy.
4. **Evaluation and Reporting:**
   - Evaluate each model (both individual trees and the ensemble) using classification accuracy and a confusion matrix.

        - Include visualizations such as a textual or graphical representation of the tree structure and performance plots.
        - Using **Markdown** cells, document your code thoroughly and provide detailed explanations for your implementation choices, including any pre-processing, early stopping criteria, or pruning strategies employed.

**Deliverables:**

- A Jupyter Notebook source file named `<your_name>_task1.ipynb`
- A PDF document named `<your_name>_task1.pdf` generated from your notebook

**Task 2 – Regression Analysis on a Large-Scale News Popularity Dataset (7.5 Marks)**

**Dataset:**

- Name: Online News Popularity
- Source: UCI Machine Learning Repository – Online News Popularity
- Description: This dataset comprises 39,797 instances with 61 columns. The target variable is the number of shares an article received, while the remaining columns consist of various features related to the article's content, publication time, and other metadata.

**Objective:**
Develop an end-to-end data mining project to predict the popularity of online news articles (measured by the number of shares) using the Online News Popularity dataset.

**Requirements:**

1. **Problem Setup:**
    - This is a regression task where your goal is to predict the number of shares based on the provided article features.
2. **Data Splitting and Validation:**
    - Perform a random split of the dataset into 70% training data and 30% test data.
    - Additionally, implement 5-fold cross-validation on the training set to assess model robustness.
3. **Project Steps:**
    - **Data Exploration and Visualization:**
        - Provide summary statistics and distribution plots for a selection of features: choose 5 highly relevant features, 5 moderately relevant features, 5 less relevant features, and include the target variable.
        - Include visualizations such as correlation heatmaps and scatter plots to identify key relationships between features and the target variable.
    - **Data Preprocessing:**
        - Address any missing values if present, perform feature scaling, and consider feature selection or dimensionality reduction given the number of features.
    - **Model Selection and Training:**
        - Implement at least two regression models using the Scikit-Learn library.
        - Explain the rationale behind your model choices.
    - **Hyperparameter Tuning:**
        - Use grid search or another tuning method to optimize model hyperparameters.
    - **Evaluation and Analysis:**
        - Evaluate model performance using metrics such as RMSE, MAE, and $R^2$.
        - Provide visualizations like predicted vs. actual plots and error distribution histograms.

        - Discuss insights gained from feature importance analysis (especially if tree-based models are used).

4. **Explanation and Reporting:**
    - Clearly explain every step with corresponding code comments and **Markdown** documentation.
    - Compare the performance of the different models and discuss the benefits and drawbacks of each approach.

**Deliverables:**

- A Jupyter Notebook source file named `<your_name>_task2.ipynb`
- A PDF document named `<your_name>_task2.pdf` generated from your notebook