CSE 643
Homework Assignment 4:

Objective

This assignment aims to help you gain practical experience in benchmarking different machine learning approaches — both traditional supervised models and modern in-context learning via large language models (LLMs). You will compare model performance across multiple datasets, perform systematic hyper-parameter tuning, and analyse the trade-offs between different learning paradigms.

Project Description

You should select three datasets from the UCI Machine Learning Repository and train, tune, and evaluate three classical machine learning models: **Naive Bayes**, **Decision Tree**, and **Multi-Layer Perceptron** (MLP). After benchmarking these models, you should also implement a fourth approach using **In-context Learning** with an LLM (e.g., GPT-4, Claude, Gemini, etc.). The LLM should be prompted with a subset of labeled training examples directly in its context window and then asked to predict labels for given test examples.

Tasks and Requirements

1. Dataset Selection

• Choose three datasets from the UCI Repository that vary in size and complexity (e.g., one small, one medium, one large). Perform data preprocessing such as handling missing values, encoding categorical features, and normalising or standardising features as appropriate.

2. Model Training and Tuning

For each dataset, train and tune the following models using scikit-learn or an equivalent framework: Naive Bayes, Decision Tree, Multi-Layer Perceptron (MLP)

Use k-fold cross-validation ($k \geq 5$) for hyper-parameter tuning. Report the best hyper-parameters and validation metrics (accuracy, precision, recall, F1-score, etc.). Also include training curves (training/validation loss and accuracy vs. epoch) for the MLP.

3. In-Context Learning with LLM

Select an LLM (e.g., OpenAI GPT, Anthropic Claude, Google Gemini, etc.) and construct prompts that include a few-shot context — examples of input features and their correct outputs from the training data — and then ask the model to predict outputs for test examples. Evaluate the LLM's performance on the same test split as the other models, and discuss the prompt design and limitations / benefits of ICL vs traditional ML.

4. Evaluation Metrics and Comparison

Use consistent metrics (e.g., accuracy, F1-score, confusion matrix) and compare performance across models (Naive Bayes vs. Decision Tree vs. MLP vs. LLM) as well as datasets (small vs. large, structured vs. semi-structured, classification vs regression).

Discuss model interpretability, training time, computational cost, robustness, generalisation, and ease of implementation.

5. Deliverables

Each student must submit:

1. Code (e.g., Python files or Jupyter notebooks), clean, well-documented, including data preprocessing, training, tuning, and evaluation.
2. Report (PDF deck) — summarising datasets, methods, metrics, and results with tables, plots, and insights.
3. Video presentation (8–10 minutes) presenting methodology, findings, key code snippets, and conclusions.

As earlier, **presentations can be done on ICAPP or otherwise; however recording link, pdf and code files need to be submitted via Google classroom.**