HW2: Classification Model Implementation

Design and implement your own machine learning model to predict a student's final grade in a particular course. Experiment with different machine learning algorithms for this task and report only the best-performing model you can get. It is highly recommended that you build this model using scikit-learn library.

Task Details

You are presented with students' data (xAPI-Edu-Data.csv) and you are asked to design and implement a classification model that can predict the performance of students (Low-Level, Middle-Level, High-Level) as defined below. Experiment with different machine learning algorithms, and report only the best performing model you can get for this task. Evaluation results of your best model must be using a percentage split methodology on the provided data (80% for training and 20% for testing with random_state=91). More info about the train_test_split method is found here:

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

Data Information:

The dataset consists of 480 student records (number of instances) and 16 features (number of attributes). The features are classified into three major categories:

(1) Demographic features such as gender and nationality.
(2) Academic background features such as educational stage, grade level, and section.
(3) Behavioral features such as raised hand-on class, opening resources, answering surveys by parents, and school satisfaction.

The data set also includes the school attendance and parent participation features.

The students are classified into three numerical intervals based on their total grade/mark (the "**Class**" column):

- Low-Level (L): interval includes values from 0 to 69
- Middle-Level (M): interval includes values from 70 to 89
- High-Level (H): interval includes values from 90-100

To download and get more information about the data:

https://www.kaggle.com/aljarah/xAPI-Edu-Data

Submission Instructions:

After you finish your work, submit **two** files in the LMS (a **PDF** report & a **ZIP** archive for the source code) in the LMS (Blackboard), and then fill in the provided form. Follow these instructions:

1. Submit **two** files in the LMS (a **PDF** report & a **ZIP** archive for the source code) as the following:
   1.1. A **PDF** report that describes all the steps you took to design and implement your models along with the following information:
      1.1.1. Your name and student ID
      1.1.2. The IDE you used
      1.1.3. The libraries you used to implement the models
      1.1.4. Description of the chosen algorithms including all the selected hyperparameters
      1.1.5. Description of all data preprocessing you applied to the provided dataset
      1.1.6. Description of all other steps you have done to get your final results
      1.1.7. Evaluation results of your models using a percentage split methodology (80% for training and 20% for testing with **random_state=91**) and report the following results:
         - Classification accuracy
         - F1 Score (Macro Averaged)
         - Confusion Matrix (Optional)
      1.1.8. Screenshots of **all the code** & **evaluation results** from the platform you chose
   1.2. A **ZIP** compressed file containing all the source code of your models (i.e., Jupyter Notebook files)
2. Fill in this online form after you submit the aforementioned deliverables:
   - https://forms.gle/Jtb1eYFGBWWdj1AZ7

Important Notes:

- Don't forget to fill the provided Google form after you submit the required files in the LMS.
  - https://forms.gle/Jtb1eYFGBWWdj1AZ7
- Make sure to submit the two files **separately** (do not include the **PDF** report inside the **ZIP** archive).
- The report file must be in **PDF** format. Don't use other formats such as doc, docx, odt, or txt.
- The compressed file containing your source code must be in **ZIP** format. Don't use other formats such as RAR, 7s, or tar.
- Provide screenshots of all the code, not just some parts.