Project Title: Adaboost for Medical Image Classification

Objective: The goal of this machine learning project is to utilize the Adaboost algorithm to enhance the accuracy of medical image classification. Adaboost is an ensemble learning technique that combines multiple weak classifiers to create a strong classifier. In this project, we will focus on using Adaboost to classify medical images from a given dataset.

Dataset: You can use a popular medical image dataset such as the MedMNIST. The dataset contains multiple subsets with images from different organs acquired with different modalities. You can focus on only a couple of them (you can choose them).

Tasks:

1. Data Preprocessing:
    - Load and preprocess the medical image dataset.
    - Apply any necessary preprocessing steps such as resizing, normalization, or data augmentation.
2. Implement Weak Classifiers:
    - Choose a simple classifier (e.g., decision stump, shallow decision tree) as the weak learner.
    - Implement and train multiple instances of this weak classifier on different subsets of the dataset.
3. Adaboost Algorithm:
    - Implement the Adaboost algorithm by iteratively adjusting the weights of misclassified samples and combining the weak classifiers.
    - Experiment with different variations of Adaboost, such as the original Adaboost algorithm and its variants like SAMME (Adaboost.M1) or SAMME.R (Adaboost.M2).
4. Ensemble Model Training:
    - Train the Adaboost ensemble using the implemented weak classifiers.
    - Monitor the changes in classifier weights during each iteration to ensure the boosting process is working correctly.
5. Model Evaluation:
    - Evaluate the Adaboost ensemble model on a separate validation dataset.
    - Measure classification metrics such as accuracy, precision, recall, and F1-score.
6. Comparison and Analysis:
    - Compare the performance of the Adaboost ensemble model with that of individual weak classifiers.
    - Analyze how Adaboost's accuracy changes over iterations and whether it converges.
7. Bonus: Hyperparameter Tuning
    - Experiment with different hyperparameters such as the number of weak classifiers and learning rate.
    - Use techniques like cross-validation to find the optimal hyperparameters.

Resources:

- Python with libraries like NumPy, SciPy, Scikit-learn, and Matplotlib for implementation and visualization, and handling image data
- MedMNIST dataset, which can be installed using pip.

Deliverables:

Does not apply