In this homework, you will implement a Boosting algorithm to conduct classification on the medical image tampering and accent recognition datasets. You will compare several methods to train a boosting model.

1 Build Boost Data Structure

Each weak learner needs to perform just slightly better than average. There are many methods that can do this, so we'll set up a structure to handle all of them.
You'll create a Booster class with the following functions and behaviors:

- **__init__**:
  :param n_estimators: The number of estimators that will be used as Weak Learners. Default=500
  :type n_estimators: int
  :param learning_rate: Modifies how strong the gradient is. Default=1
  :type learning_rate: float
  :param random_state: Set random state for replicable results. Default=None
  :type random_state: int
  :param base_learner: Learner used as the framework for Weak Learners
  :type base_learner: Sklearn object

  Behavior: Sets attributes appropriately. Set self.estimators to empty list, self.alphas to np array of zeros. Set self.accuracies and self.losses to None.

- **plot** (optional)
  Behavior: plots the accuracy and losses after each Weak Learner is assigned and display plots.

- **fit**
  :param X: X data
  :type X: numpy array
  :param Y: Y data
  :type Y: numpy array
  :param batch_size: Size of batches for stochastic performance
  :type batch_size: int

  Behavior: Performs multiclass Adaboost operations explained in lecture
  NOTE: the loss for multiclass Adaboost is

  $$\sum_{i=1}^{K} \exp(-1/K* < y_k, f_k >)$$

  Where y_k is $\dfrac{-1}{K-1}$ where $f = Y$, else 1
  Where f_k is the predictions associated to the class k

- **predict**
  :param X: X data to predict
  :type X: numpy array

  Behavior: use each Weak Learner to predict from X. Multiply each prediction by its associated alpha, then sum the predictions. The prediction for each data point is the index with the largest value.

2 Test Framework

Now that we have the Boost structure, we can test the performance of boosting on a couple different structures. We will try both general boosting and stochastic boosting. Note that stochastic boosting is done by taking a random permutation our data set of size m. This is process is very similar to our normal minibatch process.
For our datasets, we will use medical image tampering and accent recognition.

Using Gaussian Naive Bayes

Use sklearn.naive_bayes.GaussianNB as the model for our WeakLearner. Train the model on Xtrain, then run it through Boost, both regularly and stochastically and visualize the performance and results.

Using Shallow Decision Trees

Use sklearn.trees.DecisionTreeClassifier with max_depth=2 as the model for our WeakLearner. Train the model on Xtrain, then run it through Boost, both regularly and stochastically and visualize the performance and results.

3 What to Turn In

For this homework, you will need to turn in the following:

- Jupyter Notebook
- PDF with graphs
  - Graph comparing Gaussian naive Bayes general boosting and stochastic boosting.
  - Graph comparing Shallow Decision Trees general boosting and stochastic boosting.
- Compare and contrast the behaviors of models that you generated graphs for. Be sure to add any notes that you observed during the training of the data.