Machine Learning Assignment - 2

Instructions.

- **This assignment will help you understand importance of data, EDA and Feature selection**
- **Use `#.py` file for this assignment. Not doing so will lead to (0 Marks) in complete assignment.**
- **Each question has its own instruction, follow precisely. If not (0 Marks) will be awarded.**
- **Submission Guidelines**
  - **Submit only a single [MLA2_*.py] where * is your roll number.**
  - **For eg: if your roll number is [MT10002], your file name should be [MLA2_MT10002.py]**
- **Not following naming convention will lead to (0 Marks) , whatever the reason you present.**
- **You are allowed to use only [*pandas, random, numpy, PIL, matplotlib, scipy, SciKit-learn*] for this assignment. Again not following the same will lead to (0 Marks) for entire assignment.**
- **If you are not able to explain in viva, irrespective of your answer correctness you will get (0 Marks)**
- **If there is 🔴 in front of the question, you have to implement it completely from scratch, and if it is 🟢 you can use functions from the above-mentioned packages.**

[30 Marks]

Dataset Link

Dataset Description:

1. **IMAGE**: This is a small dataset of some cartoon characters. Each example is a 28x28 RGB image, associated with a label from 10 classes.
2. **AUDIO**: This is a subset of a publically available dataset of crowdsourced recordings of laughter, sighs, coughs, throat clearing, sneezes, and sniffs from multiple speakers.
3. **TEXT**: This contains structured narratives, each consisting of seven sentences labeled according to their respective function in two classes `moral` and `immoral`

Students who have opted for alternate evaluation policy have to do `Beginner` and `Intermediate` questions on all three modalities of data. Rest all other students have to do the whole assignment on any two data modalities of their choice.

- **Beginner**
  1. 🔴 Data pre-processing makes the data suitable for the algorithms/models to work on, perform on randomly selected 4 data samples:
     (a) **(1 Marks)** Image Dataset: Load image using `PIL`, convert grayscale, normalize image data between $[-1, 1]$ and plot color vs grayscale image, unnormalized vs normalized image.
     (b) **(1 Marks)** Audio Dataset: Load audio using `Scipy`, normalize audio data between $[-1, 1]$ and plot unnormalized vs normalized audio.
     (c) **(1 Marks)** Text Dataset: Create a dictionary to incorporate all possible characters with keys as characters and values as numbers, and tokenize the data and print sentance and tokenized sentance.
  2. 🔴 Sampling is a process of selecting data samples in a specific order. Implement these sampling strategies.
     (a) **(1 Marks)** **Sequential Sampler:** Define a order based on number of samples in each class, if number of samples are same rearrange the order based on classes alphabetically. Implement a sampler to select data in this pre-defined order and run the sampler to select and print complete data in the sampled manner.

(b) **(1 Marks) Weighted Random Sampler:** Give weight to each class based on there similarity and implement a sampler which selects classes which are least similar. Make sure sampler do not have information about similarity of classes and run the sampler to select and print complete data in the sampled manner.
   - Image Similarity: Cosine Similarity of GrayScale Histogram.
   - Audio Similarity: Cosine Similarity of Loudness of signal
   - Text Similarity: Cosine Similarity of tokenized sentence vector.

(c) **(1 Marks) Distributed Sampler:** Split the dataset in `n` sets randomly based on classes and implement a single sampler to sample from these sets. Make sure sampler do not have information about the data in differnet sets and run the sampler to select and print complete data in the sampled manner..

3. 🟢 Randomly select two samples from data and implement mentioned feature extraction techniques on given datasets. (You have to only use `scipy, matplotlib` for this question.)
   (a) **(1 Marks)** Image Data: Features from accelerated segment test (FAST), Scale-Invariant Feature transform (SIFT)
   (b) **(1 Marks)** Audio Data: Mel-Frequency Cepstral Coefficients (MFCC), Linear-Frequency Cepstral Coefficients (LFCC)
   (c) **(1 Marks)** Text Data: Term Frequency-Inverse Document Frequency (TF-IDF)

   Plot all features of all modality for selected samples
4. 🔴 **(1 Marks)** Develop a DataLoader class that accepts the following inputs: a folder path, a sampling strategy name, and the number of samples. The class should automatically identify the data type and perform sampling based on the specified strategy. It will then normalize the data as outlined in Question 1, apply feature extraction as described in Question 3(a,b,c), and return the extracted features for all samples with there labels which is suitable for training.

- **Intermediate**

1. 🔴 🟢 **(10 Marks)** Create a pipeline from scratch that takes a sampling strategy as input and performs the following steps:
   (a) Load data from the disk.
   (b) Sample the data according to the specified strategy.
   (c) Apply relevant feature extraction (of your choice).
   (d) Train the following models using the features extracted from the sampled data:
      i. Linear Regression
      ii. Ridge Regression
      iii. Lasso Regression

   Compare the performances of these models and provide observations on their differences. Use the scikit-learn library only for the model implementations; the rest of the pipeline must be written from scratch. Marks will only be awarded if you achieve an accuracy of greater than 65% on each model.

- **Advanced**

1. 🔴 🟢 **(10 Marks)** Create a pipeline from scratch that performs the following steps:
   (a) Load data from the disk.
   (b) Sample the data with combination of `Weighted Random, Distributed` strategy.
   (c) Apply relevant feature extraction (of your choice).
   (d) Train any two of the following models using the features extracted from the sampled data:
      i. Logistic Regression & KMeansClassifier
      ii. Decision Trees
      iii. Polynomial Regression (degree >2)

   Compare the performances of implemented models and provide observations on their differences. Use the scikit-learn library only for the model implementations; the rest of the pipeline must be written from scratch. Marks will only be awarded if you achieve an accuracy of greater than 70% on each model.

Further baby instructions for questions.

- Submission
  - For submission, you have to submit a single .py file. For each question where the plot is asked use ***plt.waitforbuttonpress(0)*** to display it indefinitely and close after a button press.
- Beginner
  1. For loading **Images** use ***PIL.Image.read()***, and for **Audio**, use ***scipy.io.wavefile.read()*** for **Text** use ***json.loads()***. Except for these, everything must be implemented from scratch. Here, from scratch, you cannot use any function from any library, not even the pre-defined functions that Python provides. For traversing through the data you can only use ***os*** library.
  2. Here each and everything you have to implement from scratch. However you can use only ***map()*** and ***iter()*** from pre-defined Python functions.
  3. For this question, use above mentioned methods to load data. For feature extraction, you can only use whatever ***scipy*** can offer. You can use random implementation from ***random*** module only for selection.
  4. Again implement everything from scratch you can use ***map()*** and ***iter()*** only. For loading the dataset use what is mentioned above. Use of ***if-else-elif*** statements is prohibited.
- Intermediate & Advanced
  1. You have to code each and everything in pipeline from scratch. You can use above mentioned methods for loading image, audios and text data. For model selection and execution, you can only use ***model.fit()***, ***model.fit_transform()*** and ***model.predict()***, and for feature extraction you can use any thing that scikit-learn has to offer. You are not allowed to use any other library. [**scikit-image, scikit-sound, and any other sister libraries of scikit are not allowed**]