CSE343/ECE343: Machine Learning
Assignment-2 Naive Bayes, Decision Trees, Random Forests
Max Marks: 25 (Programming: 15, Theory: 10) **Due Date**: 29/09/2024, 11:59 PM

Instructions

- Keep collaborations at high-level discussions. Copying/Plagiarism will be dealt with strictly.
- Late submission penalty: As per course policy.
- Your submission should be a single zip file **2020xxx_HW1.zip** (Where *2020xxx* is your roll number). Include **all the files (code and report with theory questions)** arranged with proper names. A single **.pdf report** explaining your codes with results, relevant graphs, visualization and solution to theory questions should be there. The structure of submission should follow:

  2020xxx_HW2
  |– code_rollno.py/.ipynb
  |– report_rollno.pdf
  |– (All other files for submission)

- Anything not in the report will **not** be graded.
- Remember to **turn in** after uploading on Google Classroom. No excuses or issues would be taken regarding this after the deadline.
- Start the assignment early. Resolve all your doubts from TAs in their office hours at least **two days before the deadline.**
- Your code should be neat and well-commented.
- **You have to do either Section B or C.**
- **Section A is mandatory.**

---

1. (10 points) **Section A (Theoretical)**

   (a) (3 mark) An investment firm is analyzing the likelihood of companies issuing dividends based on last year's percentage profit. After reviewing extensive data, the following insights were noted:

   Companies that issued dividends last year tended to have an average profit increase of 10%, whereas companies that did not issue dividends typically had no significant profit change. The variation in profit among all companies was observed to be consistent and symmetric, with a spread of 36% around their respective averages. On average, 80% of companies opt to issue dividends. If a company reported a 4% profit increase last year, estimate the likelihood that it will issue a dividend this year, considering the above trends.

Class Time	**Had Proper Sleep**	**Weather**	**Attended ML Class**
Morning	YES	COOL	YES
Morning	NO	RAINY	NO
Morning	NO	COOL	YES
Morning	YES	HOT	YES
Noon	YES	COOL	YES
Noon	NO	HOT	NO
Noon	NO	COOL	NO
Noon	YES	HOT	YES
Afternoon	YES	COOL	YES
Afternoon	NO	RAINY	NO
Afternoon	NO	HOT	NO
Afternoon	YES	HOT	YES

(b) (2 mark) Build the decision tree for the following table using Information Gain:

(c) (2 mark) Given a sequence of labeled examples S that is linearly separable by a margin γ, where γ represents the minimum margin across all examples, a special type of Perceptron algorithm is applied. The algorithm initializes the weight vector using the first example and makes predictions based on whether the margin condition is met. It updates the weight vector if there is a classification mistake or a margin mistake, where a margin mistake occurs if the current hypothesis classifies an example with a margin less than $\gamma/2$.

Assuming all examples are normalized to have a Euclidean length of 1, we need to prove that this special type of Perceptron algorithm will halt after a number of updates (mistakes) that is polynomial in $1/\gamma$. Specifically, we demonstrate that the number of updates (including margin mistakes) is at most $\frac{8}{\gamma^2}$.

Additionally, we will consider the case where the margin threshold $\gamma/2$ is replaced by $(1-\epsilon)\gamma$ and derive the corresponding mistake bound.

(d) (3 mark) You are building a spam filter using the Naive Bayes algorithm with a small dataset of labeled emails. Each email is represented by a feature vector indicating the presence (1) or absence (0) of two words: "buy" and "cheap". Your task is to classify a new email containing the word "cheap" but not "buy". Given the following training data:

Email	**Word "buy" (Feature 1)**	**Word "cheap" (Feature 2)**	**Spam (label)**
1	1	0	1
2	1	1	1
3	0	1	0
4	1	0	0

Tasks-

(a) Calculate the probability estimates for each feature given the class label (spam and non-spam).

(b) Compute the posterior probabilities for the new email (with "cheap" but not "buy") being spam or non-spam.

(c) Identify the problem with zero probabilities and suggest methods to address it.

2. (15 points) **Section B (Scratch Implementation)**
About the **Dataset**: The dataset contains one-second .wav audio files, each with a single spoken English word. There are 105,829 audio files organized into folders by the word spoken. The audios are collected through crowdsourcing, with contributors recording words in various environments, which are converted and stored in a wave file at a 16000 sampling rate. There is also a background_noise folder with audio clips for training models in noisy environments.
Tasks:

1. (4 points) Perform extensive exploratory data analysis (EDA) by performing the below-given sub-tasks:
   (a) Find the statistical summary of the amplitude values and duration distribution of the audio files for each class. Write your observations.
   (b) Plot graphical representations such as waveform plots, spectrograms, Mel-spectrograms, etc for randomly selected 3 audio files from randomly selected 3 classes out of 35 classes. Write your observations.
   (c) Investigate any class imbalances in the dataset, and if present, handle them through techniques like oversampling, undersampling, or data augmentation.
   (d) Perform data cleaning by detecting and addressing outliers like silent segments etc.
2. (3 points) Perform Feature Extraction: Extract relevant features from the audio data, such as MFCCs, chroma features, spectral features, and features extracted using filters like bandpass filter, etc.
3. (3 points) Model Selection and Implementation:
   (a) You are allowed to use only any one or ensemble of Naive Bayes, Decision Tree, Random Forest, and Perceptron models, all implemented from scratch. Train and test these models on the prepared dataset using an 80:20 train-test split.
   (b) Evaluate the selected models using accuracy as the metric. Document the results and compare the performance of different models. (Note: The use of libraries like scikit-learn, PyTorch, etc., is strictly prohibited for this task. All steps must be implemented from scratch.)
4. (5 marks) Final Evaluation: Submit your notebook containing all code and observations, along with a **form** reporting the best accuracy you achieved. Save your model using a library like pickle, and during the demo, you will be required to load the file and demonstrate the same results to the TA. Note that this task has a relative scoring system: the student with the highest accuracy (rank 1) will secure 5 marks for this part, while a student ranked 100th will receive

$$\left(1 - \frac{99}{\text{TotalNumOfStudentsInSectionA}}\right) \times 5$$

marks, and so on.

$$\text{Marks in part 4} = \left(1 - \frac{\text{Rank} - 1}{\text{TotalNumOfStudentsInSecA}}\right) \times 5$$

Note: You are only allowed to use libraries like librosa only for loading the audio. All other functions must be implemented from scratch using basic libraries like numpy, pandas, matplotlib, etc. Strict penalties will be imposed otherwise.

3. (15 points) **Section C (Algorithm implementation using packages)**
About the **Dataset**.: The dataset contains about 12k+ images for 15 different classes of human activities. Along with the data there is a csv file label.csv containing the label for each image. This image dataset involves labeling human actions to understand human behavior. There is only a single label attached to every image.
Task:

   1. (3 marks) Perform EDA:
      (a) Provide an overview of the dataset, including the number of images per class, the distribution of image sizes, and any other relevant statistics. Write your observations.
      (b) Create visualizations to represent the distribution of classes, and display a few sample images from each class. Write your observations.
      (c) Investigate the class distribution and discuss whether there are any class imbalances. If imbalances are present, propose strategies to address them, such as data augmentation or resampling techniques.
   2. (3 marks) Perform Feature Extraction: Extract relevant features from the images such as Histogram of Oriented Gradients (HOG), color histograms, or other traditional feature extraction methods. Write your observations.
   3. (2 marks) Model Selection and Implementation:
      (a) You are allowed to use only any one or ensemble of Naive Bayes, Decision Tree, Random Forest, and Perceptron models. Train and test these models on the prepared dataset using an 80:20 train-test split.
      (b) Evaluate the selected models using accuracy as the metric. Document the results and compare the performance of different models.
   4. (7 marks) Final Evaluation: Submit your notebook containing all code and observations, along with a **form** reporting the best accuracy you achieved. Save your model using a library like pickle, and during the demo, you will be required to load the file and demonstrate the same results to the TA. Note that this task has a relative scoring system: the student with the highest accuracy (rank 1) will secure 7 marks for this part, while a student ranked 100th will receive

$$\left(1 - \frac{99}{\text{TotalNumOfStudentsInSectionA}}\right) \times 7 \text{ marks}$$

In this part, if the student achieves less than 30 percent accuracy, then 0 marks.

$$\text{Marks in part 4} = \left(1 - \frac{\text{Rank} - 1}{\text{TotalNumOfStudentsInSecA}}\right) \times 7$$

Note: You are allowed to use any library functions for EDA and preprocessing. You are also allowed to use scikit-learn for train-test split as well as model implementation. Use of any deep learning module for any feature extraction or model implementation is strictly not allowed.