CSE343/ECE343: Machine Learning
Assignment-2 Decision Trees, Random Forest & Perceptron
Max Marks: 25 (Programming: 15, Theory: 10) **Due Date**: 05/10/2022, 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_HW2.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 pandemic broke in the city of Cranberry Melon. A survivor of the pandemic can die in 3 days if not given surgery. The survivor might live for 30 days post-surgery. The probability that he does not survive the surgery is 0.2

   (a) (2 marks) Draw a decision tree. Illustrate all probabilities and outcome values.

   (b) (1 marks) L(x) denotes the patient's living function, where x represents the number of days he will survive. Assuming that L(30) = 1.0 and L(0) = 0, how low can the patient's utility for living 3 days and still have the surgery performed? For the rest of the problem, assume that L(3) = 0.8.

(c) (2 marks) The doctor of Melon city finds a low-risk test procedure that predicts whether or not the patient will survive the operation. If the test is positive, the probability of surviving the surgery increases. The test has the following characteristics:

- True-positive rate: If the patient will survive the surgery, the probability that the results of this test will be positive is 0.95.
- False-positive rate: If the patient will not survive the operation, the probability that the results of this test will be positive is 0.05.

Find the survivor's probability of having a successful surgery if the test is positive

(d) (2 marks) Should the surgery be performed if the result of the test is positive?

(e) (2 marks) Recently, it has come to the chief surgeon's notice that there is a fatal consequence to the test, i.e., the patient may contract a new disease during the test. Draw a decision tree showing all options and consequences.

(f) (1 mark) Suppose that the probability of contracting the new disease during the test is 0.005. Should the test be conducted prior to operation? Draw a decision tree.

2. (15 points) **Section B (Scratch Implementation)**

1. (2 marks) Create a dataset (with 10,000 points) using the circle equation

$$(x-h)^2 + (y-k)^2 = r^2$$

such that: h=0, k=0, r=1, with the label **0** and h=0, k=3, r=1, with the label **1**.

- Write a class named dataset which takes number of points as input.
- The class should have a function named get(add_noise=False), which should give a set of pre-defined number of points. Every call to this function returns random points, given it satisfies the conditions above. DO NOT implement this class in the main .py/.ipynb file instead create a separate utils.py and import the functions you need in main file.
- Given, add_noise=True, as an input to the get function, it should also add Gaussian noise with mean 0 and standard deviation 0.1.

2. (1 mark) Plot the dataset on a 2-D plot such that all the information related to the dataset i.e., x, y, and labels can be inferred from the 2d plot itself with add_noise argument set to True and False.

3. (5 marks) Train a classifier using the Perceptron training algorithm (PTA), taught in class, on the data you just created (with and without noise) and plot the decision boundary if there exists one. Otherwise explain why not a decision boundary exists. NOTE: You have implement the PTA algorithm as a python class yourself from scratch using only numpy and python. DO NOT implement this class in the main .py/.ipynb file instead implement this in the utils.py and import the class/functions you need in main file.

4. (3 marks) Train another classifier using the perceptron training algorithm (PTA) on the data you just created (without noise) but with a fixed bias equal to "0" and plot the decision boundary if there exists one. Compare the results with question 2.3 and write a brief report of at least 150 words.
5. (3 marks) Create a dataset (with 4 points) using the XOR, AND, and OR property. Plot decision boundary, if there exists one, using the PTA such that the bias is learnable, and fixed (equals to "0")
6. (1 mark) Given a hyperplane equation and a point how would you predict which class (0 or 1) it belongs to? Also write any assumption you made, any equations you use for explanation.

3. (15 points) **Section C (Algorithm implementation using packages)**

In this question, you are expected to understand and run Random Forests and various boosting algorithms. You can use sklearn implementation of decision tree and Adaboost but you need to perform ensembling on your own.

Dataset: **Bitcoin Heist Ransomware Address Dataset**
Target Variable: Label
You will have to handle null values in the data. Split the data into a training, validation, and testing set (70:15:15 ratio) using the custom-designed train test split method. Use the same training set for training the following models. (You can not use sklearn for splitting the dataset.)

(a) (5 marks) Train a decision tree using both the Gini index and the Entropy by changing the max-depth [4, 8, 10, 15, 20]. Don't change any of the other default values of the classifier. In the following model, use the criteria which give better accuracy on the test set with the chosen depth

(b) (5 marks) Ensembling is a method to combine multiple not-so-good models to get a better performing model. Create 100 different decision stumps (max depth 3). For each stump, train it on randomly selected 50% of the training data, i.e., select data for each stump separately. Now, predict the test samples' labels by taking a majority vote of the output of the stumps. How is the performance affected as compared to parts (a)

(c) (5 marks) Another popular boosting technique is Adaboost. Use the sklearn Adaboost algorithm on the above dataset and report the testing accuracy. Use the Decision tree as the base estimator and with a number of estimators as [4, 8, 10, 15, 20]. Compare RF and Adaboost results.