Homework-3

Machine Learning 1

Instructions: All questions are mandatory. Do submit the answers in PDF format, with the file name being your name. The deadline for submission is December 24, 2022, at 11:59 p.m.

Problem 1. Consider that you are willing to buy a car and you have collected information having four attributes 'price', 'maintenance', 'capacity' and 'airbag', and are trying to predict whether a given car is 'profitable' or not. Assume all the four attributes are categorical, with discrete values. Here is a training and test data for the car profitability prediction problem:

Training data:

Car 1: Price = High, Maintenance = High, Capacity = 4, Airbag = Yes. Profitable? No.

Car 2: Price = Low, Maintenance = Low, Capacity = 2, Airbag = No. Profitable? Yes.

Car 3: Price = High, Maintenance = Low, Capacity = 4, Airbag = Yes. Profitable? No.

Car 4: Price = Low, Maintenance = High, Capacity = 2, Airbag = No. Profitable? No.

Car 5: Price = Low, Maintenance = Low, Capacity = 4, Airbag = Yes. Profitable? Yes.

Car 6: Price = High, Maintenance = High, Capacity = 2, Airbag = No. Profitable? No.

Car 7: Price = Low, Maintenance = Low, Capacity = 4, Airbag = No. Profitable? Yes.

Test data:

Car 8: Price = High, Maintenance = High, Capacity = 4, Airbag = Yes. Profitable? No.

Car 9: Price = Low, Maintenance = Low, Capacity = 2, Airbag = No. Profitable? Yes.

Car 10: Price = High, Maintenance = Low, Capacity = 4, Airbag = Yes. Profitable? No.

Car 11: Price = Low, Maintenance = High, Capacity = 2, Airbag = No. Profitable? No.

Car 12: Price = Low, Maintenance = Low, Capacity = 4, Airbag = No. Profitable? Yes

Train a decision tree classifier on the train-data for a 2-layer decision tree computing relevant information gain at each layer. Test your model on test-data (where the "profitable" label is unseen) and report the accuracy. **(10 points)**

Problem 2. In order to reduce my email load, I decided to implement a machine learning algorithm to decide whether or not I should read an email, or simply discard it. To train my módél, I obtain the following dataset of binary valued features about each email, including whether I know the author or not, whether the email is long or short, and whether it has any of several key words, along with my final decision about whether to read it ($y = +1$ for "read", $y = -1$ for "discard").

x_1 know author?	x_2 is long?	x_3 has 'research'	x_4 has 'grade'	x_5 has 'lottery'	$y \Rightarrow$ read?
0	0	1	1	0	-1
1	1	0	1	0	-1
0	1	1	1	1	-1
1	1	1	1	0	-1
0	1	0	0	0	-1
1	0	1	1	1	1
0	0	1	0	0	1
1	0	0	0	0	1
1	0	1	1	0	1
1	1	1	1	1	-1

In the case of any ties, we will prefer to predict class +1.

1. Calculate the entropy of the class variable y

2. Calculate the information gain for each feature x_i. Which feature should I split on first?

3. Draw the complete decision tree that will be learned from these data. **(10 points)**

Problem 3. (BONUS) Consider a dataset of postgraduate students with the following features:

- Student ID
- Age
- Gender (Male or Female)
- Major (Computer Science, Biology, Psychology, or Other)
- GPA (on a scale of 0 to 4)
- Research experience (Yes or No)
- The target variable is whether the student is accepted into a postgraduate program (Yes or No).

Student 1: ID = 1, Age = 25, Gender = Male, Major = Computer Science, GPA = 3.7, Research experience = Yes. Accepted into postgraduate program? Yes.

Student 2: ID = 2, Age = 22, Gender = Female, Major = Biology, GPA = 3.5, Research experience = No. Accepted into postgraduate program? No.

Student 3: ID = 3, Age = 30, Gender = Male, Major = Psychology, GPA = 3.9, Research experience = Yes. Accepted into postgraduate program? Yes.

Student 4: ID = 4, Age = 28, Gender = Female, Major = Other, GPA = 3.2, Research experience = No. Accepted into postgraduate program? No.

Student 5: ID = 5, Age = 25, Gender = Male, Major = Computer Science, GPA = 3.8, Research experience = Yes. Accepted into postgraduate program? Yes.

Use the AdaBoost algorithm to build a classifier to predict whether a postgraduate student will be accepted into a program based on their age, gender, major, GPA, and research experience. Assume that the algorithm will use weak learners (e.g., 2-layer decision trees with a weight on the training examples). **(10 points)**

Problem 4. Use the XOR example in class with 2 inputs, 2 neurons in hidden layer and an output. Use the initial weights as in the class. For all the 4 possible inputs (01, 10, 00, 11) taken together in each batch update, perform the weight updates using gradient descent for two iterations. (Hint: Use the sum of gradients due to each of the 4 inputs in the gradient descent)

(10 points)

Problem 5. MNIST (Modified National Institute of Standards and Technology) is a popular dataset for handwritten digit classification. It consists of 60,000 training images and 10,000 test images, all of which are 28x28 grayscale images.

you can download the training and test images from MNIST DATABASE. Take the first 200 images of number 3 and use PCA on them with 3 basis vectors ($d = 3$). Pictorially represent μ, v_1, v_2, and v_3. **(10 points)**