ALDA Fall 2021
HW3
9/20/2021

HW2 contains 3 questions. Please read and follow the instructions.

- **DUE DATE FOR SUBMISSION: 9/28/2021 11:45 PM**
- **TOTAL NUMBER OF POINTS: 70**
- **NO PARTIAL CREDIT** will be given so provide concise answers.
- **Submissions and updates should be handled by the same person.**
- **You MUST manually add ALL team members in the submission portal when you submit through Gradescope.**
- Make sure you clearly list **your homework team ID, all team members' names and Unity IDs, for those who have contributed to the homework contribution** at the top of your submission.
- **[GradeScope and NCSU Github]:** Submit a PDF on GradeScope. **You must submit your code in Github, and give the instructors access.**

  Follow these Github instructions: Please use the same homework repository you created for ALDA class and shared with us for HW1. The only thing you need to do for HW3 is to **create a new folder named HW3**. **All code MUST be in HW3 folder before the deadline. No credit will be given if code is not submitted for a programming question.** In your PDF submitted on GradeScope, reference the script/function for each question (e.g. "For solution to question 2, see matrix.py"). **Include your team's GitHub repository link in the PDF.**
- The materials on this course website are only for use of students enrolled in this course and **MUST NOT** be retained or disseminated to others.
- By uploading your submission, you agree that you have not violated any university policies related to the student code of conduct (https://policies.ncsu.edu/policy/pol-11-35-01/), and you are signing the Pack Pledge: **"I have neither given nor received unauthorized aid on this test or assignment"**.

1. (16 points) [**KNN + CV**] [**Ge Gao**] Considering the dataset with two real-valued inputs $x1$ and $x2$ and one binary output y in the table below. Each data point will be referred using the first column ”ID” in the following. You will use KNN with Manhattan distance to predict y.

   Write code in Python to perform the following tasks; if needed, you are allowed to use `scipy`, `sklearn`, and `numpy` packages. Please submit one code file via the NCSU GitHub repository you have been given. **Show your work. Show steps for reaching the answer.**

ID	x1	x2	y
1	-5.86	-2.0	$\star$
2	-10.97	-1.0	$\star$
3	0.79	-2.0	♠
4	-0.59	1.0	♠
5	3.63	-2.0	♠
6	2.02	-5.0	♠
7	-6.41	-1.0	$\star$
8	6.13	-7.0	♠
9	-2.35	6.0	$\star$
10	2.66	-3.0	♠
11	-3.71	2.0	$\star$
12	2.4	1.0	$\star$

   (a) (4 points) What is the leave-one-out cross-validation error of 1NN on this dataset?

   (b) (2 points) What are the 3 nearest neighbors for data points 3 and 10 respectively.

   (c) (5 points) What is the 3-folded cross-validation error of 3NN on this dataset? For the ith fold, the testing dataset is composed of all the data points whose (ID mod $3 = i - 1$).

   (d) (5 points) Based on the results of (a) and (c), can we determine which is a better classifier, 1NN or 3NN? Why? (Answers without a correct justification will get zero points.)

2. (30 points) [**Adaboost**] [**John Wesley Hostetter**] Consider the labeled data points in Figure 1, where '+' and '-' indicate class labels. We will use AdaBoost with Separating Hyperplane to train a classifier for the '+' and '-' labels. Each boosting iteration will select a horizontal or vertical Separating Hyperplane: **a vertical or horizontal line** that would split the space into half-spaces with a goal of minimizing the weighted training error. Breaking ties by choosing '+'. All of the data points start with uniform weights. Please display your answers for (a), (b), (d) and (e) in a single figure.

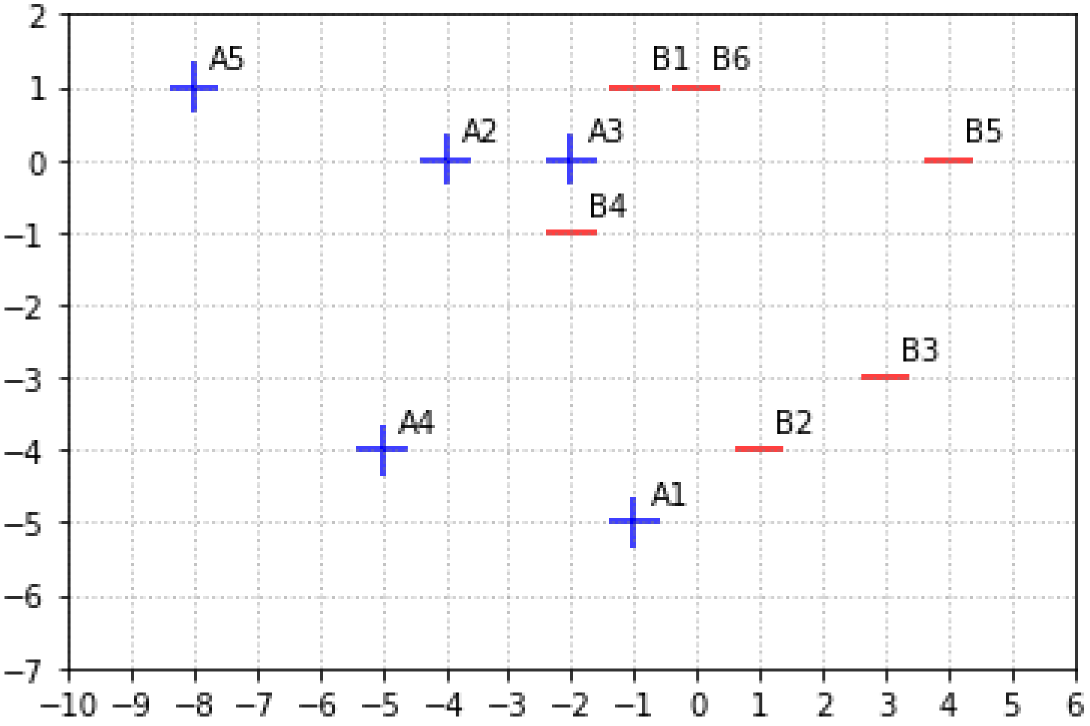


Figure 1: The Original Data Observations.

(a) (3 points) In Figure 1, draw a decision boundary corresponding to the first decision stump that the algorithm would choose (the decision boundary should be either a vertical or horizontal straight line). Label the decision boundary as (1), also indicate the '+' / '-' sides of this boundary.

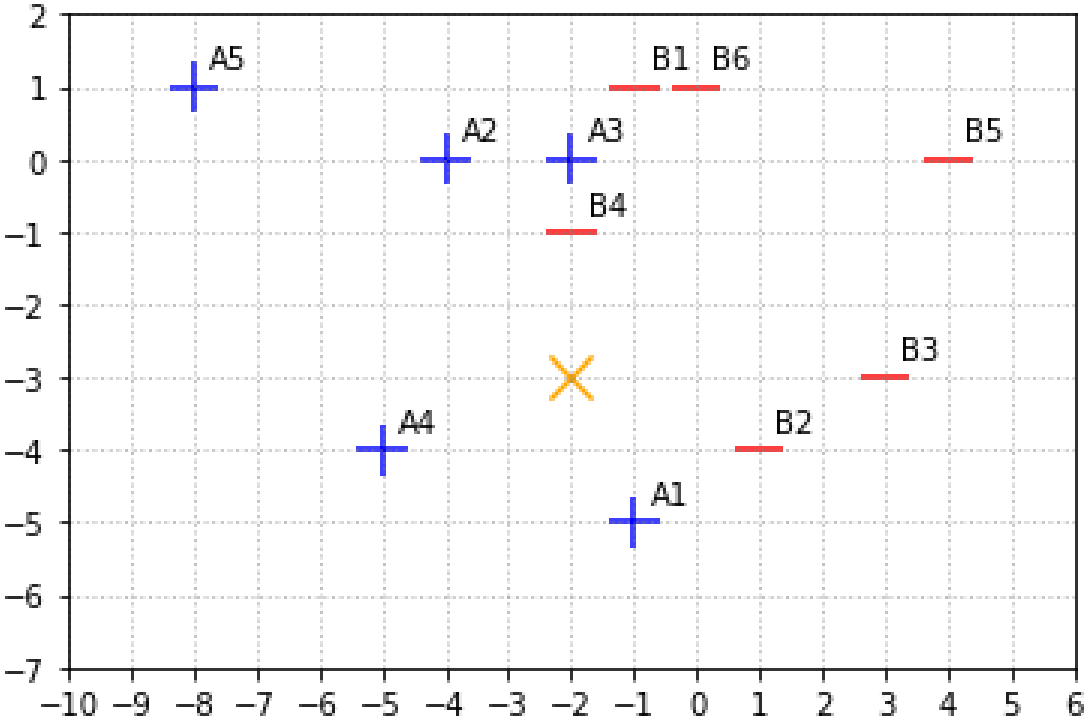
(b) (2 points) Circle the point(s) that have the highest weight after the first boosting iteration. Also, report the value of the highest weight and show your calculations.

(c) (5 points) After the labels have been re-weighted in the first boosting iteration, what is the weighted error of the decision boundary (1)?

(d) (3 points) Draw the decision boundary corresponding to the second decision stump that the algorithm would choose. Label the decision boundary as (2), also indicate the '+' / '-' sides of this boundary.

(e) (5 points) Next, compute the weighted error of the decision boundary (2) and draw a decision boundary corresponding to the third decision stump that the algorithm would choose. Label the decision boundary as (3), also indicate the '+' / '-' sides of this boundary.

(f) (7 points) Assuming that a "New Data point" is given (shown in the graph below), using your classifier built from decision boundaries (1), (2) and (3) to predict the class label for the new data point. Provide your final classifier along with the class label. Show your work.



3. (24 points) [**Naïve Bayes + Decision Tree**] [**Angela Zhang**] Consider the training dataset below. Your goal is to build a classifier to predict *whether a customer will click on an ad.* The output class is in the last column "Clicked Ad" and the input attributes are: "Image Colors", "Image Size", "Product History" and "Ad Placement". More specifically, you will compare Naïve Bayes (NB) and Decision Tree (DT).

   For Naïve Bayes (NB), you will use *m-estimate* from the lecture with $m = 2$ and $p = 0.5$ for probability estimations.

   For Decision Tree (DT), you will follow the lecture's code to build your trees without pruning except that multiple-way splitting is allowed, and use Information Gain (IG) to select the best attribute. In the case of ties, break ties in favor of the leftmost feature.

ID	Image Colors	Image Size	Product History	Ad Placement	Clicked Ad
1	Neutral	Large	Well-Known	Bottom	No
2	Neutral	Medium	Well-Known	Top	No
3	Bright	Large	New	Bottom	No
4	Neutral	Medium	New	Bottom	No
5	Neutral	Large	New	Bottom	No
6	Neutral	Large	New	Top	No
7	Bright	Large	New	Bottom	No
8	Bright	Medium	Well-Known	Bottom	Yes
9	Bright	Large	New	Top	Yes
10	Bright	Large	Well-Known	Bottom	Yes
11	Bright	Medium	Well-Known	Bottom	Yes
12	Neutral	Medium	Well-Known	Top	Yes
13	Bright	Large	New	Bottom	Yes
14	Bright	Large	Well-Known	Top	Yes

   (a) (18 points) Compare the performance of NB vs. DT using 2-fold cross-validation (CV) and **report their 2-fold CV accuracy**. **For the ith fold, the testing dataset is composed of all the data points whose (ID mod 2 $= i - 1$). For each fold, show the induced Naïve Bayes (in order of left to right columns) and DT models.**

   (b) (6 points) Based on the **2-fold CV accuracy** from (a), which classifier, NB or DT, would you choose? Report your final model for the selected classifier.