AI Academy: Introduction to Data Mining
Week 6 Workshop

Workshop 6 contains 2 questions.

---

1 KNN + CV

1. (16 points) [**KNN + CV**] 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 unweighted Euclidean distance to predict y.

   You can write code in Python to calculate a distance matrix, which will help you in your calculations.

ID	x1	x2	y
1	-3.44	1	⋆
2	-6.48	5	♠
3	0.93	-2	⋆
4	0.2	2	♠
5	-6.69	13	⋆
6	-5.85	4	⋆
7	3.0	0	♠
8	-0.36	0	♠
9	1.68	-3	♠
10	-0.45	-3	⋆

   (a) (2 points) What are the 3 nearest neighbors for data points 2 and 8 respectively.

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

   (c) (5 points) What is the 3-folded cross-validation error of 3NN on this dataset?

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

2 Adaboost Classifier

In this problem you will perform some steps of the Bagging algorithm on the dataset given in Table 1. In the dataset, each data point has a continuous attribute x, a categorical Class label y and an index i for reference. Table 2 shows the samples generated during the first iteration of bootstrap sampling.

Table 1: Dataset for adaboost classification

i (index)	1	2	3	4	5
x (attribute)	0.1	0.2	0.3	0.4	0.7
y (class label)	1	-1	1	1	-1

Table 2: Boosting (Round 1) data samples and their corresponding classes

sampled i	1	1	3	5	5
x	0.1	0.1	0.3	0.7	0.7
y	1	1	1	-1	-1

The decision stump f_1 was generated during Round 1 using data from Table 2 and is as shown below:

$$x <= 0.35 \Rightarrow y = 1$$
$$x > 0.35 \Rightarrow y = -1$$

You are given the following formula to calculate the error ϵ_m for round m:

$$\epsilon_m = \sum_{i=1}^{N} w_i^{(m-1)} I(f_m(\mathbf{x}_i) \neq y_i)$$

where $I(p) = 1$ if the predicate p is true, 0 otherwise and $w_i^{(m-1)}$ refers to the weight from round $m-1$ (or the initial weights when $m = 1$).
The importance α_m of a classifier f_m on round m is given by the formula:

$$\alpha_m = \frac{1}{2} ln(\frac{1-\epsilon_m}{\epsilon_m})$$

When answering the following questions, record your answers in the table below.

i	1	2	3	4	5
$w_i^{(0)}$					
$\Delta w_i^{(0 \Rightarrow 1)}$ (+/−/=)					
$F(x_i)$					

1. (2 points) Calculate the starting weight $w_i^{(0)}$ of each instance and record it in the table above.

2. (4 points) Calculate α_1, the importance of the first classifier, f_1, using the formulae above.

3. (2.5 points) At the end of Round 1, the Adaboost algorithm will update the each weight w_i. In the second row of the table above ($\Delta w_i^{(0 \Rightarrow 1)}$), indicate whether the weight Decreased ($-$), Increased ($+$) or Stayed the same ($=$) from Round 0 to the end of round Round 1. For example, if $w_1^{(0)} < w_1^{(1)}$, write "+" in the first blank.

4. (2.5 points) The decision stump f_2 was generated during Round 2 is as shown below:

$$x <= 0.5 \Rightarrow y = 1$$
$$x > 0.5 \Rightarrow y = -1$$

   The importance of f_2 was calculated at $\alpha_2 = 0.81$. Assume the Adaboost algorithm ended after 2 rounds. Use f_1, f_2, α_1 and α_2 to calculate how each training instance $x_1 \ldots x_5$ will be classified by the final classifier $F(x_i)$. Put your answers in the final row of the table above (write either -1 or 1).

5. (4 points) The table below gives results from running the Adaboost algorithm on a *different* dataset for 2 rounds. Each row gives the Round (m), the importance for the classifier α_m, and whether each of four instance 1, 2, 3 and 4, were classified correctly during that round. (False = misclassified, True = classified)

Round (m)	α_m	$f_m(x_1) = y_1$	$f_m(x_2) = y_2$	$f_m(x_3) = y_3$	$f_m(x_4) = y_4$
1	0.670	False	True	True	False
2	0.243	True	False	True	False

   After Round 2, rank the weights of each instance from **lowest to highest**. Write the following in the correct order: $w_1^{(2)}$, $w_2^{(2)}$, $w_3^{(2)}$, $w_4^{(2)}$. (Note: you do not actually have to calculate any weights.)