Bronco ID: 014429779
Last Name: Koepke
First Name: Christopher

CS 4210.01 - Assignment #3

1. Considering the 1-D dataset below and the following bootstrap samples (bagging rounds 1 to 5) randomly generated during the bagging process. Show how a bagging algorithm can perfectly classify this data by **drawing** and **writing** the decision stumps for each round, the summary table of the trained decision stumps, and the combination table of your base classifiers with the final predictions. Hint: you might need to test alternative but equally accurate decision stumps on your training set to get maximum accuracy.

x	1	2	3	4	5
y	-1	-1	1	1	-1

DATASET

x	1	1	2	4	5
y	-1	-1	-1	1	-1

ROUND 1

$x \leq 3 \rightarrow y = -1$

$x > 3 \rightarrow y = 1$

x	3	3	4	4	5
y	1	1	1	1	-1

ROUND 2

$x \leq 4.5 \rightarrow y = 1$

$x > 4.5 \rightarrow y = -1$

<u>NOTE</u>: RED BARS DENOTE DECISION STUMPS.

x	1	2	2	5	5
y	-1	-1	-1	-1	-1

ROUND 3

$x \leq 0.5 \rightarrow y = 1$

$x > 0.5 \rightarrow y = -1$

x	1	3	4	4	5
y	1	1	1	1	-1

ROUND 4

$x \leq 4.5 \rightarrow y = 1$

$x > 4.5 \rightarrow y = -1$

x	1	2	3	3	4
y	-1	-1	1	1	1

ROUND 5

$x \leq 2.5 \rightarrow y = -1$

$x > 2.5 \rightarrow y = 1$

- FINAL TABLES

Rounds	x = 1	x = 2	x = 3	x = 4	x = 5
1	-1	-1	-1	1	1
2	1	1	1	1	-1
3	-1	-1	-1	-1	-1
4	1	1	1	1	-1
5	-1	-1	1	1	1
Sign	-1	-1	1	1	1

Summary Table

Rounds	Split	Left	Right
1	3	-1	1
2	4.5	1	-1
3	0.5	1	-1
4	4.5	1	-1
5	2.5	-1	1

Combination Table

2. Considering the different 1-D dataset below and the following rounds from 1 to 3 randomly generated during the boosting process. Show how a boosting algorithm can perfectly classify this data by **drawing** and **writing** the decision stumps and weights for each round, the summary table of the trained decision stumps, and the combination table of your base classifiers with the weighted final predictions. Hint: there is a single best decision stump (more accurate) for each round.

<u>NOTE</u>: RED BARS DENOTE DECISION STUMPS.

x	1	2	3	4	5
y	1	1	-1	-1	1

DATASET

x	1	2	3	4	4
y	1	1	-1	-1	-1

Round 1

$x \leq 2.5 \rightarrow y = 1$

$x > 2.5 \rightarrow y = -1$

x	5	5	5	5	5
y	1	1	1	1	1

Round 2

$x \leq 5.5 \rightarrow y = 1$

$x > 5.5 \rightarrow y = -1$

x	3	3	4	4	5
y	-1	-1	-1	-1	1

Round 3

$x \leq 4.5 \rightarrow y = -1$

$x > 4.5 \rightarrow y = 1$

- WEIGHTS FOR ROUND 1:

$$W_1^1 = W_2^1 = W_3^1 = W_4^1 = W_5^1 = \frac{1}{5} = 0.2$$

> - $\boldsymbol{Error\ Rate}\ (\epsilon_1) = \left(\frac{1}{N}\right)\sum_{j=1}^{N} W_j^{(i)}\delta(C_1(x_j \neq y_j))$
> - $\boldsymbol{Importance\ of\ a\ classifier}\ (a_i) = \left(\frac{1}{2}\right)\ln\left(\frac{1-\epsilon_i}{\epsilon_i}\right)$
> - $\boldsymbol{Weight\ Update\ by\ using\ a_i}$: $W_j^{(i+1)} = \frac{w_j^i}{Z_i} * \begin{cases} e^{-a_i} & if\ C_i(x_i) = y_i \\ e^{a_i} & if\ C_i(x_i) \neq y_i \end{cases}$
>
> $$where\ Z_i\ is\ the\ normalization\ factor\ to\ ensure\ that\ \sum_j W_j^{(i+1)} = 1$$

- UPDATED WEIGHTS FOR ROUND 2:
  - $For\ a_1:\ \ \epsilon_1 = 0.2[(0.2*0) + (0.2*0) + (0.2*0) + (0.2*0) + (0.2*1)] = 0.04$
  - $a_1 = \left(\frac{1}{2}\right)\ln\left(\frac{1-0.04}{0.04}\right) = 1.589026915 \ \approx 1.589$
  - **Updating Weights**

$W_5^2 = \dfrac{0.2 * e^{1.589}}{Z_i} = \dfrac{0.980}{Z_i} = \dfrac{0.980}{1.144} \approx 0.8566$ $W_1^2 = W_2^2 = W_3^2 = W_4^2 = \dfrac{0.2 * e^{-1.589}}{Z_i} = \dfrac{0.041}{Z_i} = \dfrac{0.041}{1.144} \approx 0.036$	$Z_1 = (0.980 * 1) + (0.041 * 4) = 1.144$

- UPDATED WEIGHTS FOR ROUND 3:
  - $For\ a_2:\ \ \epsilon_2 = 0.2[(0.036*0) + (0.036*0) + (0.036*1) + (0.036*1) + (0.8566*0)]$ $= 0.0144$
  - $a_2 = \left(\frac{1}{2}\right)\ln\left(\frac{1-0.0144}{0.0144}\right) = 2.113011193 \ \approx 2.113$
  - **Updating Weights**

$W_3^3 = W_4^3 = \dfrac{0.036 * e^{2.113}}{Z_i} = \dfrac{0.2978}{Z_i} = \dfrac{0.2978}{0.7079} \approx 0.4207$ $W_1^3 = W_2^3 = \dfrac{0.036 * e^{-2.113}}{Z_i} = \dfrac{0.0044}{Z_i} = \dfrac{0.0044}{0.7079} \approx 0.0062$ $W_5^3 = \dfrac{0.8566 * e^{-2.113}}{Z_i} = \dfrac{0.1035}{Z_i} = \dfrac{0.1035}{0.7079} \approx 0.1462$	$Z_1 = (0.2978 * 2) + (0.0044 * 2) + (0.1035 * 1)$ $= 0.7079$

- CALCULATING ERROR RATE AND ALPHA FOR NEXT POSSIBLE ROUND:
  - $For\ a_3:\ \ \epsilon_3 = 0.2[(0.0062*1) + (0.0062*1) + (0.4207*0) + (0.4207*0) + (0.1462*0)]$ $= 0.00248$
  - $a_3 = \left(\frac{1}{2}\right)\ln\left(\frac{1-0.00248}{0.00248}\right) = 2.998506819 \ \approx 2.9985$

- FINAL TABLES

Round	x=1	x=2	x=3	x=4	x=5
1	0.2	0.2	0.2	0.2	0.2
2	0.036	0.036	0.036	0.036	0.8566
3	0.0062	0.0062	0.4207	0.4207	0.1462

Weights Table

Round	Split point	Left	Right	Alpha
1	2.5	1	-1	1.589
2	5.5	1	-1	2.113
3	4.5	-1	1	2.9985

Summary Table

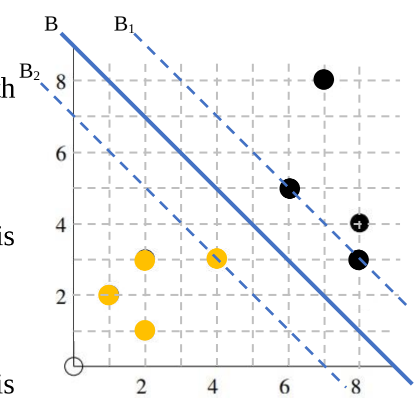
Round	x=1	x=2	x=3	x=4	x=5
1	1	1	-1	-1	-1
2	1	1	1	1	1
3	-1	-1	-1	-1	1
Sum	0.7035	0.7035	-2.4745	-2.4745	3.5225
Sign	1	1	-1	-1	1

Combination Table

3. Complete the Python program (bagging_random_forest.py) that will read the file optdigits.tra (3,823 samples) that includes training instances of handwritten digits (optically recognized). Read the file optdigits.names to get detailed information about this dataset. Also, check the file optdigits-orig.tra and optdigits-orig.names to see the original format of this data, and how it was transformed to speed-up the learning process (pre-processing phase). Your goal is to build a base classifier by using a single decision tree, an ensemble classifier that combines multiple decision trees, and a Random Forest classifier to recognize those digits. To test the accuracy of those distinct models, you will use the file optdigits.tes (1,797 samples).

https://github.com/chris-k87/CS_4210.01/tree/main/Assignment_3/Bagging_Random_Forest

4. Say you are given the training dataset shown to the right. This is a binary classification task in which the instances are described by two integer-valued attributes.



   a. Draw the decision boundary and its parallel hyperplanes for a linear SVM with maximum margin (hard margin formulation) and identify the support vectors.
      - Support vectors: (4,3), (6,5), and (8,3)

   b. If a black circle is added as a training sample in the position (7,5), does this affect the previously learned decision boundary? Explain why.
      - No; its position is “above” the current B_1 hyperplane.

   c. If a yellow circle is added as a training sample in the position (4,2), does this affect the previously learned decision boundary? Explain why.
      - No; its position is “below” the current B_2 hyperplane.

   d. If a black circle is added as a test sample in the position (7,5), will this sample be classified correctly according to the previously learned decision boundary? Explain why.
      - Yes; it is “above” the decision boundary B.

   e. If a black circle is added as a test sample in the position (6,4), will this sample be classified correctly according to the previously learned decision boundary? Explain why.
      - Yes; it is “above” the decision boundary B.

f. If a yellow circle is added as a test sample in the position (4,2), will this sample be classified correctly according to the previously learned decision boundary? Explain why.
   - Yes; it is “below” the decision boundary B.

g. If a yellow circle is added as a test sample in the position (5,3), will this sample be classified correctly according to the previously learned decision boundary? Explain why.
   - Yes; it is “below” the decision boundary B.

h. If a black circle is added as a test sample in the position (5,3), will this sample be classified correctly according to the previously learned decision boundary? Explain why.
   - No; it is “below” the decision boundary B.

i. If a yellow circle is added as a test sample in the position (6,4), will this sample be classified correctly according to the previously learned decision boundary? Explain why.
   - No; it is “above” the decision boundary B.

j. If a black circle is added as a training sample in the position (4,4), how this will affect the decision boundary if C = 1 and C = ∞? Consider the soft margin formulation.
   - If C = 1, the decision boundary would not change.
   - If C = ∞, then the decision boundary would move, and its margins would become very narrow.

5. Consider the following 1-dimensional data with two classes:

x	-3	0	1	2	3	4	5
Class	−	−	+	+	+	+	+

   a. Find the decision boundary of a linear SVM on this data (hard-margin formulation) and identify the support vectors (write the x coordinate to provide your answer).
      - The decision boundary would be at coordinate $x = 0.5$ with the left side classified as negative and right side classified as positive.
      - The support vectors would be at coordinate $x = 0$ and $x = 1$.

   b. Find the solution parameters w and b for this linear SVM and the width of the margin. Hint: place the identified support vectors (positive and negative) into the formula $y_i(w.x_i + b) = 1$ since you know this formula holds for them.
      - For $x = 0$:
         - $-1(w.0 + b) = 1$
         - $-b = 1$
         - $b = -1$

      - For $x = 1$:
         - $1(w.1 + b) = 1$
         -

   c. Show mathematically that the SVM classifications for the test data {-1.5, 1.5} are negative and positive respectively.
      - A

   d. Suppose we remove the point (1,+) from this training set and train the SVM again. Find the new values of the solution parameters w and b and the width of the margin.
      - A

6. The quadratic kernel $K(x, y) = (x.y + 1)^2$ should be equivalent to mapping each x into a six-dimensional space, where $\Phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$ for the case where $x = (x_1, x_2)$. Demonstrate this equivalence by answering the following questions while using the data points: A = (1,2), B = (2,4).

   a. $\Phi(A)$
      - $\Phi(A) = \left(1^2,\ 2^2, \sqrt{2}(1)(2), \sqrt{2}(1), \sqrt{2}(2), 1\right)$
        $= (1, 4, 2\sqrt{2}, \sqrt{2}, 2\sqrt{2}, 1)$

   b. $\Phi(B)$
      - $\Phi(B) = \left(2^2,\ 4^2, \sqrt{2}(2)(4), \sqrt{2}(2), \sqrt{2}(4), 1\right)$
        $= (4, 16, 8\sqrt{2}, 2\sqrt{2}, 4\sqrt{2}, 1)$

   c. $\Phi(A)\Phi(B)$
      - $\Phi(A)\Phi(B) = \left(1, 4, 2\sqrt{2}, \sqrt{2}, 2\sqrt{2}, 1\right)\left(4, 16, 8\sqrt{2}, 2\sqrt{2}, 4\sqrt{2}, 1\right)$
        $= \left((1 * 4) + (4 * 16) + \left(2\sqrt{2} * 8\sqrt{2}\right) + \left(\sqrt{2} *\ 2\sqrt{2}\right) + \left(2\sqrt{2} +\ 4\sqrt{2}\right) + (1 * 1)\right)$
        $= (4 + 64 + 32 + 4 + 16 + 1) = 121$

   d. $K(A, B)$. Hint: your answers for (c) and (d) should be the same. By using the kernel function, SVM "cheats" and performs significantly fewer calculations (kernel trick).
      - $K(A, B) = (A.B + 1)^2 = ([(1 * 2) + (2 * 4)] + 1)^2 = ([10] + 1)^{\ 2} =\ (11)^2 = 121$

7. Complete the Python program (svm.py) that will also read the file optdigits.tra to build multiple SVM classifiers. You will simulate a grid search, trying to find which combination of four SVM hyperparameters (c, degree, kernel, and decision_function_shape) leads you to the best prediction performance. To test the accuracy of those distinct models, you will also use the file optdigits.tes. You should update and print the accuracy, together with the hyperparameters, when it is getting higher.

https://github.com/chris-k87/CS_4210.01/tree/main/Assignment_3/SVM