CS 4210 – Assignment #3
Maximum Points: 100 pts.

Bronco ID: 015412449
Last Name: Langgeng
First Name: Leonardo

Note 1: Your submission header must have the format as shown in the above-enclosed rounded rectangle.
Note 2: Homework is to be done individually. You may discuss the homework problems with your fellow students, but you are NOT allowed to copy – either in part or in whole – anyone else's answers.
Note 3: Your deliverable should be a .pdf file submitted through Gradescope until the deadline. Do not forget to assign a page to each of your answers when making a submission. In addition, source code (.py files) should be added to an online repository (e.g., github) to be downloaded and executed later.
Note 4: All submitted materials must be legible. Figures/diagrams must have good quality.
Note 5: Please use and check the Canvas discussion for further instructions, questions, answers, and hints. The bold words/sentences provide information for a complete or accurate answer.

1. [15 points] 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.

Dataset

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

Round 1

x	1	1	2	4	5
Class:	-1	-1	-1	|X| 1	-1

x <= 3 : y = -1
y > 3 : y = 1

Round 2

x	3	3	4	4	5
Class:	1	1	1	1	|X| -1

x <= 4.5: y = 1
y > 4.5 : y = -1

Round 3

x	1	2	2	5	5
Class:	-1	-1	-1	-1	-1 |X|

x <= 5 : y = -1
y > 5 : y = -1

Round 4

x	1	3	4	4	5
Class:	-1	1	1	1	|X| -1

x <= 4.5: y = 1
y > 4.5 : y = -1

x	1	2	3	3	4
Class:	-1	-1	|X|	1	1	1

Round 5

x <= 2.5: y = -1
y > 2.5 : y = 1

Round	Split pt	Left class	Right class
1	3	-1	1
2	4.5	1	-1
3	5	-1	-1
4	4.5	1	-1
5	2.5	-1	1

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
4	1	1	1	1	-1
5	-1	-1	1	1	1
Sum	-1	-1	1	3	-1
Pred class	**-1**	**-1**	**1**	**1**	**-1**

2. [12 points] 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.

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

Dataset

x	1	2	3	4	4
Class:	1	1	|X|	-1	-1	-1

Round 1

x <= 2.5: y = 1
y > 2.5 : y = -1

Round 2

x	5	5	5	5	5
Class:	1	1	1	1	1 |X|

x <= 5 : y = 1
y > 5 : y = 1

x	3	3	4	4	5
Class:	-1	-1	-1	-1	|X|	1

Round 3

x <= 4.5: y = -1
y > 4.5 : y = 1

Weights:

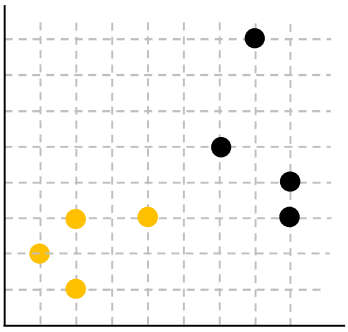
Round	x = 1	x = 2	x = 3	x = 4	x = 5
1	0.2	0.2	0.2	0.2	0.2
2	0.03571	0.03571	0.03571	0.03571	0.85714
3	0.00142	0.00142	0.49786	0.49786	0.00142

Round	Split pt	Left class	Right class	Alpha
1	2.5	1	-1	1.5890
2	5	1	1	2.9276
3	4.5	-1	1	4.5413

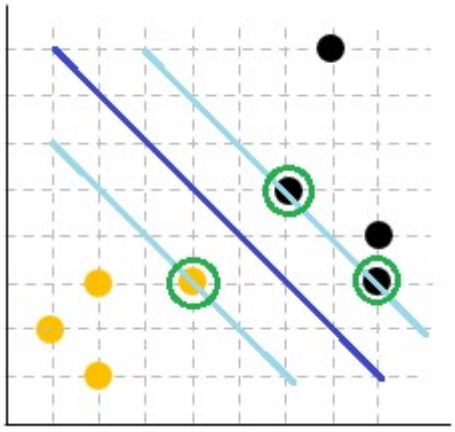
Class:

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	1	1	-1	-1	1
Pred class	**1**	**1**	**-1**	**-1**	**1**

3. [15 points] 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/leolanggeng/assgn3_bagging_svm
4. [20 points] Say you are given the training dataset shown below. This is a binary classification task in which the instances are described by two integer-valued attributes.



   a. [2 points] Draw the decision boundary and its parallel hyperplanes for a linear SVM with maximum margin (hard margin formulation) and identify the support vectors.



SUPPORT VECTOR: GREEN CIRCLE

b. [2 points] 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, as that position is inside the black area(top right of boundary line), and also not inside or crossing the margin.

c. [2 points] 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, as that position is inside the yellow area(bottom left of boundary line), and also not inside or crossing the margin.

d. [2 points] 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, as it is above/right of the boundary line which is area for black dot

e. [2 points] 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, as it is above/right of the boundary line which is area for black dot and still within the area of margin.

f. [2 points] 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, as it is above/right of the boundary line which is area for yellow dot

g. [2 points] 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, as it is below/left of the boundary line which is area for yellow dot and still within the area of margin.

h. [2 points] 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, as its below/left of the boundary line which is classified as yellow

i. [2 points] 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, as its below/left of the boundary line which is classified as black

j. [2 points] 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.
On C = 1, the boundary will probably looks the same as it allows large number of point to be inside(and even crossing) the margin area. If C = inf, the boundary line will move and the margin area will be much smaller as it is basically becoming hard margin classification.

5. [11 points] Consider the following 1-dimensional data with two classes:

x	-3	0	1	2	3	4	5
Class	−	−	+	+	+	+	+

   a. [3 points] 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).
   Decision boundary: x = 0.5
   Margin : 1
   Support vector: x=0, x=1
   b. [3 points] 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.
   x=0, y=-1 : -1 (w * 0 + b) = 1; b = -1
   x=1, y=1 : 1 (w * 1 + b) = 1; w = 2
   c. [2 points] Show mathematically that the SVM classifications for the test data {-1.5, 1.5} are negative and positive respectively.
   b = -1, w = 2
   x = -1.5 : Yi (2 *(-1.5) – 1) = 1; Yi = -1/4 (- class)
   x = 1.5 : Yi (2 *(1.5) – 1) = 1; Yi = ½ (+ class)
   d. [3 points] 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.
   Support vector: x=0, x=2
   Margin = 2
   x=0, y=-1 : -1 (w * 0 + b) = 1; b = -1
   x=2, y=1 : 1 (w * 2 + b) = 1; w = 1

6. [12 points] 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. [3 points] $\Phi(A)$
   (1, 4, 2.8284, 1.4142, 2.8284, 1)
   b. [3 points] $\Phi(B)$
   (4, 16, 11.3137, 2.8284, 5.6568, 1)
   c. [3 points] $\Phi(A)\Phi(B)$
   120.999
   d. [3 points] $K(A, B)$. 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).
   (10+1)^2 = 121 (same, except rounding error)

7. [15 points] 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/leolanggeng/assgn3_bagging_svm

Important Note: Answers to all questions should be written clearly, concisely, and unmistakably delineated. You may resubmit multiple times until the deadline (the last submission will be considered).

NO LATE ASSIGNMENTS WILL BE ACCEPTED. ALWAYS SUBMIT WHATEVER YOU HAVE COMPLETED FOR PARTIAL CREDIT BEFORE THE DEADLINE!