1.

Instance	Attribute 1 (a_1)	Attribute 2 (a_2)	Class
1	T	1.0	+
2	T	6.0	+
3	T	5.0	-
4	F	4.0	+
5	F	7.0	-
6	F	3.0	-
7	F	8.0	-
8	T	7.0	+
9	F	5.0	-

Info Gain is maximized when entropy minimized.

Entropy Parent)

$= -4\log 4/9 - 5\log 5/9$

$= 0.99102$

Entropy a_1

	+	−
T	3	1
F	1	4

Entropy T:

$= -3/4 \log(3/4) - 1/4 \log(1/4)$

$= 0.8113$

Entropy F:

$= -1/5 \log(1/5) - 4/5 \log(4/5)$

$= 0.7219$

Entropy total:

$= 4/9\,(0.8113) + 5/9\,(0.7219)$

$= 0.7616$

Entropy a_2 (using calc)

a_2)

	+	−	+	−	−	+	−	+	−
	1.0	3.0	4.0	5.0	5.0	6.0	7.0	7.0	8.0
	0	2	3.5	4.5	5.5	6.5	7.5		9
+	0 | 4	1 | 3	1 | 3	2 | 2	2 | 2	3 | 1	4	0	4 | 0
−	0 | 5	0 | 5	1 | 4	1 | 4	3 | 2	3 | 2	4	1	5 | 0
Gini	0.991	0.848	0.989	0.918	0.983	0.972	0.889		0.991

↑

minimum entropy

→ Info Gain a_1 = 0.99102 − 0.7616
= 0.22942

Info Gain a_2 = 0.99102 − 0.848
= 0.1519

Thus split first on a_1, as maximizes gain.

1.2) IS we split by "Instance" our entropy after the split will be 0, as each split will hold a unique element either + or −.

Obliously this is an issue as "Instance" or any ID based feature is arbitrarily defined, and provides no useful information about the probability distr. we're trying to estimate.

So no, bad idea to use in descision tree.

#2)

Problem 2. Decision Trees based on GINI index (20 points)

The following table summarizes a dataset with two attributes A, B, and two class labels +, −. The original dataset had 5 attributes, but only 2 of the most informative attributes are selected for further analysis.

A	B	Class Label	
		+	−
T	T	0	20
T	F	20	10
F	T	15	0
F	F	0	35

Cost Matrix	Attribute Value		
Actual Class		T	F
	+	-1	100
	−	0	-10

(1) According to **GINI index**, which attribute would be chosen as the first splitting attribute?
(2) Use the given cost matrix and decide the first splitting attribute. The total cost is the metric for splitting, and this question is independent to question (1). Note that this question is independent to question (1). You will need to use both tables provided above to calculate the cost of splitting A and B, and choose the attribute with smaller cost to split.

Please write down every step including the calculation of the count matrix after splitting, GINI index of the parent, GINI index of the children after splitting, and the cost of splitting based on each attribute.

Gini parent)

$1-\left(\frac{35}{100}\right)^2+\left(\frac{65}{100}\right)^2$

$= 0.455$

Gini A)

T label) $1-\left(\frac{20}{50}\right)^2-\left(\frac{30}{50}\right)^2=$

$1-\left(\frac{4}{25}\right)-\left(\frac{9}{25}\right)=\frac{12}{25}$

	+	−
T	20	30
F	15	35

F label) $1-\left(\frac{15}{50}\right)^2-\left(\frac{35}{50}\right)^2$

$=1-\left(\frac{3}{10}\right)^2-\left(\frac{7}{10}\right)^2$

$=1-\frac{9}{100}-\frac{49}{100}=\frac{42}{100}$

Gini (A's children)

$=\frac{1}{2}\left(\frac{12}{25}\right)+\frac{1}{2}\left(\frac{42}{100}\right)=\frac{1}{2}\left(\frac{20}{100}\right)=0.45$

→ Gain = 0.455 − 0.45 = 0.005

Gini B)

T label)

$=1-\left(\frac{15}{35}\right)^2-\left(\frac{20}{35}\right)^2=0.4898$

	+	−
T	15	20
F	20	45

F label)

$=1-\left(\frac{20}{65}\right)^2-\left(\frac{45}{65}\right)^2=0.426$

Gini (B's children)

$$\frac{35}{100}(0.4898) + \frac{65}{100}(0.426) = \boxed{0.426}$$

→ Gain = 0.455 − 0.426 = 0.029

Thus since gain is maximized when gini minimized,

→ Split on attribute **B**

2.2)

A	B	Class Label	
		+	−
T	T	0	20
T	F	20	10
F	T	15	0
F	F	0	35

Cost Matrix	Attribute Value		
		T	F
Actual Class	+	-1	100
	−	0	-10

Cost A: $\#TT \cdot w_{TT} + \#TF \cdot w_{TF} + \#FT \cdot w_{FT} + \#FF \cdot w_{FF}$

$= 20 \cdot (-1) + 30 \cdot (100) + 0 + 35(-10)$

$= -20 + 3000 - 350$

$= \boxed{2630}$

Cost B: $\#TT \cdot w_{TT} + \#TF \cdot w_{TF} + \#FT \cdot w_{FT} + \#FF \cdot w_{FF}$

$= 15 \cdot (-1) + 20(100) + 0 - 10(45)$

$= 2000 - 465 = \boxed{1535}$

Since B has the lower cost, we'd split on B.

Problem 3. AdaBoost (20 points)

Let us consider the AdaBoost algorithm in which all the data points are initially given the uniform weights. Given bellow is a dataset with feature X and response Y.

ID	1	2	3	4	5	6	7	8	9	10
X	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Y	1	1	1	-1	-1	-1	-1	-1	1	1

We will now consider three weak classifiers (Hypotheses) H1, H2, and H3.

$$H1:\ if\ X \le 0.35 \rightarrow Y = 1, else\ Y = -1$$

$$H2:\ if\ X \le 0.75 \rightarrow Y = -1, else\ Y = 1$$

$$H3:\ if\ X \le 0.3\ or\ X \ge 0.95 \rightarrow Y = 1, else\ Y = -1$$

For each of the three weak classifiers, answer the following questions. Note that each weak classifier is independent and will start with the initial weight.

(1) Compute the weights of all the instances after the first round of the AdaBoost algorithm. No need to perform normalization of the weights at the end of the iteration.
(2) Clearly specify the data instances which will be reweighted after the first iteration. The answer can be 'None of them', 'All of them', or you can list the specific data instances.

3.1)

ID	1	2	3	4	5	6	7	8	9	10
X	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Y	1	1	1	-1	-1	-1	-1	-1	1	1

step 1) determine where to split

→ Find smallest Gini index of classifiers

Gini H1:

	+	-
T	3	0
F	2	5

T label) $1 - (3/3)^2 = 0$

F label) $1 - (2/7)^2 - (5/7)^2 = 0.5102$

Gini H1) $(0)(3/10) + (0.5102)\ 7/10$

$= \boxed{0.3571}$

Gini X_2:

	+	−
T	3	4
F	2	1

T label) $1 - (3/7)^2 - (4/7)^2$
$= 0.4898$

F label) $1 - (2/3)^2 - (1/3)^2$
$= 0.4444$

Gini X_2) $0.4898\left(\frac{7}{10}\right) + 0.4444\left(\frac{3}{10}\right)$
$= \boxed{0.4762}$

Gini X_3:

	+	−
T	4	0
F	1	5

T label) $1 - 1 = 0$

F label) $1 - (1/6)^2 - (5/6)^2 = 0.2778$

Gini X_3) $0.2778\left(\frac{6}{10}\right) = \boxed{0.1668}$

→ Thus split on X_3. Taking our previous count table.

	+	−
T	4	0
F	1	5

⇒ 9 correctly classified
1 incorrectly classified

Since we have uniform weights

$\rightarrow 1/10 \cdot 1 = err_1 = 1/10$

$\rightarrow \alpha_1 = 1/2 \ln\left(\frac{1-0.1}{0.1}\right)$

$= 1/2 \ln(1/9) = 1.0986$

Incorrectly classified weights:

$0.1 \cdot e^{1.0986} = 0.300$

Correctly classified weights:

$0.1 \cdot e^{-1.0986} = 0.03333$

3.2)

All data will be reweighted, with the incorrectly classified points increasing in weight, & correctly classified points having their weight decreased, as to put more weight on points yet to be correctly classified for future classifiers.

#4)

Problem 4. K-nearest Neighbor Classification (40 points; 10+30)

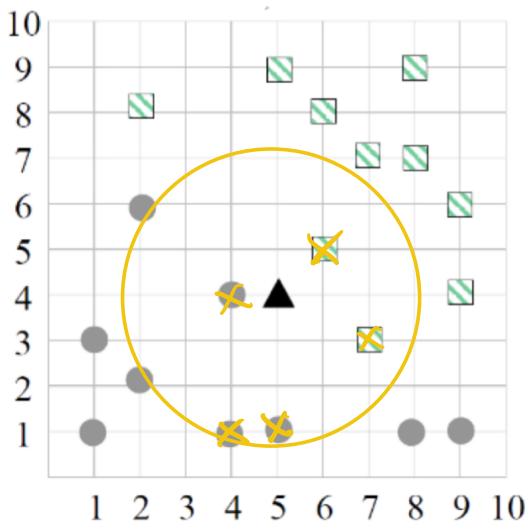
Part I. Consider the following 2-dimensional dataset (Treat the squares as + and circles as –). Classify the test point (triangle) using the following two strategies:

(1) 5- nearest neighbor
(2) Manhattan distance weighted 3-nearest neighbor (the weight is $1/d^2$)

4.1.1) 5-nearest neighbors reveals

2 + □

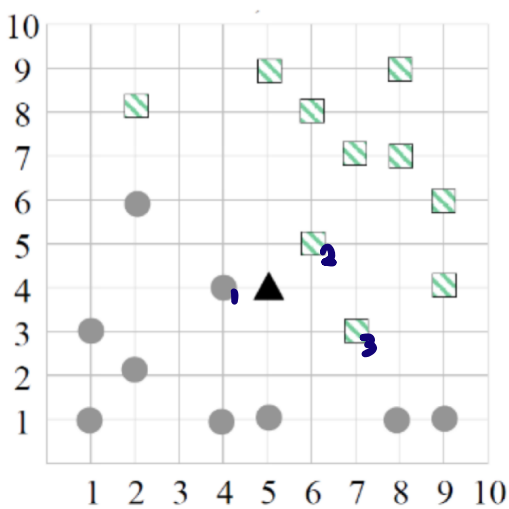
3 − ○



Thus since there are more − classified datapoints close to our point then +, our datapoint is classified as −.

4.1.2) 3-nearest neighbors reveals a weighted sum of

$$-\left(\frac{1}{1^2}\right)+\left(\frac{1}{2^2}\right)+\left(\frac{1}{3^2}\right)$$



→ $-1 + 1/4 + 1/9$

→ $-1 + 9/36 + 4/36$

→ $-\frac{23}{36}$ thus our point is classified as −.

Note: Our 3rd point can be chosen to be + or −, since there is a point for each data type 3 away from our relevant point. Regardless of our choice, our classification would remain −.

4.2.1)

```
Probabilities for KNN:
[[0.         1.        ]
 [0.66666667 0.33333333]
 [1.         0.        ]
 [0.         1.        ]
 [1.         0.        ]
 [0.         1.        ]
 [1.         0.        ]
 [1.         0.        ]
 [1.         0.        ]
 [1.         0.        ]
 [0.         1.        ]
 [1.         0.        ]
 [1.         0.        ]
 [1.         0.        ]
 [0.33333333 0.66666667]
 [0.         1.        ]
 [0.         1.        ]
 [1.         0.        ]
 [1.         0.        ]
 [1.         0.        ]]
Prediction for 3 nearest neighbors [1 0 0 1 0 1 0 0 0 0 1 0 0 0 1 1 1 0 0 0]
```

← probability prediction

Classification of points

4.2.2)

```
Probabilities for ISDW_KNN:
[[0.         1.        ]
 [0.66165411 0.33834589]
 [1.         0.        ]
 [0.         1.        ]
 [1.         0.        ]
 [0.         1.        ]
 [1.         0.        ]
 [1.         0.        ]
 [1.         0.        ]
 [1.         0.        ]
 [0.         1.        ]
 [1.         0.        ]
 [1.         0.        ]
 [1.         0.        ]
 [0.3229923  0.6770077 ]
 [0.         1.        ]
 [0.         1.        ]
 [1.         0.        ]
 [1.         0.        ]
 [1.         0.        ]]
Prediction for ISD weighted KNN    [1 0 0 1 0 1 0 0 0 0 1 0 0 0 1 1 1 0 0 0]
```

← probability prediction

note: each classified point remains the same as 4.2.1

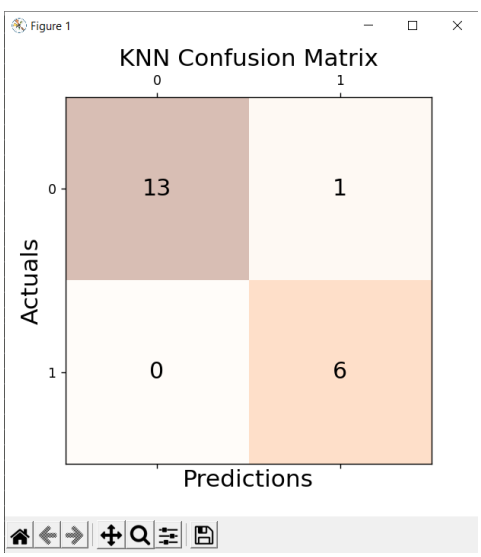
classification of points

4.2.3)

```
Prediction for 3 nearest neighbors [1 0 0 1 0 1 0 0 0 0 1 0 0 0 1 1 1 0 0 0]
Prediction for ISD weighted KNN    [1 0 0 1 0 1 0 0 0 0 1 0 0 0 1 1 1 0 0 0]
Precision: 0.857
Recall: 1.000
F-measure: 0.923
Accuracy: 0.950
```

Since both models classify our test set to the same classes, all statistical data pertaining to the binary classification of points will be the same. Thus I provide it only once.

As for which model has better performance, looking at the points our models were least certain of:



KNN

```
[0.66666667 0.33333333]
[0.33333333 0.66666667]
```

✓
✗

KNN-w(1/d²)

```
[0.66165411 0.33834589]
[0.3229923  0.6770077 ]
```

✓
✗

✗ → misclassified
✓ → correctly classified

Our KNN-w(1/d²) misclassified with higher certainty, and correctly classified with lower certainty hard to classify points, than the normal KNN model.

Thus though it is by a very very small margins our KNN provides better classification performance.

CS 559: Machine Learning

Homework Assignment 3

Due Date: Wednesday 6:30 PM, March 30, 2022 **Total: 100 points**

Problem 1. Decision Trees based on Entropy (20 points)

Consider the following dataset for a binary classification problem and answer the following questions.

Instance	Attribute 1 (a_1)	Attribute 2 (a_2)	Class
1	T	1.0	+
2	T	6.0	+
3	T	5.0	-
4	F	4.0	+
5	F	7.0	-
6	F	3.0	-
7	F	8.0	-
8	T	7.0	+
9	F	5.0	-

(1) Among the two attributes, show which attribute will be chosen as the first splitting for decision tree using **information gain (gain in the entropy).** Show all split points for all attributes. For the continuous attribute a_2, please perform the binary decision by considering each distinct value as the splitting threshold and find the best one.

Please write down every step including the calculation of entropy of the parent, count matrix after splitting, entropy of the children after splitting, and the information gain of the attributes at each split.

(2) What happens if we use "Instance" as another attribute? Do you think this attribute should be used for a decision in the tree?

Problem 2. Decision Trees based on GINI index (20 points)

The following table summarizes a dataset with two attributes A, B, and two class labels +, −. The original dataset had 5 attributes, but only 2 of the most informative attributes are selected for further analysis.

A	B	Class Label	
		+	−
T	T	0	20
T	F	20	10
F	T	15	0
F	F	0	35

Cost Matrix	Attribute Value		
Actual Class		T	F
	+	-1	100
	−	0	-10

(1) According to **GINI index**, which attribute would be chosen as the first splitting attribute?
(2) Use the given cost matrix and decide the first splitting attribute. The total cost is the metric for splitting, and this question is independent to question (1). Note that this question is independent to question (1). You will need to use both tables provided above to calculate the cost of splitting A and B, and choose the attribute with smaller cost to split.

Please write down every step including the calculation of the count matrix after splitting, GINI index of the parent, GINI index of the children after splitting, and the cost of splitting based on each attribute.

Problem 3. AdaBoost (20 points)

Let us consider the AdaBoost algorithm in which all the data points are initially given the uniform weights. Given bellow is a dataset with feature X and response Y.

ID	1	2	3	4	5	6	7	8	9	10
X	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Y	1	1	1	-1	-1	-1	-1	-1	1	1

We will now consider three weak classifiers (Hypotheses) H1, H2, and H3.

$$H1\text{: } if\ X \leq 0.35 \rightarrow Y = 1, else\ Y = -1$$

$$H2\text{: } if\ X \leq 0.75 \rightarrow Y = -1, else\ Y = 1$$

$$H3\text{: } if\ X \leq 0.3\ or\ X \geq 0.95 \rightarrow Y = 1, else\ Y = -1$$

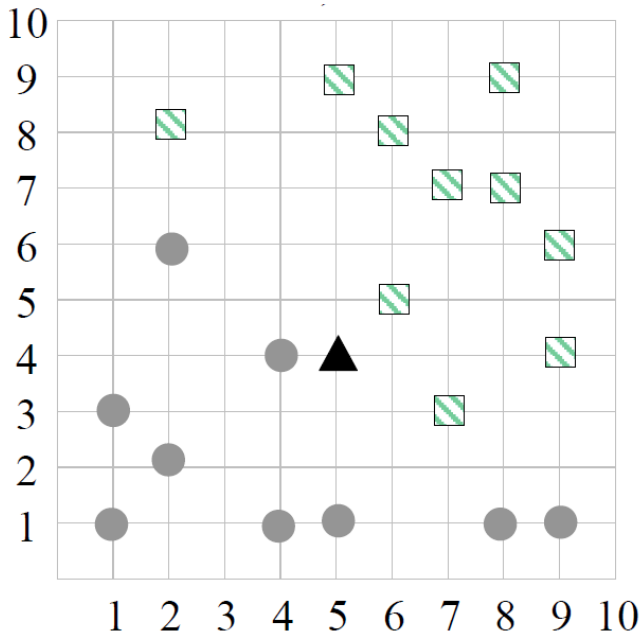
For each of the three weak classifiers, answer the following questions. Note that each weak classifier is independent and will start with the initial weight.

(1) Compute the weights of all the instances after the first round of the AdaBoost algorithm. No need to perform normalization of the weights at the end of the iteration.
(2) Clearly specify the data instances which will be reweighted after the first iteration. The answer can be 'None of them', 'All of them', or you can list the specific data instances.

Problem 4. K-nearest Neighbor Classification (40 points; 10+30)

Part I. Consider the following 2-dimensional dataset (Treat the squares as + and circles as –). Classify the test point (triangle) using the following two strategies:

(1) 5- nearest neighbor
(2) Manhattan distance weighted 3-nearest neighbor (the weight is $1/d^2$)



Part II. Consider the three-dimensional data set in **train.csv**.

(1) Classify the data points in **test.csv** according to their 3-nearest neighbors. Also give the probability estimates for the final decision.
(2) Do the same for the Euclidean distance weighted 3-nearest neighbors ($1/d^2$). Doe the predicted label for each point remain the same as that in question (1)?
(3) In the test.csv, the true class labels are also provided. Construct the confusion matrix and calculate Accuracy, Precision, F-measure for questions (1) and (2). From your results, which method gives better performance?

You can implement by yourself or use the sklearn packages to implement KNN. Please submit your source code, outputs, and the details of the calculation in (3) for full credit.