Homework Assignment 3

Problem 1:

1) For attribute 1 (a_1), the count matrix is

a_1	+ve	-ve
T	3	1
F	1	4

$S = [4,5]$ $\quad$ $\text{Entropy}(S) = -\frac{4}{9}\log_2\frac{4}{9} - \frac{5}{9}\log_2\frac{5}{9}$

$= 0.52 + 0.471$

$= 0.99$

$S_T = [3,1]$ $\quad$ $\text{Entropy}(S_T) = \frac{-3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}$

$= 0.31 + 0.5$

$= 0.81$

$S_F = [1,4]$ $\quad$ $\text{Entropy}(S_F) = \frac{-1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{1}{4}$

$= 0.46 + 0.25$

$= 0.72$

Information gain $= \text{Entropy}(S) - \sum_{i=1}^{k} \frac{n_i}{n}\,\text{Entropy}(i)$

$= 0.99 - \left[\frac{4}{9}(0.81) + \frac{5}{9}(0.72)\right]$

$= 0.99 - [0.36 + 0.39]$

$= 0.235$

For attribute 2 (a_2), we consider ~~#~~ each ~~attribute~~ distinct value as the splitting threshold & find the best one.

For 1.0,

$S[4,5]$ Entropy $(S) = 0.99$

$S_{\leq} = [1,0]$ Entropy $(S_{\leq}) = 0$

$S_{>} = [3,5]$ entropy $(S_{>}) = -\frac{3}{8} \log_2 \frac{3}{8} - \frac{5}{8} \log_2 \frac{5}{8}$

$= 0.95$

$I.G. = 0.99 - [0 + \frac{8}{9} 0.95]$

$= \cancel{0.99-0}$

$= 0.99 - 0.848 = 0.143$

For 3.0,

$S_{\leq} = [1,1] = 1$

$S_{>} = [3,4]$. Entropy $(S_{>=}) = \frac{3}{7} \log_2 \frac{3}{7} + \frac{4}{7} \log_2 \frac{4}{7}$

$= 0.985$

$I.G. = 0.99 - \left[\frac{2}{9}(1) + \frac{7}{9}(0.985)\right]$

$= 0.003$

For 4.0,

$S_{<=}$ [2,1] Entropy($S_{<=}$) = 0.9183

$S_{>}$ [2,4] Entropy($S_{<}$) = 0.9183

$\therefore$ I.G. = $0.99 - \left[\frac{3}{9}(0.9183) + \frac{6}{9}(0.9183)\right]$

= 0.0727

For 5.0,

$S_{\cancel{x}}$ = [2,3] Entropy($S_{<=}$) = 0.970

$S_{>}$ = [2,2] Entropy($S_{>}$) = 1

I.G. = $0.99 - \left[\frac{5}{9}(0.970) + \frac{4}{9}(1)\right]$

= 0.007

For 6.0,

$S_{<=}$ [3,3] Entropy($S_{<=}$) = 1

$S_{>}$ = [1,2] Entropy($S_{>}$) = 0.918

I.G. = $0.99 - \left[\frac{6}{9}(1) + \frac{3}{9}(0.918)\right]$

= 0.0183

For 7.0,

$S_{<=} = [4,4]$ Entropy$(S_{<=}) = 1$

$S_{>} = [0,1]$ Entropy$(S_{>}) = 0$

$$I.G. = 0.99 - \left[\frac{8}{9}(1) + \frac{1}{9}(0)\right]$$

$$= 0.10211$$

For 8.0,

$S_{<=} [4,5]$ Entropy$(S_{<=}) = 0.991$

$S_{>} = [0,0]$ Entropy$(S_{>}) = 0$

$$I.G. = 0.99 - \left[\frac{9}{9}(0.991)\right]$$

$$= 0$$

Among attributes 1 (a_1) and 2 (a_2), the attribute a_1 has a greater information gain (0.235) compared to the information gain for attribute 2 (0.012), hence a_1 is chosen as the first splitting attribute for decision tree.

2) The instance is the index for the dataset. It should not be ~~added~~ used as an attribute for a decision in the tree since it does not improve the performance and decision making of the decision tree and does not contribute in the finding of the best split.

Problem 2:

1) For attribute A, the count matrix is

A	+ve	-ve
T	20	30
F	15	35

$$\text{Gini (Parent)} = 1 - \sum [p(C_j|t)]^2$$

$$= 1 - \left[\left(\frac{50}{100}\right)^2 + \left(\frac{50}{100}\right)^2\right]$$

$$= 1 - \left[\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right]$$

$$= 1 - 0.5$$

$$= 0.5$$

$$Gini(T) = 1 - \left[\left(\frac{20}{50}\right)^2 + \left(\frac{30}{50}\right)^2 \right]$$
$$= 1 - \left[\frac{4}{25} + \frac{9}{25} \right]$$
$$= 1 - \frac{13}{25}$$
$$= \frac{120}{250}$$

$$Gini(F) = 1 - \left[\left(\frac{15}{50}\right)^2 + \left(\frac{35}{50}\right)^2 \right]$$
$$= 1 - \left[\frac{225}{2500} + \frac{1225}{2500} \right]$$
$$= 1 - \frac{1450}{2500}$$
$$= \frac{105}{250}$$

$$\text{Total Gini Impurity} = \sum_{i=1}^{k} \frac{n_i}{n} \, Gini(t)$$
$$= \frac{50}{100} \left[\frac{120}{250} + \frac{105}{250} \right]$$
$$= 0.5 \, [0.9]$$
$$= 0.45$$

For attribute B, the cost matrix is

B	+ve	-ve
T	15	20
F	20	45

$$\text{Gini (Parent)} = 1 - \left[\left(\frac{35}{100}\right)^2 + \left(\frac{65}{100}\right)^2\right]$$

$$= 1 - \left[\frac{1225}{10000} + \frac{4225}{10000}\right]$$

$$= 1 - \left[\frac{5450}{10000}\right]$$

$$= 0.455$$

$$\text{Gini}(T) = 1 - \left[\left(\frac{15}{35}\right)^2 + \left(\frac{20}{35}\right)^2\right]$$

$$= 1 - \left[\frac{225}{1225} + \frac{400}{1225}\right]$$

$$= 1 - \left[\frac{625}{1225}\right]$$

$$= 0.49$$

$$\text{Gini}(F) = 1 - \left[\left(\frac{20}{65}\right)^2 + \left(\frac{45}{65}\right)^2\right]$$

$$= 1 - \left[\frac{400}{4225} + \frac{2025}{4225}\right]$$

$$= 1 - \left[\frac{2425}{4225}\right]$$

$$= 0.43$$

$$\text{Total Gini Impurity} = \frac{35}{100}(0.49) + \frac{65}{100}(0.43)$$

$$= 0.17 + 0.26$$

$$= 0.449$$

Since, attribute B has a lower gini impurity, it is considered as the first splitting attribute.

2) For attribute A, the total cost of splitting can be calculated using the count and the cost matrix.

$$\text{Total cost} = (-1)(20) + 0(30) + 100(15) + (-10)(35)$$
$$= -20 + 0 + 1500 - 350$$
$$= 1130$$

For attribute B, the total cost is.

$$\text{Total cost} = (-1)(15) + (0)(20) + 100(20) + (-10)(45)$$
$$= -15 + 0 + 2000 - 450$$
$$= 1535$$

Since the total cost of attribute A is smaller than the total cost of attribute B, we chose attribute A to split.

Problem 3:

1. For H1,

ID	X	Y	H1(x)	Weight	New Weight
1	0.1	1	1	0.1	0.05
2	0.2	1	1	0.1	0.05
3	0.3	1	1	0.1	0.05
4	0.4	-1	-1	0.1	0.05
5	0.5	-1	-1	0.1	0.05
6	0.6	-1	-1	0.1	0.05
7	0.7	-1	-1	0.1	0.05
8	0.8	-1	-1	0.1	0.05
9	0.9	1	-1	0.1	1.999
10	1	1	-1	0.1	1.999

After the first round of the AdaBoost algorithm, we compute the amount of say using the total error.

$$\alpha_1 = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

$$\varepsilon_t = \text{Total error} = 0.1 + 0.1 = 0.2$$

$$\alpha_1 = \frac{1}{2} \ln\left(\frac{1-0.2}{0.2}\right)$$

$$= \frac{1}{2} \ln(4)$$

$$= 0.693$$

For correctly classified examples,

$$D_2(i) = 0.1 \times e^{-0.693}$$

$$= 0.1 \times 0.5 = 0.05$$

For incorrectly classified examples,

$$D_2(i) = 0.1 \times e^{0.693} = 0.1 \times 1.999$$
$$= 0.199$$

For H2,

ID	X	Y	H2(X)	Weight	New Weight
1	0.1	1	-1	0.1	0.122
2	0.2	1	-1	0.1	0.122
3	0.3	1	-1	0.1	0.122
4	0.4	-1	-1	0.1	0.0818
5	0.5	-1	-1	0.1	0.0818
6	0.6	-1	-1	0.1	0.0818
7	0.7	-1	-1	0.1	0.0818
8	0.8	-1	1	0.1	0.122
9	0.9	1	1	0.1	0.0818
10	1	1	1	0.1	0.0818

$$\alpha_1 = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

$$\varepsilon_1 = 0.1 + 0.1 + 0.1 + 0.1$$
$$= 0.4$$

$$\alpha_1 = \frac{1}{2} \ln\left(\frac{1-0.4}{0.4}\right)$$
$$= \frac{1}{2} \ln(1.5)$$
$$= 0.20$$

For correctly classified examples,

$$D_1(i) = 0.1 \times e^{-0.20}$$
$$= 0.1 \times 0.818$$
$$= 0.0818$$

For incorrectly classified examples,

$$D_1(i) = 0.1 \times e^{0.20}$$
$$= 0.1 \times 1.22$$
$$= 0.122$$

For H3,

ID	X	Y	H3(X)	Weight	New Weight
1	0.1	1	1	0.1	0.033
2	0.2	1	1	0.1	0.033
3	0.3	1	1	0.1	0.033
4	0.4	-1	-1	0.1	0.033
5	0.5	-1	-1	0.1	0.033
6	0.6	-1	-1	0.1	0.033
7	0.7	-1	-1	0.1	0.033
8	0.8	-1	-1	0.1	0.033
9	0.9	1	-1	0.1	0.299
10	1	1	1	0.1	0.033

$$\varepsilon_1 = 0.1$$

$$\alpha_1 = \frac{1}{2} \ln\left(\frac{1-0.1}{0.1}\right)$$
$$= \frac{1}{2} \ln(9)$$
$$= 1.0986$$

For correctly classified examples,

$$D_1(i) = 0.1 \times e^{-1.0986}$$
$$= 0.1 \times 0.33$$
$$= 0.033$$

For incorrectly classified examples,

$$D_2(i) = 0.1 \times e^{1.0986}$$
$$= 0.1 \times 2.99$$
$$= 0.299$$

2. After the first iteration, all the data instances will be reweighted based on whether they are classified correctly or not. The incorrectly classified instances are given more weight in comparison to the correctly classified instances for the next iteration.
For H1, data instances 9 and 10 are incorrectly classified and hence given more weightage for next round. The other data instances are given less weightage for the next round since they are correctly classified.
Similary, for H_2, data instances 1,2,3 and 8 are reweighted

with more weight since they are incorrectly classified and the correctly classified instances are reweighted with less weights.
For H3, only instance 9 is incorrectly classified and hence given more weightage compared to other instances.

Problem 4:

1. Considering the 6 nearest neighbours from the test point (5,4), we calculate the Euclidean distance from the test point to find 5 nearest neighbors (k).

Point 1 (4,4), (-ve)

$$d = \sqrt{\sum_{m=1}^{2} (x_{im} - x_{jm})^2}$$

$$= \sqrt{(4-5)^2 + (4-4)^2} = 1$$

Point 2 (6,5), (+ve)

$$d = \sqrt{(6-5)^2 + (5-4)^2} = 1.41$$

Point 3 (7,3), (+ve)

$$d = \sqrt{(7-5)^2 + (3-4)^2} = 2.2$$

Point 4 (5,1), (-ve)

$$d = \sqrt{(5-5)^2 + (1-4)^2} = 3$$

Point 5 (4,1), (-ve)

$$d = \sqrt{(4-5)^2 + (1-4)^2} = 3.16$$

Point 6 (9,4), (+ve)

$$d = \sqrt{(9-5)^2 + (4-4)^2} = 4$$

After computing the distances, the k=5 nearest neighbours are

P1	-ve
P2	+ve
P3	+ve
P4	-ve
P5	-ve

Using the majority vote of class labels among 5 nearest neighbours, the test point is classified as -ve.

2) Considering the 3 nearest neighbours, we compute the manhattan distance and the associated weight ~~for th~~ from the test point (5,4).

Point 1 (4,4) (-ve)

$$d = \sum_{m=1}^{n} |x_{im} - x_{jm}|$$

$$d = |4-5| + |4-4| = 1$$

$$w = \frac{1}{d^2} = \frac{1}{1} = 1$$

Point 2 (6,5) (+ve)

$$d = |6-5| + |5-4|$$
$$= 1+1 = 2$$

$$w = \frac{1}{2^2} = \frac{1}{4} = 0.25$$

Point 3 $(7,3)$
(+ve)

$$d = |7-5| + |3-4|$$
$$= 2+1$$
$$= 3$$
$$w = \frac{1}{3^2} = \frac{1}{9} = 0.11$$

+ve sum of weights = 0.36

−ve sum of weights = 1

Since the sum of weights for −ve label (1) is greater than the sum of weights for +ve label (0.36), the test point is classified as −ve.

Problem 4

Part II

2. After performing the classification for 3 nearest neighbors using the Euclidean distance, we see that predicted labels for weighted Euclidean based model is almost similar to the predicted labels for Manhattan based model. The Manhattan model correctly classifies all the data instances whereas the Euclidean model correctly classifies all but one test data instance into their class labels.

3. Below are the evaluation metrics for the two models.

For Manhattan model:

Confusion matrix		True Class	
		Positive	Negative
Predicted class	Positive	14	0
	Negative	0	6

Evaluation Metric	Value
Accuracy	100%
Precision	1
F-Measure	1

For weighted Euclidean model:

Confusion matrix		True Class	
		Positive	Negative
Predicted class	Positive	13	1
	Negative	0	6

Evaluation Metric	Value
Accuracy	100%
Precision	0.857143
F-Measure	0.923077

The Manhattan model has an accuracy of 100%. The confusion matrix also shows that the positive and negative labels are assigned correctly. We have a precision and F-measure of 1.0. This shows that we have a low false positive rate, and the model predicts accurately.

For Euclidean model, one of the class labels are incorrectly classified and hence we have a precision and F-measure of less than one. We can see from the confusion matrix that the model incorrectly classifies one true positive class label as false positive.

From the evaluation metrics, we can see that the Manhattan model performs better than the weighted Euclidean model since the Manhattan model classifies all the test data accurately.