Assignment 7

Task 1:
Part a:
80 — Wait
20 — Not wait
K1 = 80
K2 = 20
K = K1+K2 = 100

Entropy = H(K1/K, K2/K) = -(K1/K)log2(K1/K) - (K2/K)log2(K2/K)
= -(0.8)log2(0.8) - (0.2)log2(0.2)
= -(0.8)(-0.32193) - (0.2)(-2.32193)
= 0.25754 + 0.46439
= 0.72193

Part b:
K1 = 80, K2 = 20
K3 = 20, K4 = 15
K5 = 60, K6 = 5
K = 100
K3 + K4 = 35
K5 + K6 = 65

To find the information gain of the weekend test at node A, the following is needed:
Entropy of node A (already found) = 0.72193
Weight of node B = [(K3+K4)/K] = 35/100 = 0.35
Entropy of node B = H(K3/(K3+K4), K4/(K3+K4)) = -(K3/(K3+K4))log2(K3/(K3+K4)) - (K4/(K3+K4))log2(K4/(K3+K4)) = -(20/35)log2(20/35) - (15/35)log2(15/35) = -(0.57143)(-0.80735) - (0.42857)(-1.22239) = 0.46134 + 0.52388 = 0.98522
Weight of node C = [(K5+K6)/K] = 65/100 = 0.65
Entropy of node C = H(K5/(K5+K6), K6/(K5+K6)) = -(K5/(K5+K6))log2(K5/(K5+K6)) - (K6/(K5+K6))log2(K6/(K5+K6)) = -(60/65)log2(60/65) - (5/65)log2(5/65) = -(0.92308)(-0.11548) - (0.07692)(-3.70044) = 0.10660 + 0.28464 = 0.39124

Information gain at node A = Entropy(A) - Weight(B)*Entropy(B) - Weight(C)*Entropy(C)
= 0.72193 - (0.35)(0.98522) - (0.65)(0.39124)
= 0.72193 - 0.344827 - 0.254306
= 0.122797

Given the new information, the information gain of the weekend test at node A is equal to 0.122797.

Part c:
At Node E, all the people will answer yes to the weekend test (they already answered yes at node A to even get into the left subtree). This means that the classes at Node E are completely unevenly split, which means that Node E has an information gain of zero. The amount of people that are actually at Node E is irrelevant.

Part d:
Start at Node A
Weekend? No, move down to Node C.
Raining? Yes, move down to Node F.

This test case ends up in node F, and outputs the answer “will wait”.

Part e:
Start at Node A
Weekend? Yes, move down to node B
Hungry? No, move down to node E
Weekend? Yes, move down to node H

This test case ends up in node H, and outputs the answer “will not wait”.

Task 2:
For the sake of saving space on this document, all weight and entropy calculations are included on separate (scanned) pages. These pages are contained in the documents task2a.pdf, task2b.pdf, task2c.pdf, for attributes A, B, and C, respectively. For each node, the odd-numbered K values represent patterns that are of type X, and the even-numbered K values represent patterns that are of type Y. After calculating the first answer, several calculations on the other attributes were mathematically equivalent to calculations from the first attribute, so for the sake of brevity, I just used the number I previously calculated if the calculations are mathematically equivalent, instead of recalculating the same number each time.

Information gain at root of A = 1 - (0.3)(0) - (0.4)(0.81128) - (0.3)(0.91829) = 1 - 0 - 0.324512 - 0.275487 = 0.4

Information gain at root of B = 1 - (0.4)(0.81128) - (0.4)(0.81128) - (0.2)(1) = 1 - 0.324512 - 0.324512 - 0.2 = 0.150976

Information gain at root of C = 1 - (0.5)(0.72193) - (0.4)(0.81128) - (0.1)(0) = 1 - 0.360965 - 0.324512 - 0 =0.314523

At the root, the attribute that achieves the highest information gain (based on the training examples) is Attribute A, with an information gain of 0.4.

Task 3:
Part a:
The lowest possible entropy value at node N is zero, which means that all the training examples are of the same class. The highest possible entropy value is log2(C), or the base-2 logarithm of C, where C = number of classes. In this case, C = 4, which means the highest possible entropy value is log2(4) = 2. This would happen when all the classes are completely equally split.

Part b:
The lowest possible information gain at Node N is also zero. When the information gain is zero, the classes are completely unevenly split at N. It can also be zero if the classes are evenly split at each node (N and it's immediate children).

The highest possible information gain value would be equal to the highest possible entropy value at Node N. This would happen if the classes are evenly split at Node N, but then completely unevenly split in Node N's immediate children:

Entropy(N) = log2(4) = 2 (even split)
Entropy(children of N) = 0 (completely uneven split)

With these values, the weight of the children nodes doesn't matter (it's getting multiplied by zero). So, in this case, the highest possible information gain is equal to:

Entropy(N) - Sum(Weight(children of N)*Entropy(children of N)) = Entropy(N) - 0 = Entropy(N) = log2(4) = 2.

The highest possible information gain is 2. The lowest possible information gain is zero.

Task 4:
Assuming that the current accuracy (28%) remained constant, then the accuracy could be increased if we just force the classifier to return the opposite of whatever it predicts (if it predicts loss, return a win as the classifiers answer). Assuming again that the accuracy will remain a constant, the new accuracy will be 72%, and the classifier will be guaranteed to have an accuracy greater than 60%.

Task 5:
Based on the given information:
P(Class A) = 2/5 = 0.4
P(Class B) = 3/5 = 0.6

The mean and standard deviation for each attribute will change depending on whether we are considering class A or class B.

P(Attribute 1 = X | Class A):
Mean = 33/2 = 16.5
(StD)^2 = 1/(2-1) * ((15-16.5)^2 + (18-16.5)^2)
= 1 * (2.25 + 2.25) = 4.50
StD = 2.12132

P(Attribute 1 = X | Class A) = 1/(sqrt(2pi) * 2.12132) * e^-((X-16.5)^2/(2*(2.12132)^2))
= (0.13298) * e^(-(X-16.5)^2 / 4.5)

P(Attribute 2 = X | Class A):
Mean = 60/2 = 30
(StD)^2 = 1/(2-1) * ((28-30)^2 + (32-30)^2)
= 1 * (4+4) = 8
StD = 2.82843

P(Attribute 2 = X | Class A) = 1/(sqrt(2pi) * 2.82843) * e^(-(X-30)^2 / (2 * 8))
= (0.09974) * e^(-(X-30)^2 / 16)

P(Attribute 1 = X | Class B):
Mean = 77/3 = 25.66667
(StD)^2 = (1/2) * ((20-25.66667)^2 + (32-25.66667)^2 + (25-25.66667)^2)
= (0.5) * (32.11115 + 40.11107 + 0.44445) = 36.33334
StD = 6.02771

P(Attribute 1 = X | Class B) = 1/(sqrt(2pi) * 6.02771) * e^(-(X-25.66667)^2 / (2 * 36.33334)
= (0.06618) * e^(-(X-25.66667)^2 / 72.66668)

P(Attribute 2 = X | Class B):
Mean = 40/3 = 13.33333
(StD)^2 = (1/2) * ((10-13.33333)^2 + (15-13.33333)^2 + (15-13.33333)^2)
= (0.5) * (11.11109 + 2.77779 + 2.77779) = 8.33334
StD = 2.88675

P(Attribute 2 = X | Class B) = 1/(sqrt(2pi) * 2.88675) * e^(-(X-13.33333)^2 / (2 * 8.33334)
= (0.13820) * e^(-(X-13.33333)^2 / 16.66668)