Alfredo Reina Corona

FINAL GRADE: 100%

COMMENTS:
“task 3: calculation errors -1 point (the info gain calculation is incorrect)”

Task 1:

Each run ~10 times

decision_tree.py pendigits_training pendigits_test optimized 50 = .8393(1 minute)

decision_tree.py pendigits_training pendigits_test 1 50 = .6933 - .7573(5 Seconds)

decision_tree.py pendigits_training pendigits_test 3 50 = .8209 – 8789(1 minute)

Task 2:

a) H(0.8, 0.2) = -0.8log2(0.8) – 0.2log2(0.2) = .7219

b) Entropy(Wait) = - [(20/35) * log2(20/35) + (15/35) * log2(15/35)] = .4134

Entropy(NW) = - [(60/65) * log2(60/65) + (5/65) * log2(5/65)] = .2058

Weight(W) = 35 / 100

Weight(NW) = 65 / 100

Gain = Initial Entropy (Entropy(A)) - [Weight(W) * Entropy(W) + Weight(NW) * Entropy(NW)]

= 0.44354

c) Gain = -0.3291

d) Node F, Will Wait

e) Node E, Will Not Wait

Task 3:

A1 = x,x,x

A2 = x,y,y,y

A3 = x,y,y

H(E) = 1 → even amount of ‘X’ and’ Y’

H((A1/10),(A2/10),(A3/10)) = H(E) – [(3/10)*(1.0 * log2(1) + 0*log2(0))+

(4/10)*(-0.25*log2(0.25) -0.75*log2(0.75)) +

(3/10*(-0.33*log2(0.33) – 0.67*(log2(0.67))] = Gain_A

H(E)-(0 +(.32451) + (-.04221))

1- .2823 = .7177

B1 = x,y,y,y

B2 = x,x,x,y

B3 = x,y

H(E) = 1 → even amount of 'X' and' Y'

H((B1/10),(B2/10),(B3/10)) = H(E) – [(4/10)*(-.25 * log2(.25) - .75*log2(.75))+

(4/10)*(-0.75*log2(0.75) -0.25*log2(0.25)) +

(2/10*(-0.5*log2(0.5) – 0.5*log2(0.5)] = Gain_B

H(E)-((.32451) + (.32451) + .2)

1- .84902 = .15098

C1 = x,y,y,y,y

C2 = x,x,x,y

C3 = x

H(E) = 1 → even amount of 'X' and' Y'

H((C1/10),(C2/10),(C3/10)) = H(E) – [(5/10)*(-.2 * log2(.2) - .8*log2(.80) +

(4/10)*(-0.75log2(0.75) -0.25log2(0.25)) +

(1/10*(-1.0log2(1.0) – 0(log2(0))] = Gain_CH(E)-((.36096) + (.32451) + (0))

1 - .68547 = = .31453

A is the highest so choose it

Task 4:

a. H(A,B,C,D)_max = -(250/1000)log2(250/1000) - (250/1000)log2(250/1000) -

(250/1000)log2(250/1000) -(250/1000)log2(250/1000) = 2

b. H(A,B,C,D)_min = -(1000/1000)log2(1000/1000) - (0/1000)log2(0/1000) -

(0/1000)log2(0/1000) -(0/1000)log2(0/1000) = 0

Task 5:

If your classifier correctly predicts if the home team will win 28% of the time, using the contrapositive, it also means that it incorrectly predicts if the "away" team will lose 72% of the time.

By switching the classifier to predict if the “away” team will lose, and then outputting the opposite of the given answer, you essentially have a classifier that can predict if the “home” team will win 72% of the time.