**ITCS451: Artificial Intelligence Assignment III: Are you ready for the Final?**

# Part I: CSP

Consider the wolf, goat, and cabbage problem (WGC):

“A farmer went to a market and purchased a wolf, a goat, and a cabbage. On his way home, the farmer came to the bank of a river and rented a boat. But crossing the river by boat, the farmer could carry only himself and a single one of his purchases: the wolf, the goat, or the cabbage. If left unattended together, the wolf would eat the goat, or the goat would eat the cabbage. The farmer's challenge was to carry himself and his purchases to the far bank of the river, leaving each purchase intact.”

Answer the following problem:

1. Represent the WGC problem as a CSP problem.

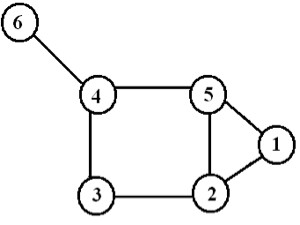
|  |  |
| --- | --- |
| Variable | {wolf, Goat, Cabbage, Human, Boat} |
| Domain for each variable | {0 (Left),1 (Right)} |
| Action | Carry(X) Return -1 as Fail |
| Constrains | - Everyone on right side of the river  - {0,0,0,0,0}  - Wolf in the same side with Goat  - W == G if H == W  - Goat in the same side with Cabbages  - G == C if H ==G |

1. Apply backtracking search on your problem for 3 iterations. Show your work.

|  |  |
| --- | --- |
| Initial State | {W, G, C, H, B} = {0, 0, 0, 0, 0} |
| Iteration 1 | Try Carry (W) fail  {W, G, C, H, B} = {1, 0, 0, 1, 1} |
| Iteration 2 | Try Carry (G)  {W, G, C, H, B} = {0, 1, 1, 0, 1} |
| Iteration 3 | Try Carry (H)  {W, G, C, H, B} = {0, 1, 0, 0, 0} |

# Part II: Local Search

Consider the problem of graph coloring problem with the following graph and setting.



We want to color the above graph using 3 colors (Red Green Blue). Notice that it is impossible.

1. Setup this problem for the hill-climbing search.

|  |  |
| --- | --- |
| Stage | {1, 2, 3, 4, 5, 6} |
| Domain | {R, G, B} |
| Goal | No Neighbor has same color |
| Action | Assign 1 color to the node |
| Heuristics | C = conflict node with same color |
| Constraint | If x and y are connected, then No same color |

1. Perform the hill-climbing search with your setting for 3 iterations. Show your work.

|  |  |  |
| --- | --- | --- |
| 1 R | 1 G | 1 B |
| 2 B | 2 R | 2 G |
| 3 G | 3 B | 3 R |

1. Setup this problem for the GA. You do not need to perform any search for this part.

L

# Part III: ML

For the following data examples, answer the following questions.

Training data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | BMI | BP | Cough | Fever | Disease |
| 1 | overweight | low | No | Yes | yes |
| 2 | normal | low | Yes | Yes | yes |
| 3 | normal | high | No | No | yes |
| 4 | overweight | normal | No | No | yes |
| 5 | underweight | low | No | No | yes |
| 6 | normal | high | Yes | No | yes |
| 7 | normal | normal | No | Yes | yes |
| 8 | underweight | low | Yes | No | yes |
| 9 | underweight | normal | No | No | no |
| 10 | overweight | low | Yes | No | no |
| 11 | underweight | normal | No | Yes | no |
| 12 | underweight | low | Yes | Yes | no |
| 13 | overweight | high | Yes | No | no |
| 14 | overweight | high | Yes | Yes | no |

Test data:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # |  | BMI | BP | Cough | Fever | Disease |
|  | 1 | overweight | high | No | Yes | ? |
|  | 2 | normal | normal | No | No | ? |
|  | 3 | underweight | low | Yes | No | ? |

1. Create a decision tree using the ID3 with the maximum depth of 3. You must show the Info() and the E() of all steps.
2. Extract all rules from your decision tree
3. What are the prediction results from your tree on the test data?
4. What are the prediction results for k-NN using hemming distance with k = 1?
5. What are the prediction results for k-NN using hemming distance with k = 3?
6. What are the prediction results for naïve bayes?
7. What are the pros and cons of evaluating your ML models using a single split vs k-fold cross validation?
8. If you want to know if your current decision tree is in the overfitting or underfitting, how would you check it?

# Answer

1. Info (,

Entropy =

Entropy(S) Or InfoNeed ( = = 0.985

S = []

BMI:

Entropy (Overweight) = = 0.971

S = []

Entropy (Normal) = = 0

S = []

Entropy (Underweight) = = 0.971

S = []

E(BMI) = Entropy (Normal) +

Entropy (Overweight) +

Entropy (Underweight)

=

= 0.693

Gain (BMI) = 0.985 – 0.693 = 0.292

BP:

Entropy(S) = = 0.985

S = []

Entropy (Normal) = = 1

S = []

Entropy (Low) = = 0.918

S = []

Entropy (High) = = 1

S = []

I(BP) = Entropy (Normal) + Entropy (Low) +

Entropy (High)

=

= 0.693

Gain (BMI) = 0.985 – 0.964 = 0.021

Cough:

Entropy(S) = = 0.985

S = []

Entropy (Yes) = = 0.985

S = []

Entropy (No) = = 0.863

S = []

I(BMI) = Entropy (Yes) + Entropy (No)

=

= 0.924

Gain (BMI) = 0.985 – 0.924 = 0.061

Fever:

Entropy(S) = = 0.985

S = []

Entropy (Yes) = = 1

S = []

Entropy (No) = = 0.954

S = []

I(BMI) = Entropy (Yes) + Entropy (No)

=

= 0.973

Gain (BMI) = 0.985 – 0.973 = 0.012

|  |  |
| --- | --- |
| Attributes | Gain |
| BMI | 0.292 |
| BP | 0.021 |
| Cough | 0.061 |
| Fever | 0.012 |

Diagram

Description automatically generated

Overweight:

Entropy (Overweight) = = 0.971

S = []

BP:

Entropy (Normal) = = 0

S = []

Entropy (Low) = = 1

S = []

Entropy (High) = = 0

S = []

E(BP) = Entropy (Normal) + Entropy (Low) +

Entropy (High)

=

= 0.4

Gain (BP) = 0.971 – 0.4 = 0.571

Cough:

Entropy (Overweight) = = 0.971

S = []

Entropy (Yes) = = 0

S = []

Entropy (No) = = 0

S = []

I(BMI) = Entropy (Yes) + Entropy (No)

=

= 0

Gain (BMI) = 0.971 – 0 = 0.971

Fever:

Entropy (Overweight) = = 0.971

S = []

Entropy (Yes) = = 1

S = []

Entropy (No) = = 0.918

S = []

I(BMI) = Entropy (Yes) + Entropy (No)

=

= 0.950

Gain (BMI) = 0.971 – 0.950 = 0.021

|  |  |
| --- | --- |
| Attributes | Gain |
| BP | 0.571 |
| Cough | 0.971 |
| Fever | 0.021 |

Underweight:

Entropy (Underweight) = = 0.971

S = []

BP:

Entropy (Normal) = = 0

S = []

Entropy (Low) = = 0.918

S = []

Entropy (High) = = 0

S = []

E(BP) = Entropy (Normal) + Entropy (Low) +

Entropy (High)

=

= 0.552

Gain (BP) = 0.971 – 0.552 = 0.419

Cough:

Entropy (Overweight) = = 0.971

S = []

Entropy (Yes) = = 1

S = []

Entropy (No) = = 0.918

S = []

I(BMI) = Entropy (Yes) + Entropy (No)

=

= 0.952

Gain (BMI) = 0.971 – 0.952 = 0.019

Fever:

Entropy (Overweight) = = 0.971

S = []

Entropy (Yes) = = 0

S = []

Entropy (No) = = 0.918

S = []

I(BMI) = Entropy (Yes) + Entropy (No)

=

= 0.550

Gain (BMI) = 0.971 – 0.550 = 0.421

|  |  |
| --- | --- |
| Attributes | Gain |
| BP | 0.419 |
| Cough | 0.019 |
| Fever | 0.421 |

Diagram

Description automatically generated

2.

If BMI == “Underweight” && Fever == “Yes” -> Disease = “No”

If BMI == “Underweight” && Fever == “No” -> Disease = “Yes”

If BMI == “Norma” -> Disease = “Yes”

If BMI == “Overweight” && Cough == “Yes” -> Disease = “No”

If BMI == “Overweight” && Cough == “No” -> Disease = “Yes”

3. Test Data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | BMI | BP | Cough | Fever | Disease |
|  | 1 | overweight | high | No | Yes | Yes |
|  | 2 | normal | normal | No | No | Yes |
|  | 3 | underweight | low | Yes | No | Yes |

4. Test Data When K = 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | BMI | BP | Cough | Fever | Disease |
|  | 1 | overweight | high | No | Yes | No |
|  | 2 | normal | normal | No | No | Yes |
|  | 3 | underweight | low | Yes | No | Yes |

5. Test Data When K = 3

2 Yes, 1 No

2 Yes, 1 No

1 Yes, 2 No

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | BMI | BP | Cough | Fever | Disease |
|  | 1 | overweight | high | No | Yes | No |
|  | 2 | normal | normal | No | No | Yes |
|  | 3 | underweight | low | Yes | No | Yes |

1. What are the prediction results for naïve bayes?

= = 0.0084

=

= = 0.0119

Ans No

= = 0.028

=

= = 0

Ans Yes

= = 0.0167

=

= = 0.024

Ans No

* What are the pros and cons of evaluating your ML models using a single split vs k-fold cross validation?

|  |  |  |
| --- | --- | --- |
|  | Pros | Cons |
| **Single split**: | It is better for the small data set. | It may cause some conflict. |
| **K- fold**: | Reduced bias | It is not good for small datasets. It may also elicit an appropriate response. |

* If you want to know if your current decision tree is in the overfitting or underfitting, how would you check it?
* If the training and test results are both close, the model has not overfit.
* If the training result is better than the result, the model has been overfit.
* If the training and test results are both low, the model has underfit.