Northern Michigan University (Marquette Co, MI)

CS470-01-26W: Artificial Intelligence (Andrew A. Poe) Name: ______________________________
Practice Endterm Examination (Exam 2) Monday 20 March 2026 9:00 A.M. EDT

Time: 50 minutes

1. Here is some training data concerning comedians used to determine whether they are worth going to see.

Age	**Experience**	**Rank**	**Nationality**	**Go**
36	**10**	**9**	**UK**	**NO**
42	**12**	**4**	**USA**	**NO**
23	**4**	**6**	**N**	**NO**
52	**4**	**4**	**USA**	**NO**
43	**21**	**8**	**USA**	**YES**
44	**14**	**5**	**UK**	**NO**
66	**3**	**7**	**N**	**YES**
35	**14**	**9**	**UK**	**YES**
52	**13**	**7**	**N**	**YES**
35	**5**	**9**	**N**	**YES**
24	**3**	**5**	**USA**	**NO**
18	**3**	**7**	**UK**	**YES**

What would be some important issues to consider when making a decision tree that will match the training data and be expected to work reasonably well for the data at large?

Age, certainly, but also experience and rank seems to have enough discrete values that making an attribute for each specific value would be overkill. You would probably want to divide age into categories (10's, 20's, 30's, etc., perhaps, and probably experience, too. Even rank could be split into two groups, up to seven, and above seven.

The decision tree algorithm will make the decision tree, but redefining the attributes might help it along.

2. Use the algebraic algorithm to demonstrate that the logical propositions:

P=>R Q=>R PvQ

derive the conclusion

R

understanding that P=>Q is another way of writing (~PvQ).

(~PvR) = (1-P+R)-((1-P)R) = 1-P+PR

(~QvR) = 1-Q+QR

(PvQ) = P+Q-PQ

The product of these three is PR+QR - PQR.

Multiplying this by 1-R yields 0.

3. Use the resolution algorithm on the same problem: to demonstrate that the logical propositions:

P=>R Q=>R PvQ

Northern Michigan University (Marquette Co, MI)

CS470-01-26W: Artificial Intelligence (Andrew A. Poe) Name: ____________________________
Practice Endterm Examination (Exam 2) Monday 20 March 2026 9:00 A.M. EDT

derive the conclusion

R

understanding that P=>Q is another way of writing (~PvQ).

(~PvR) (~QvR) (PvQ) ~R

(~PvR) (~QvR): do not resolve
(~PvR)(PvQ): (QvR)
(~PvR) (~QvR) (PvQ) ~R (QvR)
(~PvR) ~R: ~P
(~PvR) (~QvR) (PvQ) ~R (QvR) ~P
(~PvR) (QvR): do not resolve
(~PvR) ~P: do not resolve
(~QvR) (PvQ): (PvR)
(~PvR) (~QvR) (PvQ) ~R (QvR) ~P (PvR)
(~QvR) (~R): ~Q
(~PvR) (~QvR) (PvQ) ~R (QvR) ~P (PvR) ~Q
(~QvR) (QvR) : R
(~PvR) (~QvR) (PvQ) ~R (QvR) ~P (PvR) ~Q R
(~QvR) ~P: do not resolve
(~QvR) (PvR): do not resolve
(~QvR) ~Q: do not resolve
(~QvR) R: do not resolve
(PvQ) ~R: do not resolve
(PvQ) (QvR): do not resolve
(PvQ) ~P : Q
(~PvR) (~QvR) (PvQ) ~R (QvR) ~P (PvR) ~Q R Q
(PvQ) (PvR) : do not resolve
(PvQ) ~Q : P
(~PvR) (~QvR) (PvQ) ~R (QvR) ~P (PvR) ~Q R Q P
(PvQ) R: do not resolve
(PvQ) Q: do not resolve
(PvQ) P: do not resolve
~R (QvR): Q (already present)
~R ~P: do not resolve
~R (PvR): P (already present)
~R ~Q: do not resolve
~R R: contradiction
Theorem is proven.

4. Design a neural node that accepts ten inputs. One of the ten inputs will be one. The other nine will be zero. The output of the neural node will be an integer from 1 to 10, indicating which of the ten inputs is one.

Input 1 has a multiplier of 1. Input 2 has a multiplier of 2. and so forth. The Hidden input has a multiplier of zero. Since only one of the ten inputs will be 1, the sum of the weighted inputs will be 1 through 10. The node's function need only be the identity function since the weighted sum is already the identity of the non-zero weight.