**CS480 Mid Term**

2.Consider using first-order logic to describe the rules of British crown succession. Assuming the following relations and functions are given. Monarch(x): True if x is the Monarch. Child(x, y): True if y is x’s child. EldestMaleChild(x, y): True if y is x’s eldest male child. EldestFemaleChild(x, y): True if y is x’s eldest female child. Male(x): True if x is male. Female(x): True if x is female. Catholic(x): True if x is a Roman Catholic. Heir(x, y): True if y is the heir of x. Age(x): Function returns the age of x. 1). There is one and only one monarch. 2). Roman Catholic may not be the monarch. 3). The eldest child has the oldest age among one’s male children. 4). If the monarch has male children, the eldest male child will be the heir. 5). If there are no male children, the eldest female child will be the heir.

Solution: The British crown succession using first-order logic are as follows:-

1. There is one and only one monarch: ∃x (Monarch(x)∧∀y (Monarch(y)→x=y))∃x(Monarch(x)∧∀y(Monarch(y)→x=y))
2. A Roman Catholic may not be the monarch: ∀x (Catholic(x)→¬Monarch(x))∀x(Catholic(x)→¬Monarch(x))
3. The eldest child has the oldest age among one’s male children: ∀x ∀y (Child(y,x)∧Male(y)∧∀z (Child(z,x)∧Male(z)→Age(y)≥Age(z)))∀x∀y(Child(y,x)∧Male(y)∧∀z(Child(z,x)∧Male(z)→Age(y)≥Age(z)))
4. If the monarch has male children, the eldest male child will be the heir: ∀x (Monarch(x)∧∃y (Child(y,x)∧Male(y))→Heir(x,EldestMaleChild(x,y)))∀x(Monarch(x)∧∃y(Child(y,x)∧Male(y))→Heir(x,EldestMaleChild(x,y)))
5. If there are no male children, the eldest female child will be the heir: ∀x (Monarch(x)∧¬∃y (Child(y,x)∧Male(y))→Heir(x,EldestFemaleChild(x,y)))∀x(Monarch(x)∧¬∃y(Child(y,x)∧Male(y))→Heir(x,EldestFemaleChild(x,y)))

These formulas capture the specified rules for British crown succession in first-order logic.

4. Let α = A ∨ B and KB = { A ∨ C ∨ D, A ∨ ¬ C ∨ D } Is it the case that KB α? Justify your answer with your method.

Solution:

To determine if KB⊨αKB⊨α (i.e., whether αα logically follows from KBKB), we can use the resolution inference rule. The resolution rule states that if PP and QQ are clauses, and PP contains AA and QQ contains ¬A¬A, then we can resolve on AA and obtain a new clause that is the union of the remaining literals in PP and QQ.

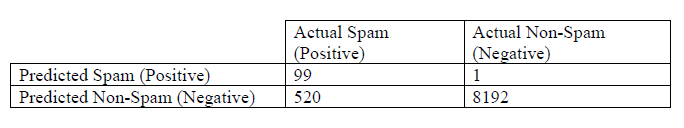
Given α=A∨Bα=A∨B and KB={A∨C∨D,A∨¬C∨D}KB={A∨C∨D,A∨¬C∨D}, let's perform the resolution steps:

1. Resolve on AA in A∨C∨DA∨C∨D and A∨¬C∨DA∨¬C∨D: R1=(C∨D)∨(¬C∨D)R1​=(C∨D)∨(¬C∨D)
2. Resolve on CC in R1R1​ and DD in R1R1​: R2=DR2​=D

Now, R2R2​ is a resolvent that does not contain αα (since BB is not present). Therefore, KB⊬αKB⊬α.

In simpler terms, we couldn't derive αα from KBKB using resolution, so the answer is KB⊬αKB⊬α.

5. Assuming that you build a neural network for Spam detection and you generate the following confusion matrix:



1) Calculate the Precision.

2) Calculate the Recall.

3) Calculate the Accuracy.

4) Is your neural network a good Spam detector? Please justify your answer.

Solution :

* True Positive (TP): 99 (Predicted Spam, and it is actually Spam)
* False Positive (FP): 1 (Predicted Spam, but it is not Spam)
* True Negative (TN): 8192 (Predicted Non-Spam, and it is actually Non-Spam)
* False Negative (FN): 520 (Predicted Non-Spam, but it is actually Spam)

Now, let's calculate the metrics:

1. **Precision:** Precision=TPTP+FP=9999+1=99100=0.99Precision=TP+FPTP​=99+199​=10099​=0.99
2. **Recall (Sensitivity):** Recall=TPTP+FN=9999+520Recall=TP+FNTP​=99+52099​
3. **Accuracy:** Accuracy=TP+TNTP+TN+FP+FN=99+819299+8192+1+520Accuracy=TP+TN+FP+FNTP+TN​=99+8192+1+52099+8192​

Now, let's evaluate the results:

* The high precision value indicates that when the model predicts Spam, it is correct 99% of the time.
* The recall value would be lower due to the relatively high number of false negatives. This means the model is not capturing all instances of Spam in the dataset.
* The accuracy would be high because it considers both true positives and true negatives, but accuracy alone may not be a sufficient metric, especially in imbalanced datasets.

Whether the neural network is a good Spam detector depends on the specific requirements and trade-offs. In spam detection, false positives (non-spam emails classified as spam) are usually less critical than false negatives (spam emails classified as non-spam). Therefore, if the goal is to minimize false negatives, the model may need improvement. However, if a high precision is crucial (minimizing false positives), the model seems to be performing well.

In summary, the model has high precision, but the trade-off is a lower recall. The evaluation of whether it's a good spam detector depends on the specific priorities and requirements of the application.