CS 3600 PROJECT 4a Analysis

Name: CHANG Yingshan
GT Account: ychang363
GT ID: 903457645

Question 6

1. Dummy 1:

5 = 0
|---1
5 = 1
|---0

The Decision Tree makes very accurate classification on the dummyDataSet1. The tree size is 3. It is apparent that all the classification results only depend on the attribute with index 5. If that attribute is false, then the result is true, and vice versa. Overall, the decision tree algorithm performs well on this dataset because the accuracy is high and the tree size is small.

2. Dummy 2:

2 = 0
|---0 = 0
|---|---0
|---0 = 1
|---|---4 = 0
|---|---|---1
|---|---4 = 1
|---|---|---0
2 = 1
|---5 = 0
|---|---6 = 0
|---|---|---0
|---|---6 = 1
|---|---|---1
|---5 = 1
|---|---1

The decision tree algorithm does not make very accurate classification on dummyDataSet2. The tree size is 11, but the classification rate is only 0.65. The tree size is much larger than the one generated on dummyDataSet1 because, unlike dummyDataSet1, the label in dummyDataSet2 depends on multiple attributes. However, the dependency is somewhat more complicated, which cannot be well-learnt by the decision tree. There are two reasons that may account for this. Firstly, the attribute with the most information gain might not be the best attribute for splitting. Secondly, the training and testing datasets are relatively small. Thus, we should expect variations between the

training and testing datasets. Therefore, the decision tree built on the training dataset might not perform well on the testing dataset.

3. Connect4:

The average classification rate is 0.7618 and the tree size is 41521. The classification rate is not high, and the tree size is huge. The possible explanations for this result are summarize below.

1) The attributes might not be highly correlated with the label. Thus, each attribute contributes very little to the data distribution and the decision tree algorithm has to choose many attributes and let the tree grow deeper and deeper. However, there might by noises near the leaf nodes because near the bottom of the tree, the number of data points inside one node is small, and noises could have big influences.
2) Since the stopping criteria are defined as "stop when a node is empty", or "stop when we run out of attributes" or "stop when all data points in a node have the same label", the decision tree algorithm is actually very greedy, which tries to classify every data point in the training set. This results in an extremely deep tree and, more importantly, may lead to overfitting. The best way to prevent such problem is using pruning methods.
3) When I look into the description for this dataset, I noticed that the training set is highly unbalanced, which means the label distribution is highly uneven. There are three labels for this dataset: {"win", "loss", "draw"}. Training data that has label "win" makes up a huge proportion: 65.83%, while those with labels "loss" and "draw" only account for 24.62% and 9.55%. Therefore, in the learning process, the algorithm will see "win" much more times than the other two labels, and this label is likely to "dominate" the classification result. To address this problem, some pre-processing techniques are needed to make the label distribution more balanced.

4. Car:

The average classification rate is 0.94575% and the tree size is 408. The decision tree algorithm achieves high classification rate with a fairly small tree, which may due to close correlation between the attributes and label, as well as fewer attributes and less noise. Besides, there are attributes contribute a lot to the classification results, which allows the algorithm to reach decision after only seeing a few attributes. For example, cars whose safety is rated as "low" is immediately classified as unacceptable and cars with "med" or "high" safety score but can only accommodate 2 persons is classified as unacceptable

Question 7

1. Car:

Decision tree can be used by websites like www.car.com to make suggestion for potential customers. The website would need to collect user preferences like price range, safety rating, # doors, etc. Then the pre-learned decision tree can be applied on the cars to decide whether a certain car is acceptable to the user. The website can recommend those cars that are predicted to be highly acceptable for the user. After the recommendation is evaluated by the user, the website knows whether the recommendation is accurate or not. And the website can incorporate this data point into its training

dataset and refine its classifier.

2. Connect4:

The decision tree can be incorporated with the search algorithm to evaluate the "goodness" of a state. States with higher goodness scores will be expanded first. Whenever the agent come into a new state, all the attributes should be embedded in the state vector, or be sensed from the environment, in order to form a data instance. Then the decision tree can be applied to score a game state. Probably, this could be effective in Minimaxing (minimizing the possible loss for a worst case) or Expectimaxing in AI.