Analysis of DT (Q 6&7)

Q 6:

Testing dummy dataset 1. Number of examples 20.

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

Tree size: 3.
Classification Rate: 1.0

The Decision Tree Performs great for the Dummy Dataset 1. Even with 20 training examples it has been phenomenal. This is because the data is linearly separable. And the 5th attribute does that. And that's why the classification rate is maximum, and the tree size is 3 (because it uses only the 5th Attribute).

Testing dummy dataset 2. Number of examples 20.

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

Tree size: 11.
Classification Rate: 0.65

The Decision Tree performs only slightly better than chance. Its classification rate is 0.65, and the tree size is 11. Here, the data is not linearly separable. And as we don't have enough training data, the Decision Tree struggles to get good accuracy.

Testing Car dataset. Number of examples 1728.

Tree size: 408.
Average classification rate over all runs: 0.942

The Decision Tree performs great on the Car Dataset. It has 6 attributes and 4 classes. Since we don't have sufficient examples for the small number of attributes, and probably because our dataset is linearly separable, it performs well. I am also of the thought that the small error rate might even be because of noise in the dataset, or it might be because our attributes don't define everything that causes the class to take on certain value.

Testing Connect4 dataset. Number of examples 67557.

Tree size: 41521.
Average classification rate over all runs: 0.75805

The Decision Tree performs okay on the Connect4 Dataset. The Tree size is HUGE (41521), because of the number of attributes on this dataset. It has a total of 42 attributes, and 3 class values. The data is possibly not linearly separable. And that may be the reason why it performs like this. That, and also even 67.5k examples might not be enough on this, because of the number of attributes.

Q 7:

For the Car dataset, we can have the manufacturers actually predict the value for each of attribute, and run it on this DT to see if it scores high. They can then make an informed decision whether to move ahead with making this car or to go back to the drawing board, and see how they can affect certain attributes, so that the class value is high.

For the Connect4 DT, we can have a GUI prompting the player with the "value" of a certain move. Given this dataset, we can figure out the percentage of games that ended up in a win, and inform the player the best move to play, or the score for the move he/she made.

A similar dataset could be a dataset where we have attributes as search keywords. For example, let's take Amazon. The attributes will be the keywords searched, and the class could be the product that the customer ended up buying. We can collect a large amount of data, and deploy a DT, which will give a set of cars that the previous users ended up buying given the current keywords. With that, Amazon can order products such that the most bought item for that keyword would appear first. This would reduce shopping time, which is definitely a win-win for both amazon and the customer.

The DT we obtained here itself can be used as an heuristic to any of the informed searches we learnt about in the past. With that, we can make the search agent explore states that might lead to a high percentage of win, and therefore would definitely reduce search time.