Question 6:

The decision tree algorithm is accurate for classifying data in the dummyDataSet1. But does not perform well for dummyDataSet2. The classification rate is 100% and 65%. I think Because all examples in DataSet1 that has class value equal to 1 has 0 for their No.5 attribute, and all examples that has class value equal to 0 has 1 for their No.5 attribute, No.5 attribute has the largest info gain, and when we use No.5 attribute to split data, we can reach the final decision tree that has two leaf nodes. However, for Dataset2, I think because of the training dataset only has input size of 20 and the attribute with the most information gain in the training set might not be the best attribute in the testing data set.

The tree generated for the Connect4 dataset has size 41521 based on 67557 examples and accuracy about 76% whereas the tree for the cars dataset has size 408 based on 1728 examples and accuracy about 95%. The Cars tree input size is much smaller than the Connect4 tree, but it's more accurate in classification rate. The plausible reason is that since the dataset of Connect4 has a lot of examples, there might be many noises near the leaf nodes, which the decision tree is trying to fit. And near to the bottom of the tree, the attributes for splitting data are not selected based on statistically supported decisions. On the contrary, for Cars dataset, there are only 6 attributes, 4 possible values for first four and 3 possible values for last three, so there are 1728 combinations. The data set also has 1728 examples, so there are no useless attributes. Because of the example is not very big, there are not many noises near the leaves, and overfitting is less likely to happen.

Question 7:

The decision tree for classifying cars can used to preowned car dealer websites to help user make decisions. The data set has searches with attributes including mileage, model, doors, mpg, price, style, etc. After a big amount of data is collected, we can use the data set to build a decision tree to give user recommendations. When a user makes a search on the website, the system will collect his preferences, and apply these attribute values to the decision tree. The decision tree will finally lead to a car model that best match for the users.

The Connect 4 game analysis could help an agent figure out heuristics for where the s the best move on the board. We can apply the minimaxing algorithms to the connect4 decision tree making. We check all neighboring four consecutive piece, +1 if the piece is ours, and -1 if it is the opponents, if it is empty we do nothing. And sum of all evaluations as the evaluation function. Each set of the dataset results in a move decision by the minimaxing algorithm, and we can use the minimaxing to decide which move to take by our decision tree.