The decision tree algorithm is very accurate for classifying data in the dummyDataSet1. Because all examples that has class value equal to 1 has 0 for their 5th attribute, and all examples that has class value equal to 0 has 1 for their 5th attribute, 5th attribute has the largest info gain, and when we use 5th attribute to split data, we can reach the final decision tree that has two leaf nodes. Therefore, the tree size is 3. Since we are testing data sets that are randomly selected from the training data, testing data examples have the same patterns for the 5th attribute as the training data, when applying the decision tree to testing data all outputs should be correct. That is why the classification rate is 1.0.

The decision tree does not perform very well for dummyDataSet2, because it has a large tree size (11) and low classification rate (0.65). The reason is that the training data set has a small size. The attribute with the most information gain in the training set might not be the best attribute in the testing data set. Also, it might need additional attributes to classify data in the testing data set, so the decision tree build from training data might not be accurate.

For connect4 dataset, there are 42 attributes and each attribute has 3 possible values, so there are 3^42 combinations, but there are only 67,557 examples and 3 classes. Therefore, each attribute contributes very little to the data distribution, which means we have to use a large amount of attributes to classify the examples. This explains why the tree size is big (41521). Also, as can be seen in the output file, the first attribute (a1) contributes nothing to the data distribution, and the second and third ones contribute very little. The classification rate for this dataset is 0.763000, which is not very high. The most plausible reason is that since the training set has a lot of examples (67,557), there might be many noises near the leaf nodes, which the decision tree is trying to fit. The problem is that near the bottom of the tree, the attributes for splitting data are not selected based on statistically supported decisions. So our decision might do poorly when it is applied to another problem. The best way to get around such problem is to construct Dtree with pruning.

For Car dataset, there are only 6 attributes, with the first three having 4 possible values and last three having 3 possible values, so there are 1728 combinations. The data set also has 1728 examples, so even one attribute can contribute a lot to classifying data into subsets. Since there are no useless attributes, the decision tree has a fairly small size (408), and as shown in the output file, the decision tree starts to split at the first attribute. The classification rate for this dataset is 0.940000. The reason for this high score is that the data set is not very big (1,728 examples). Because of that, there are not many noises near the leaves, and overfitting is less likely to happen. So when we apply this decision tree to the testing data its accuracy will not decrease very much.