Solution Notes

A possible output:

*Decision Tree with No pruning*

*The train error of the decision tree is 0.021621621621621623*

*The test error of the decision tree is 0.42*

*The amount of rules generated from the tree 92*

*Decision Tree with Chi pruning*

*The train error of the decision tree with Chi pruning is 0.2810810810810811*

*The test error of the decision tree with Chi pruning is 0.3*

*The amount of rules generated from the tree 28*

*Decision Tree with Rule pruning*

*The train error of the decision tree with Rule pruning is 0.06486486486486487*

*The test error of the decision tree with Rule pruning is 0.4*

*The amount of rules generated from the tree 85*

Some notes:

1. The prune is dealing with the overfit. You can see that the training error is higher whereas the test error is lower in prune decision trees. In addition you can also see that the chi square pruning deals with it better and reduces the number of rules much better than the rule pruning.

Remember that the rule pruning that you implemented is very simple and needs to reduce the validation error in order to prune rules. The validation set contains only 50 instances, so the theoretical maximum possible number of rules to prune is 50 (starting from 100% error down to 0% error), and of course in practice the number is much lower.

1. The chi square pruning was executed during the building of the tree.
2. If you got around 270 rules in the regular decision tree (without pruning) you probably created a child for all the attribute possible values – which is a mistake. What you should have done is to create a child only for the attribute values of the instances that reached the current node.