EECS 349 PS2 Report

By Jiaming Li, Ivy Zheng, Annie Fu

1. The decision tree is represented as a Python list, in which every node takes form of [best\_attribute, best\_split\_value, less\_than\_value\_node, more\_than\_value\_node] or a binary integer (0 or 1).

2. Examples are represented by a matrix. The matrix has as many rows as the number of attributes. Each column is an example. We chose this design because the matrix in Python is stored in row-major order. If we want to create a decision tree, we want to access the information in each attribute. Therefore it makes more sense to store data of each type of attribute in the same row.

3. For each node, we calculate the information gain (minimum entropy) for a split in each attribute. We pick the attribute with largest information gain. For each node in the decision tree, we store several information: the attribute we chose to split the node, the threshold for that attribute, the left node, right node, and parent node.

4. For the missing attributes, we replace them with the average of all other attributes of the same type. So for example we replace “?” in winning rate with 0.5 if the average winning rate is 0.5

5. We search the lowest point of entropy by calculating the entropy on both side of a point and moving to the side with lower entropy. The termination for each node split is when we reach a local minimum in entropy. The whole training process terminates when the height of the tree reaches the MAX\_HEIGHT constant defined in the script.

6. The Boolean formula of the unpruned tree of maximum height 4 (15 nodes) is the following:

(((oppnuminjured<1.11)and(((numinjured<0.72)and((oppnuminjured<0.01)and(numinjured>-0.8)))or((numinjured>0.72)and((oppwinpercent>0.41)or((oppwinpercent<0.41)and(oppwinpercent>0.26))))))or((oppnuminjured>1.11)and(((numinjured<1.08)and((opprundifferential<40.6)and(opprundifferential<24.7)))or((numinjured>1.08)and((oppnuminjured<2.02)or((oppnuminjured<2.02)(numinjured>2.97)))))))

7. The Boolean formula above, organized by the parenthesis, is pretty self-explanatory. The predicted result will be 1 if and only if the formula above returns true.

8.

9. [[[[ run printTree(tree) to get the boolean formula. ]]]]

10.

11. For the unpruned tree, the accuracies are: [[[[choose the one you like]]]]]

MAX\_HEIGHT = 3: 0.848 4: 0.879 5: 0.894 6: 0.904 7: 0.909

12. [[[run showLearningCurve(theData, theValidationData, trial\_num, will\_prune) and there will be a popup window of the curve in the end]]]]

13. [[[[[ call the outputCSVResult(test\_data, tree) when you decide on the tree. ]]]]]

14. Jerry Li: splitting; Annie Fu: pruning; Ivy Zheng: everything else.