CST8390 - Lab 4
Classification by Decision Trees

Due Date: Week 5 in corresponding lab sessions

Introduction

The goal of this lab is to perform classification on Diabetes dataset using Decision Trees.

Steps:

1. Create an answer document named Lab4_Answers.doc.

2. Open Diabetes dataset in text editor (from datasets that came with Weka). Read the information about the file. Fill in the following information (should be typed in) in the answer document.

   a. Number of instances:

   b. Number of attributes:

   c. List of attributes (NOT abbreviation, should be **typed** in):

   d. Class labels and their relabelled values:

   e. Number of instances for each class label:

3. Load the dataset in Weka. Take a screenshot and paste it that shows **class** distribution.

4. Click on the "Choose" button on "Classify" tab, and select J48 from "trees". It is the implementation of the C4.5 algorithm which uses entropy to create a decision tree.

5. For testing the classification accuracy, make sure that "(Nom) Class" is selected, and cross-validation has 15 folds (Make sure that seed = 1). Click start and you should see a textual version of the decision tree. Fill in the following information in the answer document.
   a. Copy and paste the confusion matrix:

   b. Number of leaves:

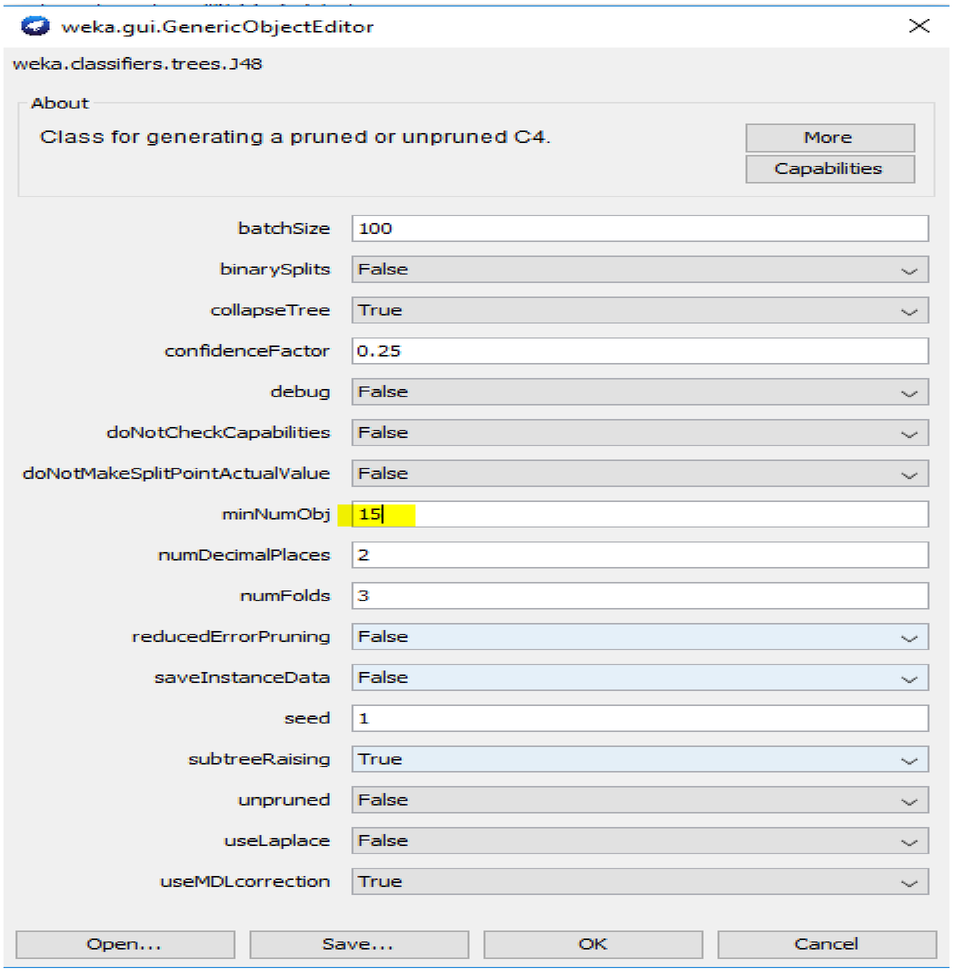
   c. Size of the tree:

   d. Correctly classified instances:

6. Right click on the result buffer and select "Visualize tree". From the new window, make it full screen and then right-click on the window and select "auto scale". It will draw the tree so that it's wide enough to read the text. You might have to right-click again on the screen and "Center on Top Node". You can use the mouse to pan around the tree to see all of the decision splits. Right-click again on the screen and select "Fit to Screen". Here you can see the tree all in one place, but the text might be hard to read. Have this window open for your lab demonstration. Also, take a screenshot and paste it in the answer document.

Even though the attributes are numeric, you can see that decision tree created categories. For example, first category may be based on ‘plas’, i.e. those instances that has plas as <=127 and those with >127.

7. Set minNumObj to 15 in the settings window of the classifier, as shown below (This means that don’t continue splitting if the nodes get very small. Default value is 2):



weka.gui.GenericObjectEditor

weka.classifiers.trees.J48

About

Class for generating a pruned or unpruned C4.

batchSize	100
binarySplits	False
collapseTree	True
confidenceFactor	0.25
debug	False
doNotCheckCapabilities	False
doNotMakeSplitPointActualValue	False
minNumObj	15
numDecimalPlaces	2
numFolds	3
reducedErrorPruning	False
saveInstanceData	False
seed	1
subtreeRaising	True
unpruned	False
useLaplace	False
useMDLcorrection	True

Run the classifier with this setting and fill in the following information in the answer document:

a. Copy and paste the confusion matrix here:

b. Number of leaves:

c. Size of the tree:

d. Correctly classified instances:

8. Take a screenshot of the tree and paste it (from “Visualize tree”) in the answer document.

9. Now, turn off pruning by setting unpruned property to True (also, se minNumObj to 10, seed = 1) in the settings window of the classifier, as shown below (this means that we are not reducing the size of the tree even if it is not giving much value for the task). Run the classifier with this setting and fill in the following information in the answer document:

    a. Copy and paste the confusion matrix here:

    b. Number of leaves:

    c. Size of the tree:

    d. Correctly classified instances:

10. Run the classifier again with unpruned property to True and minNumObj to 25, and fill in the answers for the questions below in the answer document:

    a. Copy and paste the confusion matrix here:

    b. Number of leaves:

    c. Size of the tree:

    d. Correctly classified instances:

11. Take a screenshot of the tree and paste it (from "Visualize tree") in the answer document.

12. Decision trees have a problem with overfitting. One way to correct overfitting is with using random forests. This uses many decision trees, each built with random subset of the data. When a new item is going to be classified, the trees all vote when classifying each data item, with the majority deciding the final answer. The probability of an outlier being selected to be in several of the trees is highly unlikely so they will have less impact on the final classification.

    To run the random forest algorithm, click the "Choose" button and select "Random Forest". Select Run the algorithm and paste the confusion matrix in the answer document. Also fill in the following information:

    a. Details of random forest: __________________ with _____ iterations

    b. Time taken to build model:

    c. Correctly classified instances:

In order to get credit for this lab,

1. You should be ready with all your results in the result pane.
2. Show trees for steps 5, 7 and 10.
3. Upload answer document to Brightspace (should have answers to questions 2, 5-12)

Both demo during lab hours and submission in Brightspace are required to get credits for the lab.