Classification Trees: A Problem Set

Team 8: Shea Thomas, Jake Seefeldt, Max Meneses, Patrick Poehailos

I. Coding and Analytical Exercises
   a. Preparation:
      i. Your data file is myWeatherData.csv
      ii. You will need packages rattle, rpart, and pROC
      iii. Available information is both in the in-class scripts and slides
   b. Check your Reflexes:
      i. Build and evaluate a Classification Tree to predict "Rain Tomorrow"
         1. Remove NA's
         2. Remove Date and Location (not useful to predict)
         3. Also remove RISK_MM (perfect predictor)
         4. 90/10 Training Test Split
         5. 525600 Random Seed
         6. Maximal Tree, Prune Up, and Visualize It, Print Rules
         7. Evaluate Test Set (confusion matrix)
         8. Produce ROC Curve
         9. Compare to Default and Maximal Trees
      ii. Refresher! Build another classification model of your choice to compare the model's effectiveness and comment in your code.
         1. You can use any other classification model available, and any particular metrics you'd like (power, type error rates, AUCC, etc.) to make your comparison
         2. How different is the effectiveness of the two models? What variability/bias trade-offs do they each face?
         3. Why might you choose the Classification Tree? Why might you choose your alternative.
         4. Note: When building your model, it is worth trying to reserve the given training-test set splits from above for an effective comparison on the test set. This means you will likely want to use cross-validation to do any tuning selection, but you may also choose to do standard validation.

c. Experimentation:
    i. First, build a pruned model without removing the NA's, but instead setting the usesurrogate argument to 1 and the maxsurrogate argument to 2.
        1. Remember to remove columns!
        2. Reperform the 90/10 test split with seed 5072
    ii. Compare the predictive power to your <u>pruned model with NA's removed</u> by running them against the **<u>second</u>** withheld test set. Do you notice any changes? Why or why not? Visualize the trees to compare as well. When would we want to use the surrogate argumen
    iii. Assume we know that in this area it rains 70% of days. Write the line of code required to implement this information into a otherwise default tree.
    iv. Finally, assume we want to penalize type II error 42 times as much as type I error (Not raining being the Null Hypothesis). Implement this into an otherwise default tree.

II. Check your understanding: True or False

a. Tree Terminology
    i. Decision Trees use greedy, recursive, binary splitting. T/F
    ii. The goal of a classification tree is high purity leaves. T/F
    iii. Decision nodes can be written as rules. T/F
    iv. Trees allowed to grow without restriction are called 'Full Trees'. T/F

b. Error Metrics
    i. RSS can be used when training a classification tree. T/F
    ii. Entropy and Gini-Index measure the homogeneity of a split. T/F
    iii. A Gini-Index or Entropy value of .5 is ideal. T/F
    iv. CP is the parameter used for pruning the tree. T/F

c. R Implementation
    i. method = 'binary' is the method used to build classification trees. T/F
    ii. You can specify split = 'gini' to perform your splits based on the GINI index. T/F
    iii. priors = c(50,50) assumes an even split of responses in the population. T/F

iv. A loss matrix derived from: matrix(c(0,1,50,0), byrow = T, nrow = 2) strongly discourages type 1 error, shifting responses to the null hypothesis. T/F