131 Project - Task 1: Classification Trees

Objective: predict the winner in each county

Tidy: Prepare dataset, grouping by county, and dropping the third-party candidates, taking the winner in each county and doing a classification tree on demographic conditions.

Workflow

- Partition data 80:20
- Grow small tree with large lower bound on observations at each terminal node.
    - Note assumptions: nmin = 60, mindev= exp(-6), split = 'deviance'
- Grow huge tree with very small lower bound on observations at each terminal node.
    - Select K-folds (k=8) to do cross-validation on huge tree.
        - I think this iteratively reduces the number of terminal nodes. At each step, it terminates the node that increases total impurity value the least.
        - Stores the size (# of terminal nodes), dev (impurity value), and k (alpha, the tuning parameter controlling complexity penalty [high alpha -> low complexity])
    - Slice the alpha corresponding to the smallest impurity and, secondarily, smallest size
    - Prune tree with prune.tree(tree, k = alpha)
- Compare misclassification rates and RMSE between huge, small, and pruned trees

Additional thoughts:

1. Consider adjusting the probability threshold of class labeling based on min(Youden)
   a) Classification error columns problem
2. Consider using Gini instead of deviance for loss function
3. *Tune hyper-parameters*
   a) Grid search
4. *Consider weighting samples so partition is representative of population*
5. Consider if the pattern of separation in the feature space is hard to approximate w/ rectangles
   a) Can we visualize this? Check lecture code
6. Consider displaying mis-classifications on the feature space of 'transit', 'white' and 'total'

Method extensions:

- Bagging (not appropriate since counties aren't iid [unless you look up how to do so])
- Random forests
- Boosting