GSE 524: Colley Iterative Method

For this assignment, you will implement the Colley iterative method to rank college football team. This should generate very similar results to the matrix method from the previous assignment. You can use the same data to test your function as you did for the previous assignment.

What to Turn In

You must follow the directions here EXACTLY. Please read carefully.

Your solution should be in a single file called colley_iter.py. You can use pandas and numpy, but no other modules.

In that file, you should create a function called colley_iter which takes two arguments, (1) a data frame, that has the same format as in the previous assignment, and (2), a float number that is used to determine when the iterative algorithm completes (details below).

The function should compute the Colley rankings and return a list with 2 elements.

The first element of the return list should have the same format as in the previous assignment: a data frame which has two columns named "team" and "score". The first column should be the team name and the second column should be the Colley score. This data frame should be sorted so that the best team (highest score) is in row 0 and the worst team (lowest score) in the last row. The sort_values function for data frames will help with this.

The second element of the return list should be the number of iterations, i.e. the number of times the body of the loop is executed. (see details below)

You can test your function using the data frame in ncaa.pkl but you do NOT need to turn in the pkl file and your function should NOT read that file, it should use the data frame passed in as an argument. Your function should work on any data passed in the correct format, not just the 2010 data.

The Iterative Method

The details on how the iterative method works are provided in the Colley documentation sections 4 and 5. The idea of the iterative method is that you start with initial scores for each team and then, each iteration of your loop, you modify the scores for each team following a formula (provided below). The method "converges", meaning that over time the changes in scores for each team from one iteration to the next get smaller and smaller. The method is complete when the team with the largest change in score from one iteration to the next is below some threshold. That threshold is the second argument of the function. So, each iteration, you should check how big the (absolute value of the) change in score was for each team, and if the largest change was below the threshold, then the loop should end. For example, if the second argument to the function was 0.001, then if no teams' scores have a change compared to the previous iteration bigger than 0.001, then stop iterating.

In the algorithm below, r is the team's score which has same meaning as from the last assignment (higher r better). Both r and effwins should be represented as numpy arrays with length=#teams, so r[0] is the score for team 0 and effwins[0] is effective wins for team 0, etc. Each time through the loop, you

first calculate effective wins (for all teams), which depends on r. Then, you calculate r (for all teams), which depends on the effective wins.

The basic algorithm (where t is the second argument):

Set initial score (r) for each team using Equation 1: r = (1+#wins)/(2+#games)

Repeat until the largest change in score (in abs value) is less than t:

- Set effective wins for each team using Equation 7
  effwins = (#wins - #losses)/2 + sum of opponents r's
  (Note: #wins is always actual wins, not effective wins! So only the "sum or r" term changes each iteration)
- Set new scores for each team using Equation 1 with effective wins in the numerator instead of actual wins (and with denominator the same as before)
  r = (1+effwins)/(2+#games)

End loop

Create and return list