CSc 496, Homework #2: Developing a win probability model

Due date: Tuesday, September 17th, 2024. **No late assignments will be accepted.**

Develop a win probability model for MLB using the 2023 season as data. You should carry out this assignment in three steps.

1. Download and process all 2340 MLB games from 2023. Make sure to put a delay (e.g., five seconds) in after downloading each play-by-play file, or Baseball Reference may stop your script. An example of where to get the data (for the Reds) is: `https://www.baseball-reference.com/teams/CIN/2023-schedule-scores.shtml`. A line in your file should look like this:

   ```
   b5,2-4,2,1-3,TF1>X,RR,CIN,Jason Vosler,Mitch Keller,-25%,45%,Triple to RF
   ```

   (The full play description, which has detail on where the triple was hit and who scored, is omitted.) This means: bottom of the fifth, score is 2-4 (score is always relative to the team at bat), 2 outs, runners on first and third, the pitch-by-pitch code is TF1 X, two runs scored on the play that was a triple (see the last field), CIN is at bat, the batter was Vosler and pitcher was Keller. (The percentages are change in win probability and current win probability for the team that ended up winning.) Note that after this play, the score is tied 4-4. You can get an explanation of each of these columns at the top of the columns on the web page. You should use CSV format.

2. Create the data file to be used in developing the win probability model. This file should have six entries per play, per game: total outs (in the game, so for a regulation game this varies between 0 and 53), run differential, whether first is occupied, whether second is occupied, whether third is occupied, and who the winner of the game is (in terms of who is currently at bat). This is a straightforward transformation of the lines described above. For example, given the play above, the line before would be 29, -2, 1, 0, 1, 0; which means there are 29 outs in the game so far, the team at bat is down by two runs, there are runners on first and third, and the home team ended up losing this game. After the play, the next line would be 29, 0, 0, 0, 1, 0; which has the same number of outs (because the result of the play was a triple), but now the home team has tied the game (so the run differential is zero) and has a runner on third. The last entry is zero because the home team lost the game, just like the previous line. Note that this means you have to make sure that you determine, for each game, who won. There are multiple ways this can be one. Here is one: look at the last line of the data file from step 1 above, and the home team is the winner if either (a) there is no bottom of the ninth, or (b) the home team batted last *and* the last line shows an "R" in the play outcome field.

3. Use a logistic regression to create a win probability model. Please print the logistic regression coefficients. Try a few game states—you should see that your model is relatively independent of the total number of outs, which is inaccurate. (For example, compare the win probability for a team leading by one run in the first inning and in the ninth inning.) For reference, here is a win probability calculator from FanGraphs: `https://www.fangraphs.com/tools/wpa-inquirer`. Compare your results to this calculator. generated as well as a few examples to show that your regression coefficients make sense This is one of the problems that can arise when using logistic regression for win probability models.

4. Improve your win probability model by splitting the data by inning. For example, try using data from innings 1-7 and generate a logistic regression and then generate a separate logistic regression using innings 8-9. Note that this means that game states from innings 1-7 must use the first regression and those from innings 8-9 must use the second regression. How much does this improve your model?

You should submit your assignment on lectura using the `turnin` command; for this program, use the assignment name `csc496-f24-hw2`. Call the files `hw2-download.py`, `hw2.csv` and `hw2-createModel.py`.