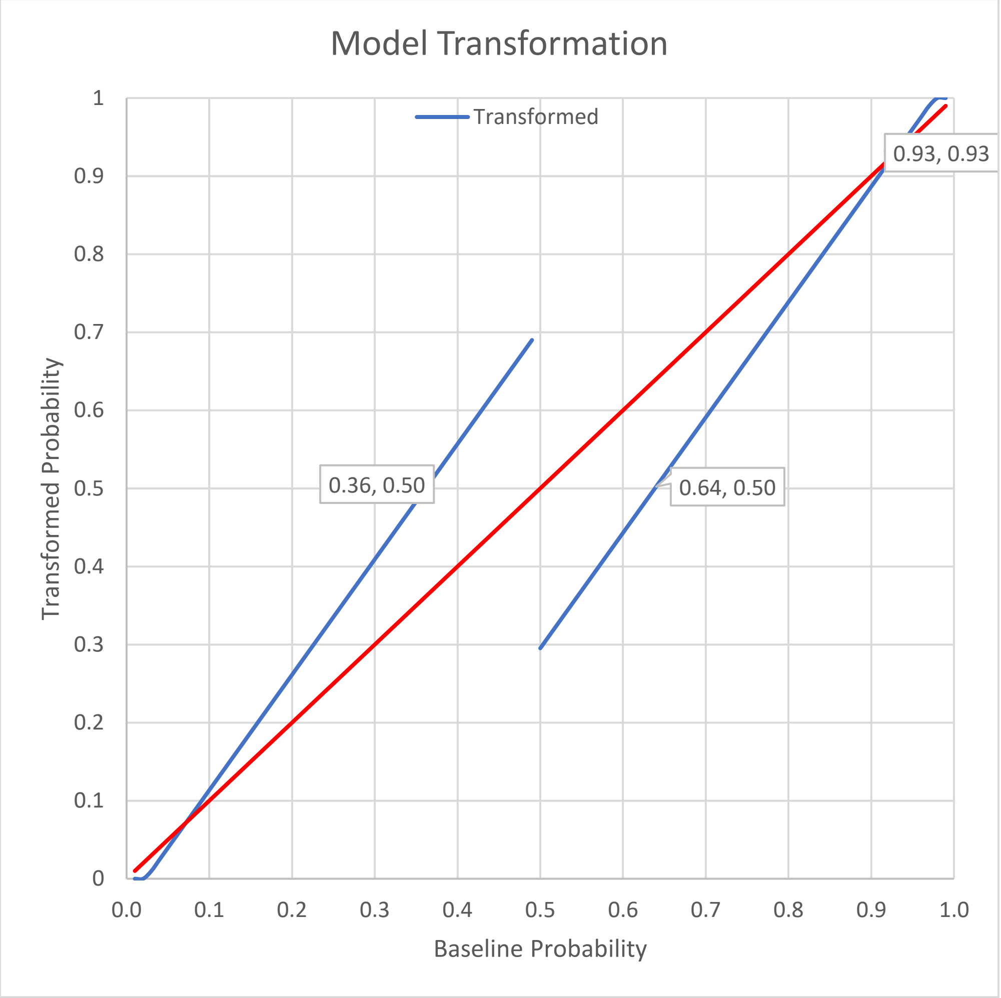
Model

My model started with a baseline that used inferred win probabilities from 538’s march madness projection. It then generated 1200 different linear transformation of those win probabilities, skewed toward small changes to mimic models clustering around 538’s baseline probabilities. It then simulated 25000 different tournaments, and selected the transformation that most often placed in the top 5 of all the transformations.

The top transformation $f(p)$ for probability p was:

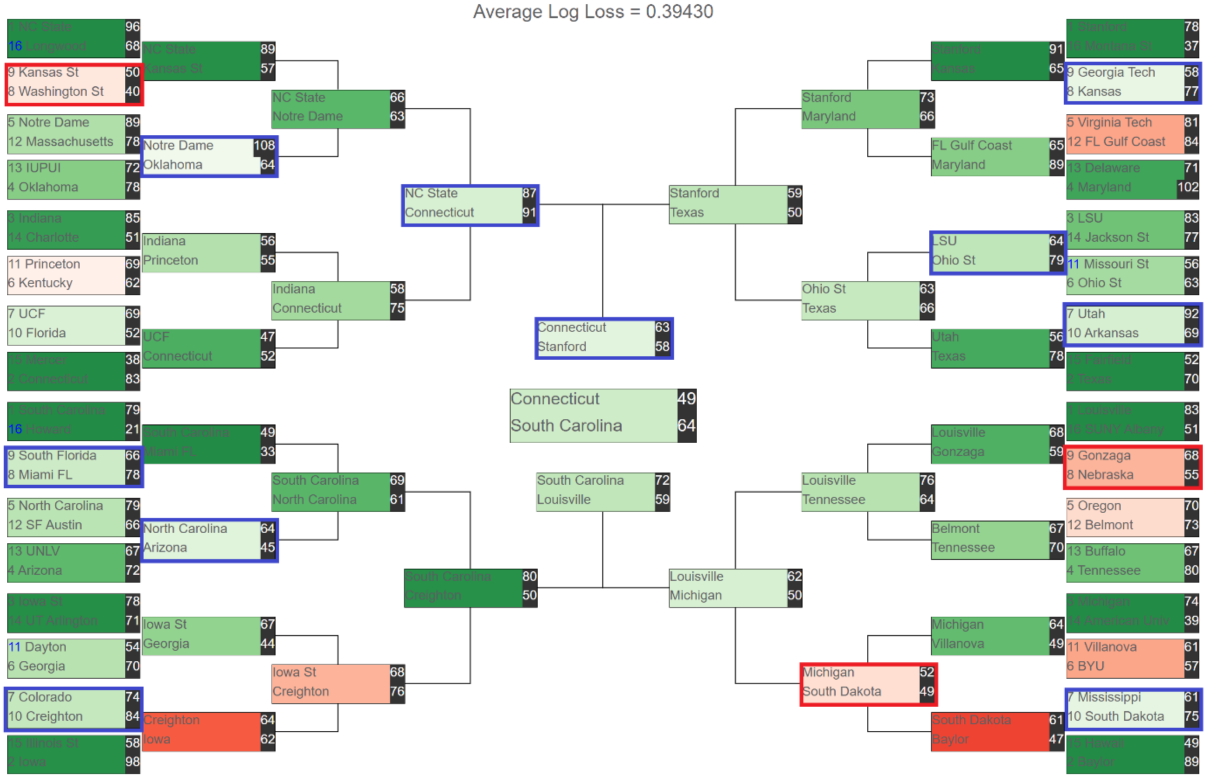
$$f(p) = \begin{cases} p - (0.2391 * |2p - 1| - 0.2048), & p < .5 \\ p + (0.2391 * |2p - 1| - 0.2048), & p \geq .5 \end{cases}$$

This transformation exaggerates win probabilities above .93, and below .93 it discounts the probabilities to the extent that it becomes contrarian between .36 and .64.



The model ended up being 10-3 in games where it was contrarian to the baseline and had an overall log loss improvement of .036.

Average Log Loss = 0.39430



Notes

- The 538 implied probabilities were calculated by using the probability a team would make it past a round, adjusted by the probability they would make it to that round and comparing that to the same calculation for their opponent.
- The transformations were made up of transforms that were solely translational, transforms solely rotational, and transforms that were a combination of the two. The skewing was done by starting at the maximal transformation and multiplying it by a factor less than 1, so that neighboring transformation continually get closer together as the transformation neared the baseline.
- The simulations used the default baseline probabilities to calculate the winners of each tournament game.
- This model is the one that I've been using for years, and I haven't adjusted it yet to output two orthogonal transformations. Both submissions were from near identical runs of the model and produced similar results: 0.39429 and 0.39662. The identical model produced a 0.74286 for the men's tournament.