**Project Introduction**

Almost every February/March since my junior year of high school, I’ve taken on the challenge of trying to forecast the NCAA tournament. For quite a few years in a row, this ritualistic undertaking served as an excellent measuring stick of sorts. In high school, my big breakthroughs were learning about correlation coefficients, so there was plenty of progress to be made.

In 2019, however, I hit a wall. Armed with some elementary knowledge of statistical learning practices, I tried more or less every model I could find in [ISLR](https://www.statlearning.com/). Needless to say, I was quite disappointed when I found that the best model I could conjure up was a linear regression with just a few variables that re-packaged other rankings data.

The message was clear – experts like [KenPom](https://kenpom.com/), who rate college basketball teams for a living set a high bar – certainly nothing I could overcome with my shoddy understanding of machine learning and rudimentary programming skills.

Last winter, though, I began to embrace a new idea: perhaps forecasting the probabilities associated with each potential outcome isn’t the most important aspect of March Madness. Even perfect knowledge of the probabilities associated with each potential matchup could not, by itself, tell anyone how to fill out a “best” bracket.

The purpose of this project is to help basketball fans ‘test’ their brackets by simulating the NCAA tournament (using pre-determined prediction inputs) and comparing their results against simulated competitors’ data. This should help fans determine what brackets might score well after accounting for:

1. What the general public has chosen, and
2. What models think the “true” probabilities are for all possible games in the tournament.

Put another way, this is not so much an exercise in forecasting or estimation, but an exercise in arbitrage. If we trust some expert-generated model as our source of truth, perhaps we can compare that to the typical selections of the general public and take advantage of any discrepancies.

**Class Model**

**“True” Model** – for each possible game in the tournament, have a probability associated with team A beating team B. This is the “true” model that is used to simulate the tournament and will be based off of some widely-accepted input like FiveThirtyEight’s projections, or perhaps by a model I create myself

- KNOW: 67 x 68 “true” game probabilities

- DO: Simulation of NCAA tournament

**People Simulator** – this takes as input, the ESPN thing that lists how many people pick each team to move to the next round. The idea here is to back out an underlying model/distribution that represents how the “people” are picking their brackets. Once the distribution has been determined, this can simulate people’s brackets

- KNOW: Percentage of teams picked to move to next round by ESPN

- DO: Simulation of a random ESPN bracket entry

-Perhaps:

-Extend further with specific fans of different teams’ inputs

-Would require scraping individual entries from ESPN

-As well as some mechanism to represent fans’ biases by fan base in easy-to-adapt manner

**My Bracket** – this will make it easy to upload your own bracket of decisions from, say, ESPN

- KNOW: Some web page or other input with bracket input

- DO: Produce a data structure representing the user’s NCAA tournament bracket selection

**Bracket Scorer**- this will actually score filled-out bracket predictions (be they simulated, or input by the user) and score them against the “True” model simulations that will be run

-KNOW: Underlying “true” model, some filled-out bracket

-DO: Score the filled-out bracket against the occurrence represented by a simulated instance of the “true” model

Use this tag to fill out my bracket

Graphical user interface, text, application

Description automatically generated