



Searching for Cinderella

— Applying Classification Models to —
Predict March Madness 2020

Kaggle Submission by:
Bert Tong

Welcome to my Flatiron End of Module 5 project where we were to use classification and machine learning techniques to solve a problem of our own choosing.

I chose to use this opportunity to submit entries into the Annual Kaggle March Madness Competition where we are trying to use whatever data we have at hand to predict with confidence, the winner of college basketball matchups for the NCAA tournament.

0.5465

Log Loss

$$\text{LogLoss} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)],$$

<https://www.kaggle.com/c/mens-machine-learning-competition-2018/leaderboard>

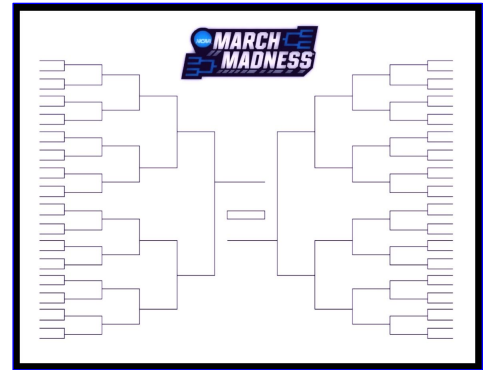
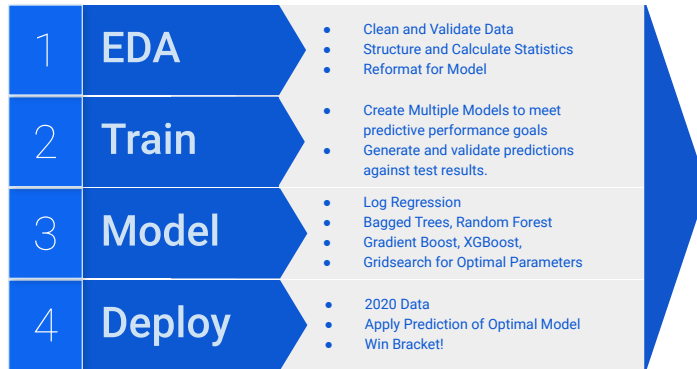
The Kaggle Prediction competitions usually use a logloss scoring system. This ends up really punishing high confidence predictions that turn out to be untrue.

The key to doing well in this competition is to be confident when you are right and not confident when you are wrong.

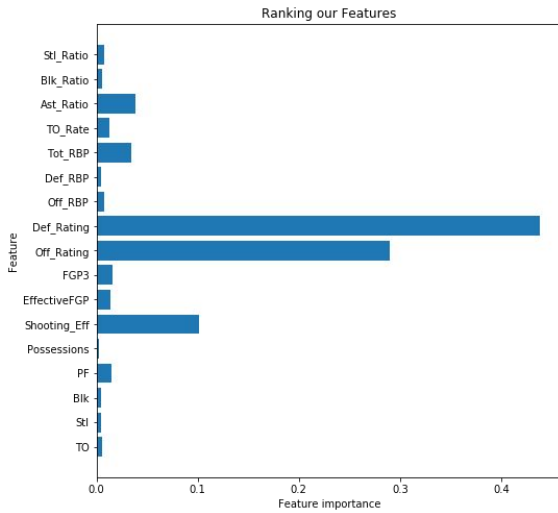
Logloss score for adjusted baseline model with seed differential in 2018.
Would have placed 4th (of 934 entries), 1 spot out of the money

Let's get into how we ended up making these predictions

Process



Apply Classification To Tournament



UNC (1)

Duke (16)



UNC

UNC will beat Duke with probability of .995

Results

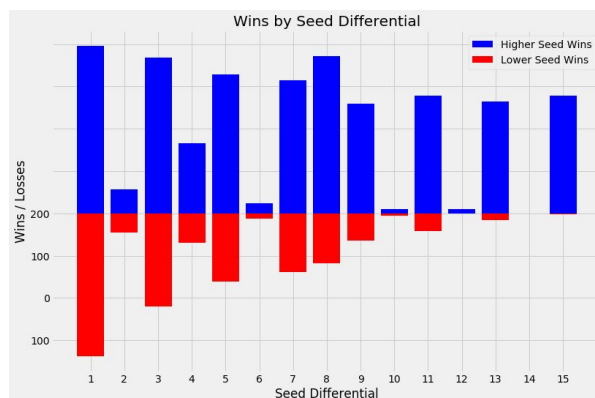
- 72% prediction accuracy (Tournaments)
- 21% logloss improvement
- ~105 minutes: gridsearch + random forest to find optimal model

Current best consistency model is ~72% accurate for tournament games from 2000-present

Log loss improvement off a baseline of using .5 for all games. Which is a .69 logloss (.15 / .69)

Opportunities for Improvement

- Shift Model POV - How accurately can identify the upset?
- Hardcode the Model (when appropriate)
- Above Replacement - type metrics
- Cap Log Loss at Extremes
- More time
- No viral or bacterial epidemics



Most Models for this effort seems to start from a bottoms up approach. Use all the data we have available, build a game prediction, and attach an outcome. But seeding generally does quite well in predicting winners (they win 72%+ of the time). It's worth seeing if we can predict an upset instead and use a straight forward model for the most part.

Hardcoding Model - 1 16/1 upset, very few 15/2 upsets. Upsets generally impact 4-6 seeds in the first round. Very rarely do they impact the 2nd weekend (Sweet 16)

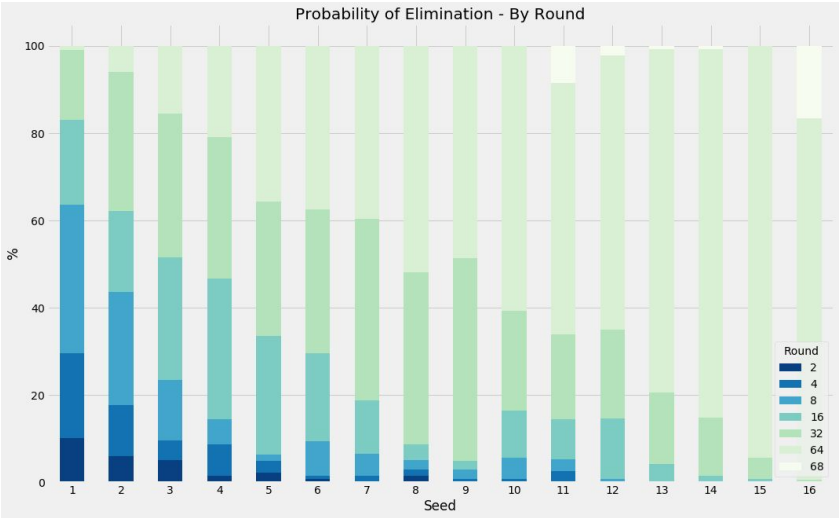
Game the Log Loss - It harshly punishes incorrect predictions with higher confidence. Capping at 0.05 and .95 maximizes logloss scores without eating too much risk.

Above Replacement-type metrics. Carolina averages 71 possessions a game, average team averages 52. Etc

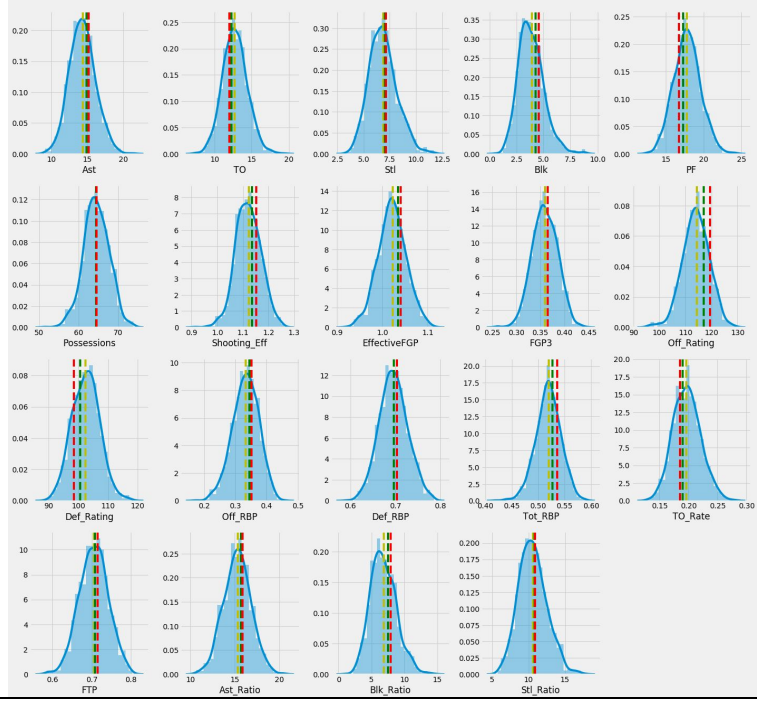
Appendix

A horizontal bar chart is positioned below the title. It consists of a single bar that is divided into two equal horizontal sections. The top section is white, and the bottom section is a solid gray color. The bar is centered horizontally within the page.

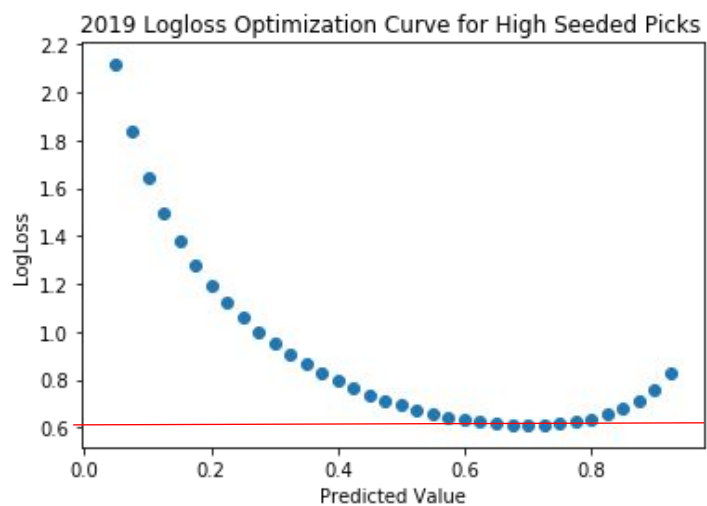
Cool Charts



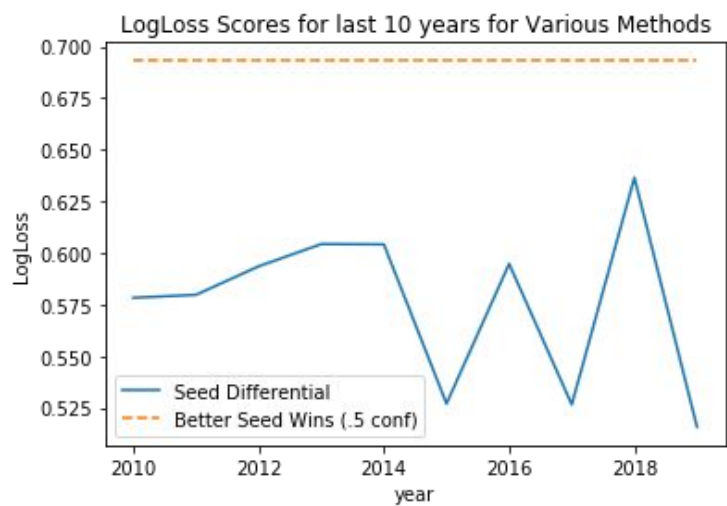
All Team Statistical Distribution with Tourney / Sweet 16 / Championship Markers








Our Baseline Competition



.5985



Kaggle Leaderboard - 2018

| # | Δpub | Team Name | Notebook | Team Members | Score ? | Entries | Last |
|---|------|---------------|----------|---|---------|---------|------|
| 1 | — | mtodisco10 | |  | 0.53194 | 2 | 2y |
| 2 | — | universe321 | |  | 0.53693 | 2 | 2y |
| 3 | — | House of Card | |  | 0.54013 | 2 | 2y |
| 4 | — | JosephDay | |  | 0.54967 | 2 | 2y |
| 5 | — | Aravindh | |  | 0.54987 | 2 | 2y |