Modeling the Madness: A Machine Learning Approach to Predicting the NCAA Basketball Tournament

Andrew DiLernia and Evan Olawsky

Division of Biostatistics
University of Minnesota

November 29, 2020

The NCAA Tournament

- The National Collegiate Athletic Association (NCAA) has an annual basketball tournament for Division I basketball teams.
- Predicting outcomes of the tournament games is of great interest to gamblers and sports fans.
- Kaggle holds an annual prediction competition for the tournament with $25,000 in prize money.

University of Minnesota

Kaggle Competition

- Kaggle provided a number of large datasets.
- Box score data for 82,041 college basketball games dating back to the 2002-2003 season.
- 3.5 million team rankings (compiled either weekly or daily) dating back to the 2002-2003 season.
- Must assign a win probability for each possible tournament game before the tournament begins.

Data: Response Variables

Randomly select one of the two teams in a given game as the reference team. Then we consider two different outcomes:

1. A binary outcome for the reference team's result,

$$Y_{ij} \sim \mathit{Bernoulli}(\pi_{ij})$$

   where $\pi_{ij} = \Pr(\text{Team } i \text{ wins against Team } j)$.

2. A continuous response given by the margin of victory (MOV),

$$Y_{ij} \sim \mathit{Normal}(\mu_{ij}, \sigma^2)$$

   In this case, we use a simple logistic model to convert estimated margin of victory into an estimate for π_{ij}.

Data: Predictors

For each team, on offense and defense, season-to-date		
Assist Ratio	Points/Possession	Rebounding Rate
eFG%	FT Rate	Tempo
Points Scored	Turnover Ratio	True Shooting %

For each team		
Location	Median ranking	Net rating

Models Considered

Ten different models were fit using both response variables:

- Full tree
- Pruned tree
- Random forest
- Lasso regression
- Principal components regression
- Linear discriminant analysis
- Gradient boosting
- Neural network
- Neural network with feature extraction
- Support vector machine

Evaluation

- We used 80% of the data (roughly 65,000 games) to train the models, and the other 20% as test data for model selection.
- The main criteria for judging model performance is the log loss function used in scoring the Kaggle competition:

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i))$$

  where $\hat{y}_i$ is the predicted probability of team A winning game i, y_i is an indicator of team A winning game i, and N is the total number of games.
- Penalizes harshly for being overly confident and wrong.

Model Selection

Model	Response	Misclass. Rate	Log Loss
LDA	Win/Loss	0.2737	0.5330
Gradient boosting	Win/Loss	0.2775	0.5332
Neural network	Win/Loss	0.2743	0.5340
Gradient boosting	MOV	0.2755	0.5354
PCR	MOV	0.2745	0.5363
SVM	MOV	0.2760	0.5364
⋮	⋮	⋮	⋮

We selected the linear discriminant analysis (LDA) model, as it performed best on the test set and is a relatively simple model.

Model Performance

Performance of binary Win/Loss models for 2019 Tournament:

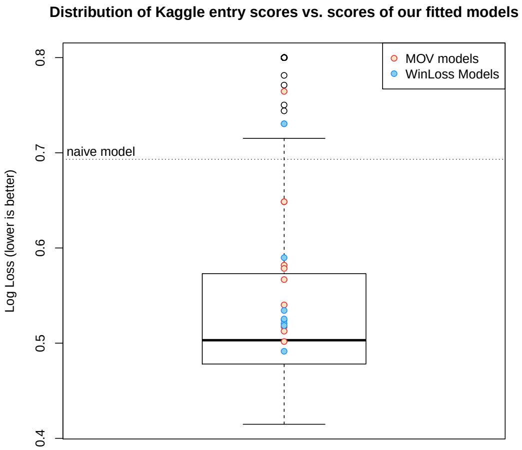
	Model	Misclassification Rate	Log Loss
1	Neural network	0.3175	0.4915
2	LDA	0.3175	0.5185
3	NNetF	0.3016	0.5207
4	Gradient boosting	0.2857	0.5242
5	PCR	0.3175	0.5254
6	CV glmnet	0.2857	0.5343
7	Random Forest	0.3175	0.5898
8	Full Tree	0.3492	0.7305
9	Pruned Tree	0.3492	0.7305

Model Performance

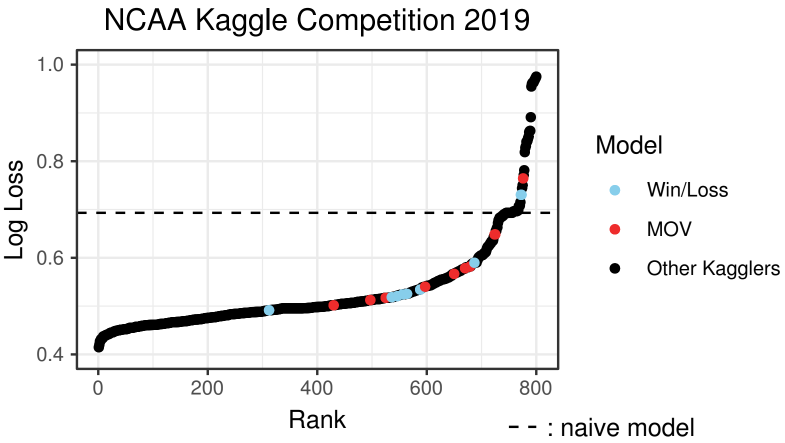
Performance of MOV models for 2019 Tournament:

	Model	Misclassification Rate	Log Loss
1	Neural network	0.2540	0.5017
2	PCR	0.3016	0.5126
3	Gradient boosting	0.2698	0.5172
4	Random Forest	0.3333	0.5402
5	SVM Radial	0.3175	0.5668
6	CV glmnet	0.3492	0.5785
7	NNetF	0.3333	0.5819
8	Full Tree	0.3492	0.6485
9	Pruned Tree	0.3492	0.6486

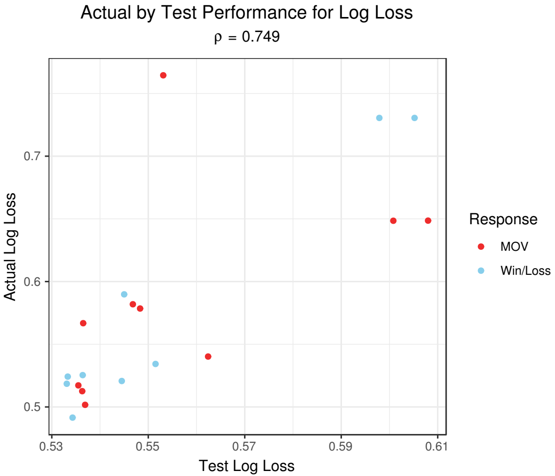
Model Performance



Model Performance



Model Performance



Discussion

- Our chosen LDA model performed reasonably well, finishing in roughly the 50th percentile of the Kaggle competition (485/866).
- The neural network for modeling win/loss performed much better in the tournament than for our test data.
- This year's tournament featured very few upsets, so entries that were very confident were likely to outperform our relatively conservative entry.

Future Directions

- Consider removing early-season games from the training data.
- Give greater weight to a team's more recent games.
- For MOV models, fit a more complicated model to convert to win probabilities
- Use feature selection to ease computation of more complex methods, such as SVM and neural network with multiple hidden layers.
- Fit models that account for the dependence of games i.e. mixed effects models.

Thank you.