HoopsML

Introducing Machine Learning to the Madness

Bisher Anadani, Andre Kurait, Surabhi Khachar, Will Duncan, Vivek Tallavajhala

Introduction

...

Project Outline

Problem statement:

- The odds of picking a perfect NCAA tournament bracket are a staggering 1 in 9,223,372,036,854,775,808 (that's 9.2 quintillion)
- That number treats all 63 games in the 2019 NCAA Division I Men's Basketball Championship bracket with 50/50 odds.
- No one has ever predicted a perfect bracket since the NCAA started tracking them.
- 7,928 out of 17.2 million Tournament Challenge entries correctly predicted Virginia, Michigan St., Texas Tech and Auburn would comprise the 2019 Final Four. That's 0.02%.

Project Outline

Strategies:

- March Madness is famous for just that – madness – so it's tempting to make bold upset picks in your bracket.
- Statistically speaking, picking No. 1 seeds to go on to the Final Four is a more successful strategy than hoping for lower seeds to upset

SEED	PERCENTAGE PUBLIC PICKS IT (SINCE 2011)	PERCENTAGE IT HAPPENS (SINCE 1985)
1	56.02	41.1
2	21.21	20.2
3	6.03	10.5
4	9.74	10.5
5	1.49	4.8
6	1.25	2.4
7	0.92	1.6
8	0.75	4.0
9	0.65	0.8
10	0.39	0.8
11	0.47	2.9
12	0.23	0.0
13	0.15	0.0
14	0.15	0.0
15	0.11	0.0
16	0.41	0.0

Project Outline

Strategies:

- The data shows that the 3rd and 4th seeds especially are underdogs
- Stats behind upsets:

UPSET	% PICKED TO WIN	ACTUAL WINNING %	DIFFERENCE	PERCENT ERROR
9 over 8	50.09%	50.00%	-0.09%	0.00
10 over 7	44.68%	38.24%	-6.45%	-0.17
11 over 6	30.88%	37.50%	6.62%	0.18
12 over 5	20.99%	34.56%	13.56%	0.39
13 over 4	11.00%	20.59%	9.59%	0.47
14 over 3	7.71%	15.44%	7.74%	0.50
15 over 2	3.32%	5.88%	2.57%	0.44
16 over 1	2.07%	0.74%	-1.33%	-1.82

Project Outline

Solution:

- Use machine learning to predict NCAA March Madness bracket
- Compile data from games in every conference
- ~70 features for every game
- Run models: Neural Network, Markov Model, Forest Regressor
- Predict bracket
- PROFIT ($1 Million A Year For Life From Warren Buffett)

Data and Models

...

Data

Team Statistics from regular season and NCAA tournament (1985-2018)

Conference Affiliations (1985-2018)

Length of NCAA Affiliation

Seed data by season (1985-2018)

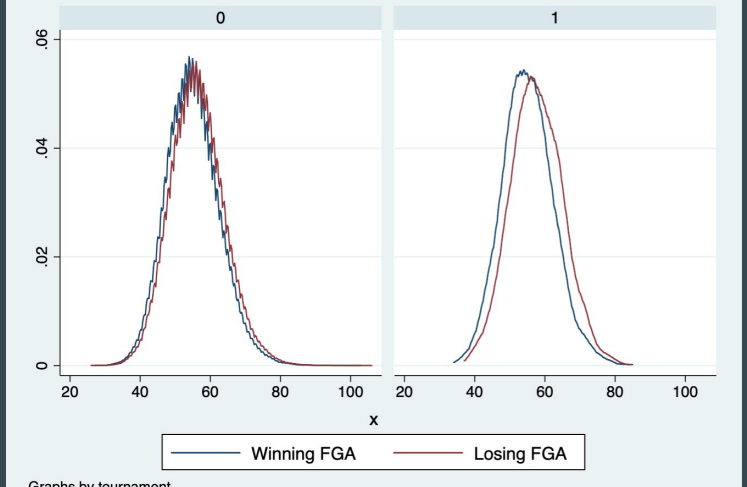
Each observation is uniquely identified by teamid, season, and game

WINNING TEAM
SUMMARY STATISTICS

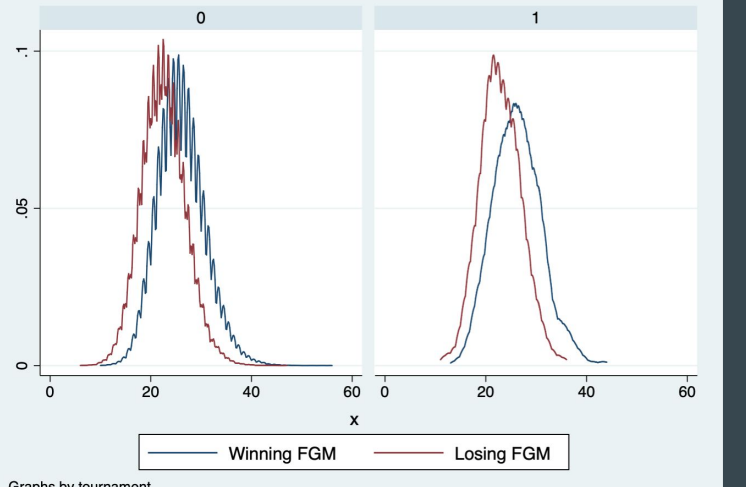
	mean	sd	min	max
wscore	75.26	11.14	34.00	144.00
numot	0.07	0.31	0.00	6.00
wfgm	26.04	4.70	10.00	56.00
wfga	55.04	7.58	27.00	103.00
wfgm3	7.03	3.04	0.00	26.00
wfga3	18.33	5.76	1.00	56.00
wftm	16.16	6.25	0.00	48.00
wfta	22.76	8.13	0.00	67.00
wor	10.97	4.14	0.00	38.00
wdr	25.59	4.90	5.00	53.00
wast	14.73	4.41	1.00	40.00
wto	13.00	4.04	1.00	33.00
wstl	7.02	3.14	0.00	26.00
wblk	3.84	2.47	0.00	21.00
wpf	17.47	4.14	3.00	41.00

LOSING TEAM
SUMMARY STATISTICS

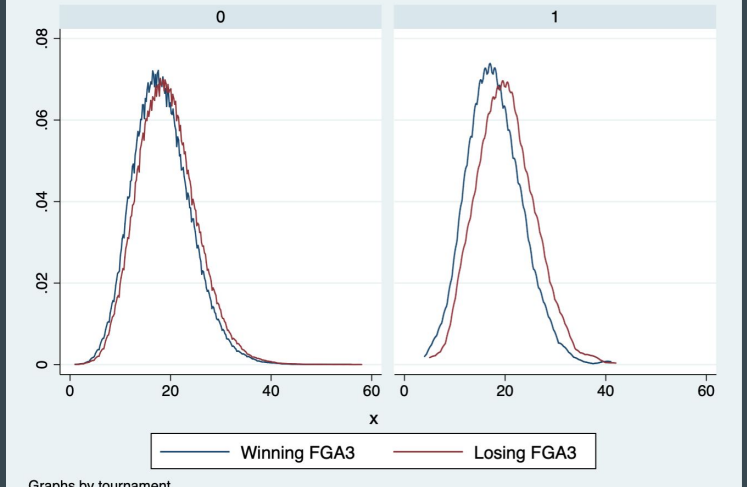
	mean	sd	min	max
lscore	63.29	10.97	20.00	140.00
numot	0.07	0.31	0.00	6.00
lfgm	22.54	4.38	6.00	47.00
lfga	56.30	7.70	26.00	106.00
lfgm3	6.00	2.76	0.00	22.00
lfga3	19.35	5.90	1.00	58.00
lftm	12.20	5.37	0.00	42.00
lfta	18.08	7.15	0.00	61.00
lor	11.19	4.21	0.00	36.00
ldr	21.53	4.53	4.00	49.00
last	11.43	3.73	0.00	31.00
lto	14.33	4.42	0.00	41.00
lstl	6.02	2.77	0.00	22.00
lblk	2.87	2.04	0.00	18.00
lpf	19.82	4.52	4.00	45.00



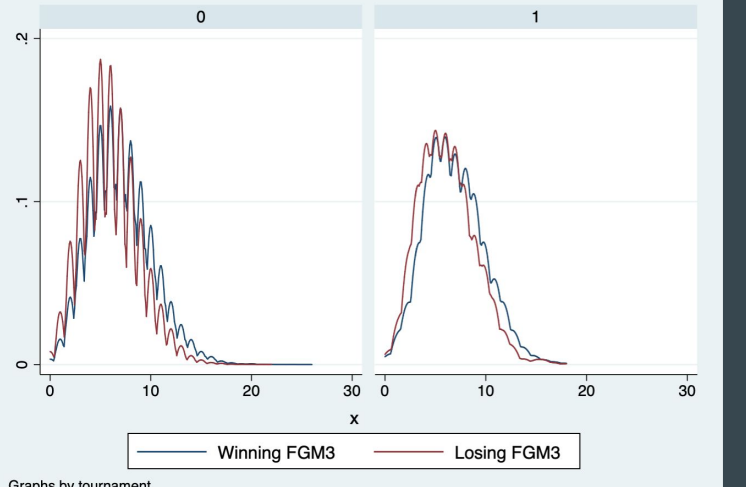
Graphs by tournament



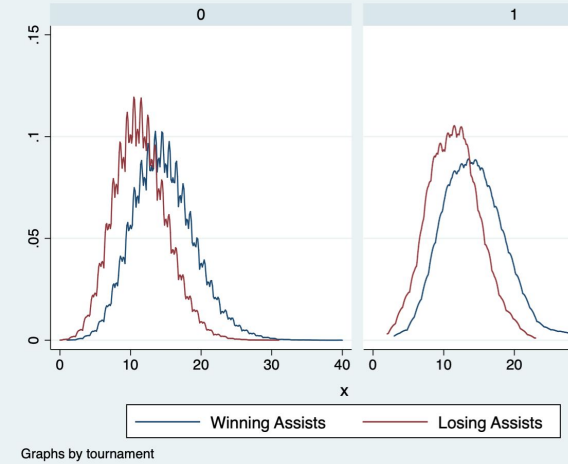
Graphs by tournament



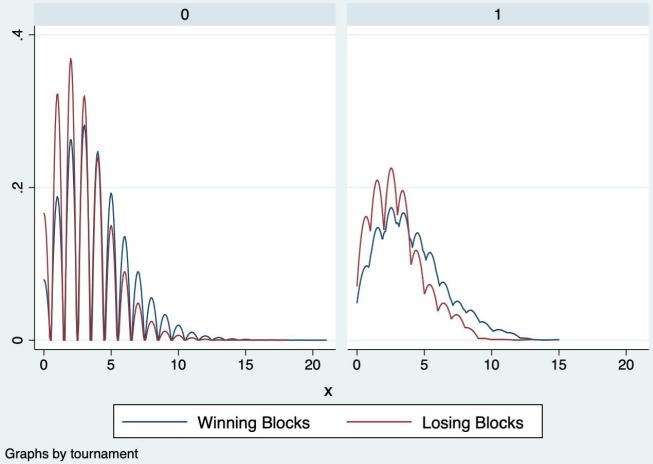
Graphs by tournament



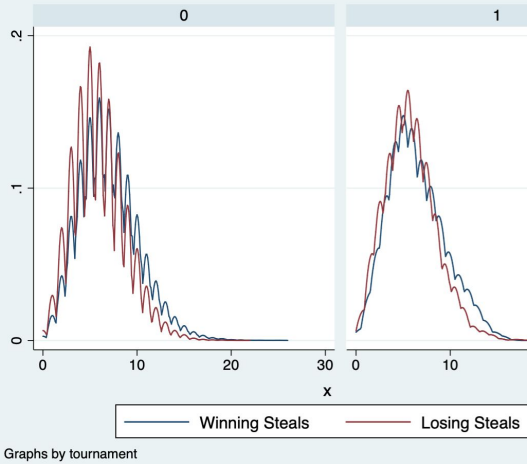
Graphs by tournament



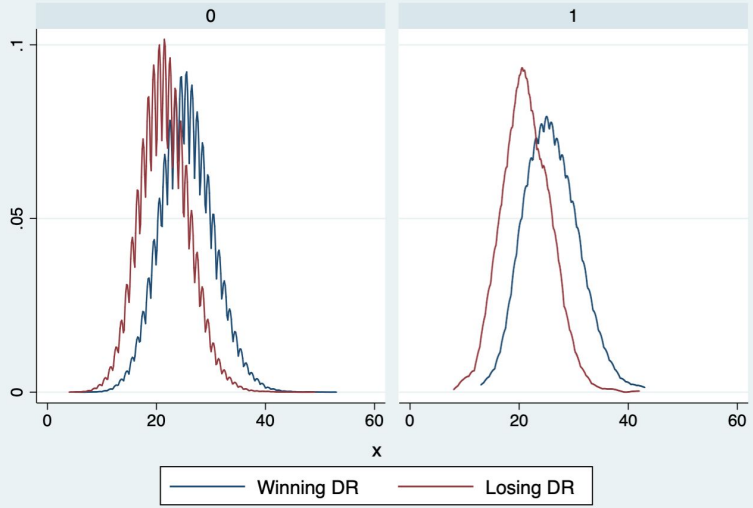
Graphs by tournament



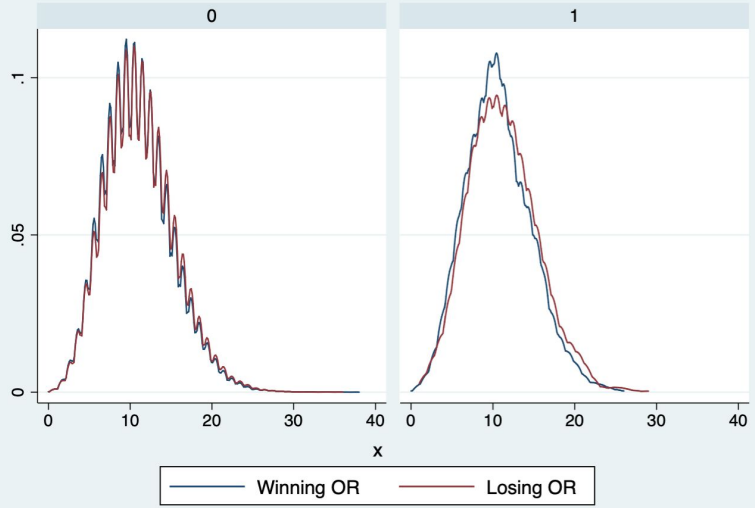
Graphs by tournament



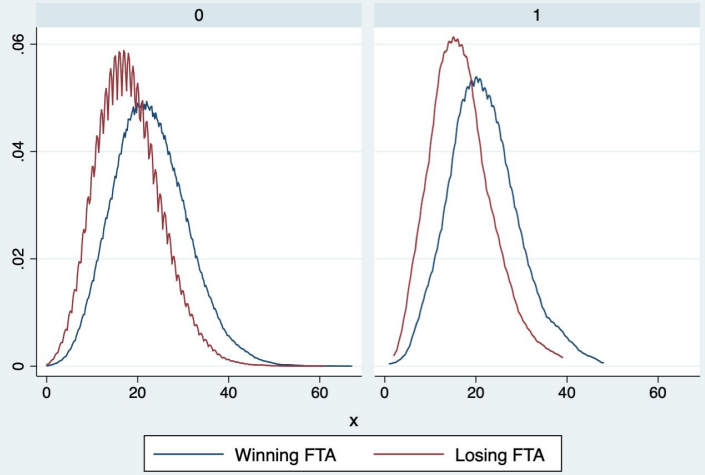
Graphs by tournament



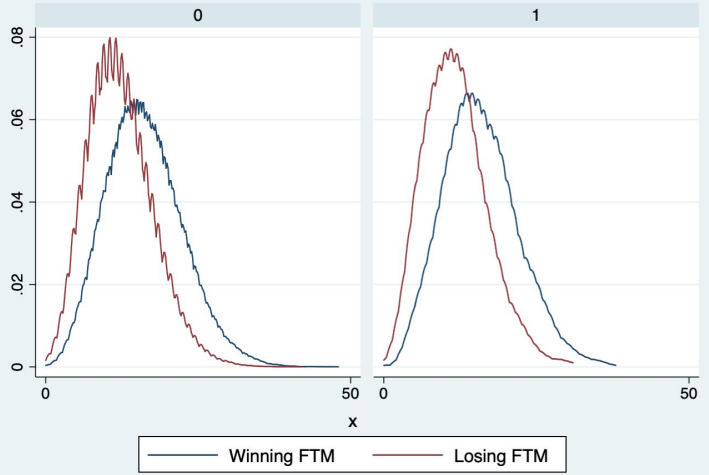
Graphs by tournament



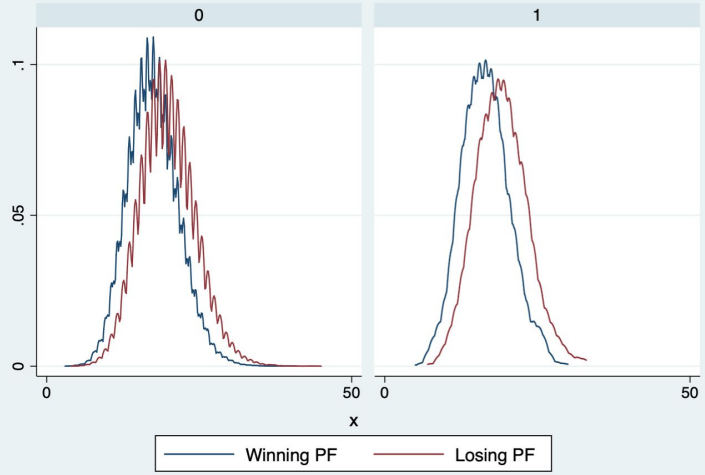
Graphs by tournament



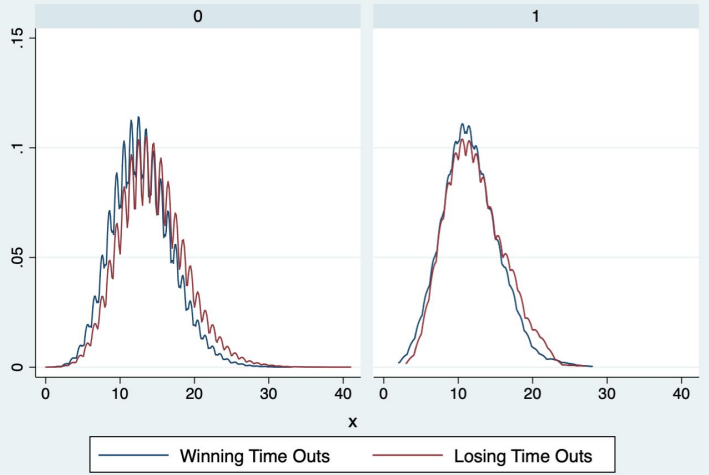
Graphs by tournament



Graphs by tournament



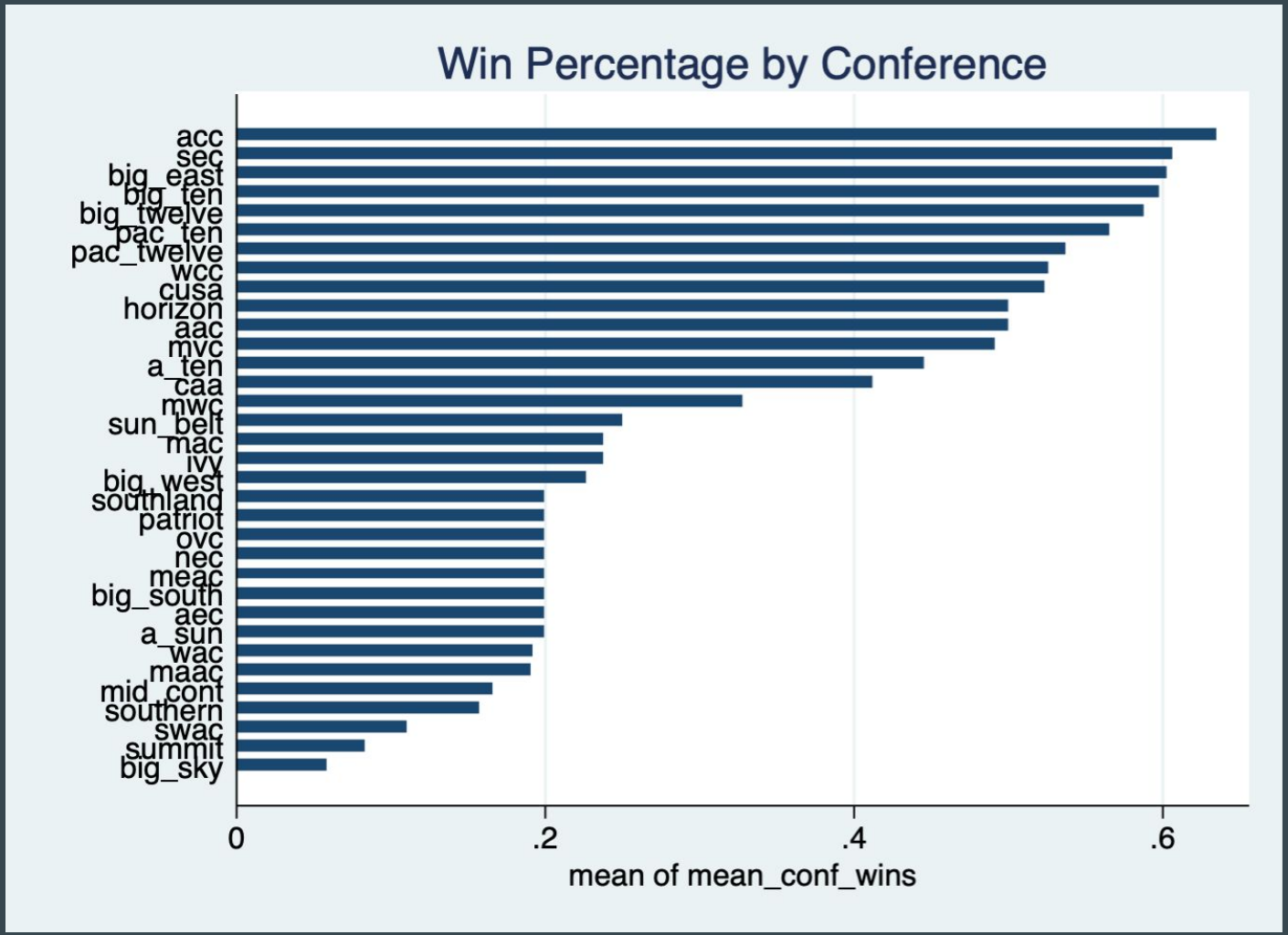
Graphs by tournament

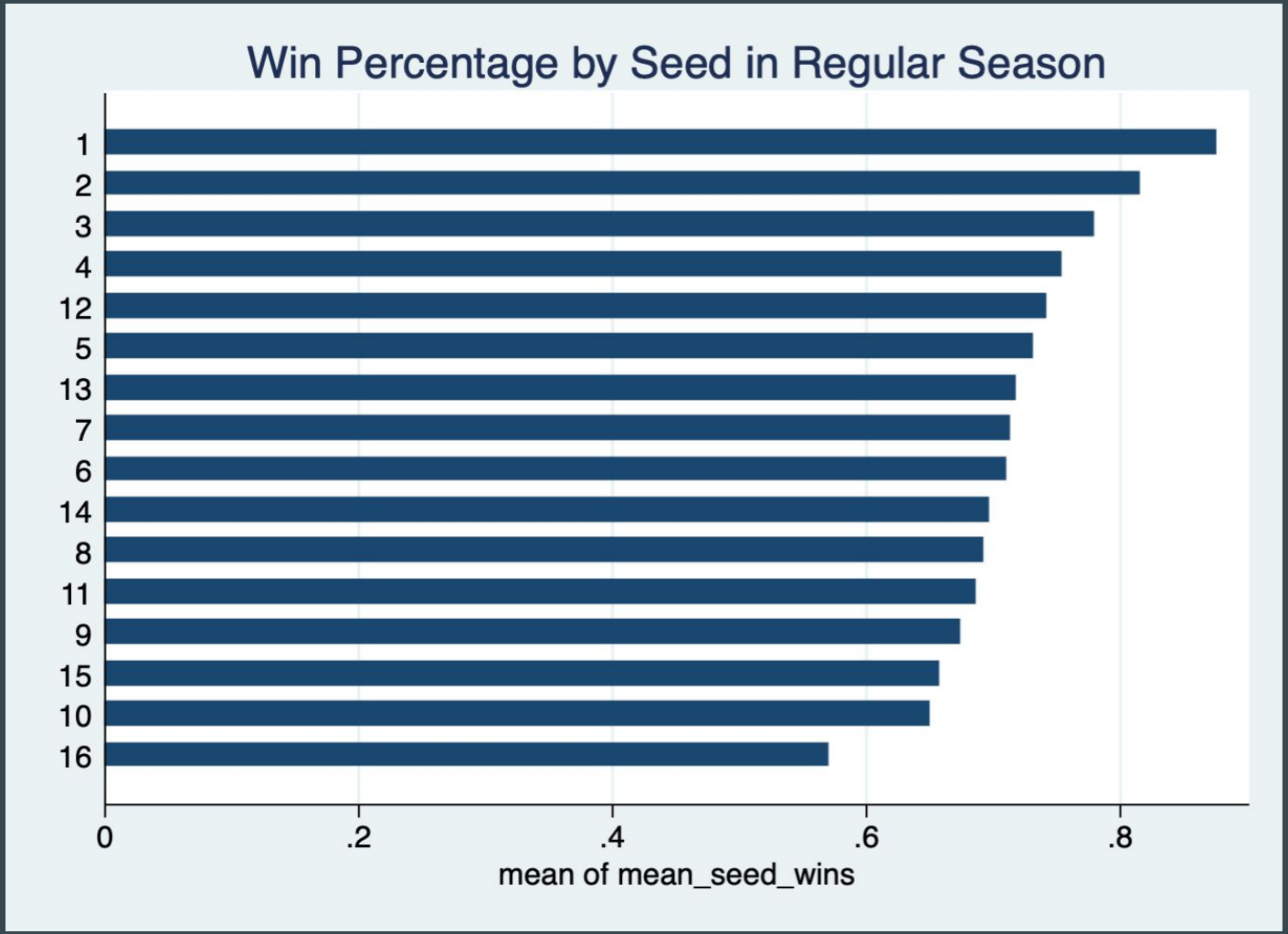


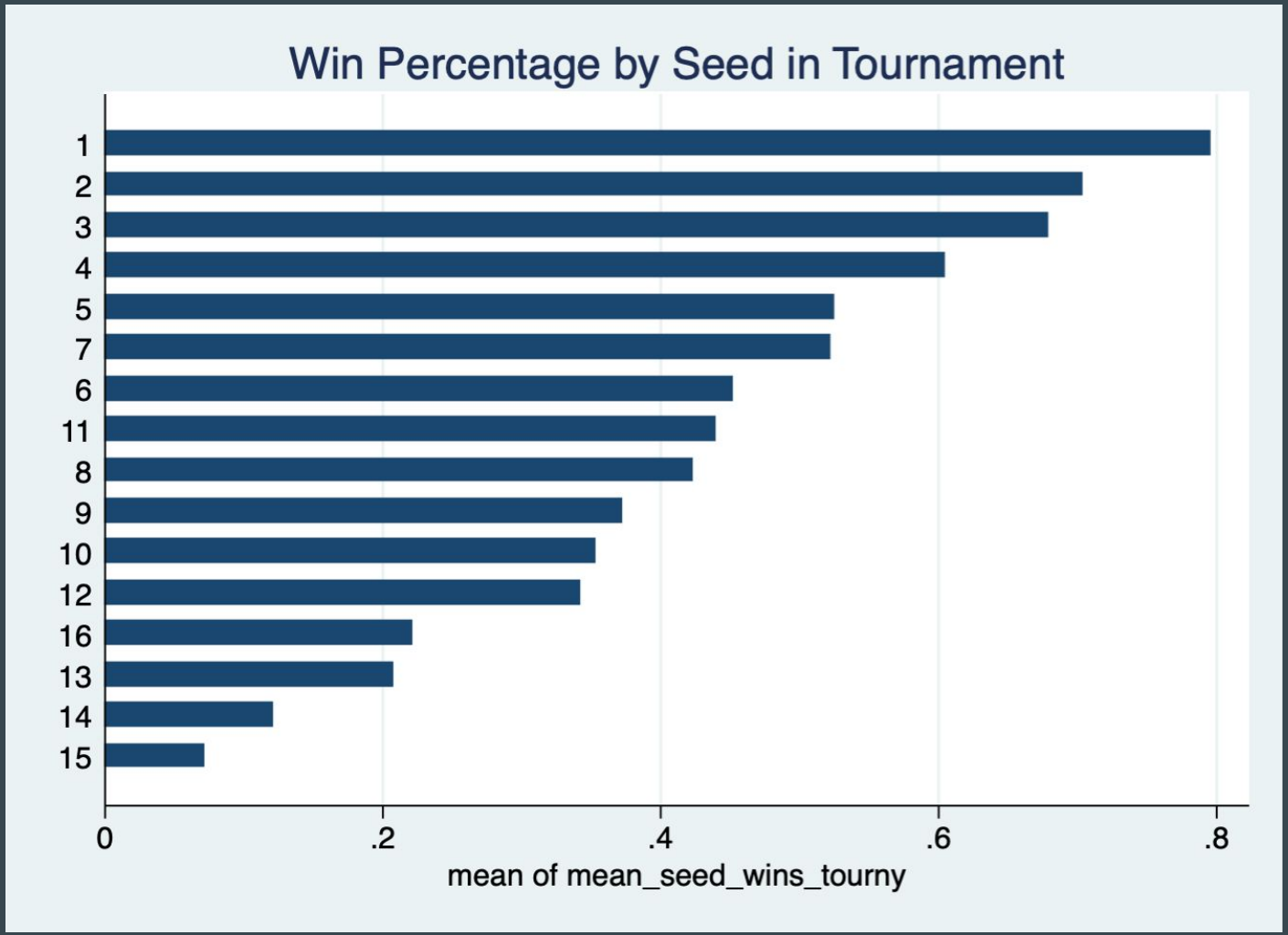
Graphs by tournament

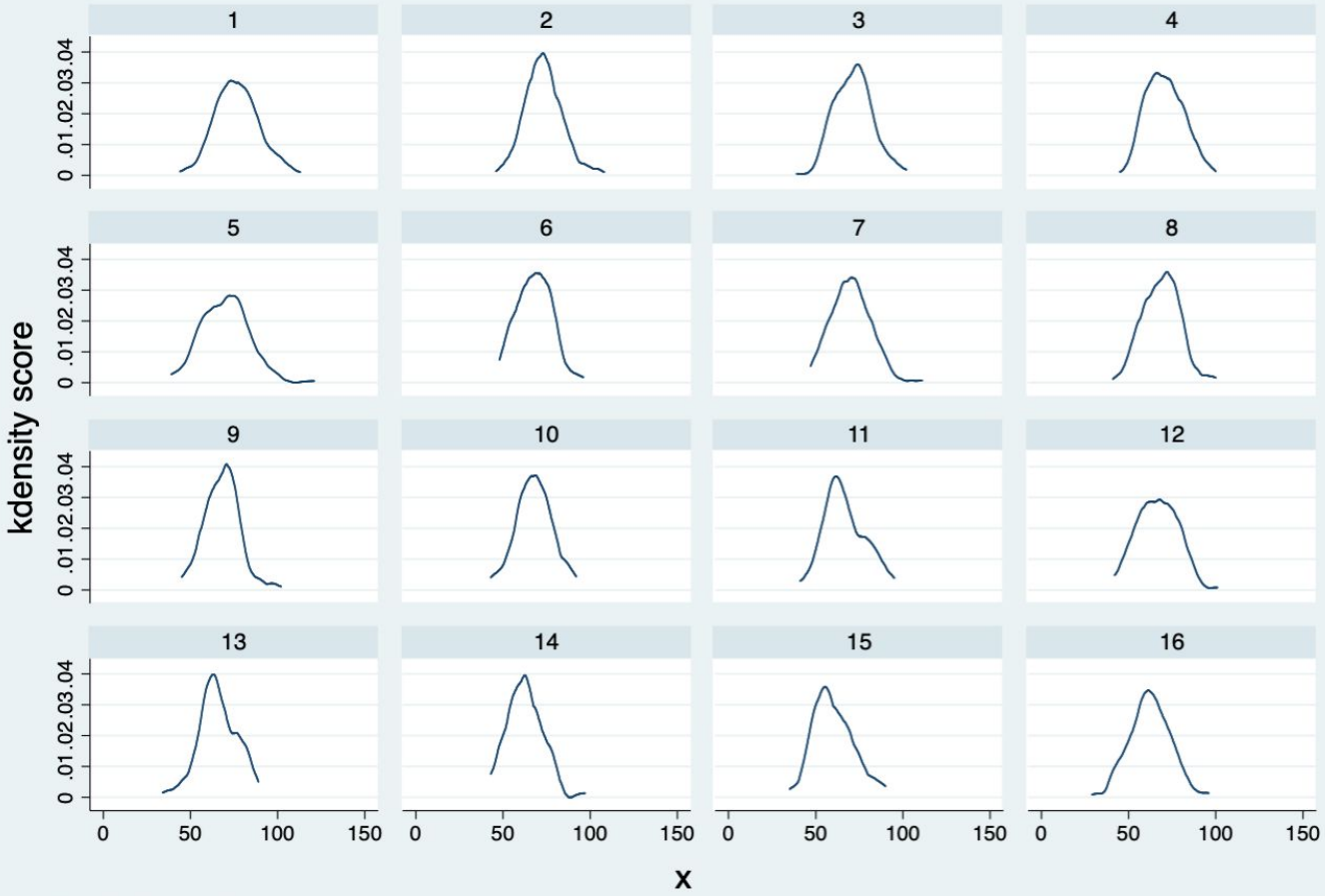

2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
0
10
20
30
LAst
0
10
20
30
40
50
LDR
0
10
20
30
40
LOR
0
5
10
15
20
LStl
0
10
20
30
40
50
LPF
20
40
60
80
100
LFGA
0
20
40
60
LFGA3
0
10
20
30
40
LPF
0
5
10
15
20
LBlk
10
20
30
40
50
LFGM
0
5
10
15
20
LFGM3
0
50
100
LScore


2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
WAst
WFGA
WFGM
WOR
WBlk
WFGA3
WFGM3
WPF
WDR
WFTA
WFTM
WScore


Win Percentage by Conference
acc
sec
big_east
big_ten
big_twelve
pac_ten
pac_twelve
wcc
cusa
horizon
aac
mvc
a_ten
caa
mwc
sun_belt
mac
ivy
big_west
southland
patriot
ovc
nec
meac
big_south
aec
a_sun
wac
maac
mid_cont
southern
swac
summit
big_sky
0
.2
.4
.6
mean of mean_conf_wins


Win Percentage by Seed in Regular Season
1
2
3
4
12
5
13
7
6
14
8
11
9
15
10
16
0
.2
.4
.6
.8
mean of mean_seed_wins


Win Percentage by Seed in Tournament
1
2
3
4
5
7
6
11
8
9
10
12
16
13
14
15
0
.2
.4
.6
.8
mean of mean_seed_wins_tourny


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
kdensity score
0 .01.02.03.04
0
50
100
150
x
Graphs by seed_num

Logistic Regression Markov Chain

Using Seeding to Predict Winners

(1) Transition Probabilities: $t_{ij} = \frac{1}{N_i}[\sum_{O=(i,j)}(1-p)I_{ik} + \sum_{O=(j,i)}(1-p)(1-I_{ij})]$

However, we want to predict based on a seeding, not specifically whether the team will win or lose, so we need to think about the transition probabilities differently.

(2) Adjusted Transition probabilities: $t_{ij} = \frac{1}{N_i}[\sum_{g=(i,j)}(1-r^B_{x(g)}) + \sum_{g=(j,i)}(1-r^W_{x(g)})]$

(3) Probabilities: $r^B_{x(g)} + r^W_{x(g)} = 1$

The resulting probabilities are used to seed the transition matrix and construct a measure of steady-state probabilities which are used to construct the seeding.

	Polls			Mathematical Models			Las Vegas Predictions			
	AP	ESPN	Seed	RPI	Massey	Sagarin	KG	Sheridan	Vegas	LRMC
Games won by predicted winner	236	235½	235¼	229	242	229	231½	244½	n/a	248
Games won by higher-ranked team	266	266½	265	262	268	264	260	268½	273½	277

Table 1. Performance of models on two metrics of prediction quality, 1999-2000 through 2004-2005 seasons (378 total games).

Kvam and Sokol (2006)

Prediction method (x)	LRMC correct, x not (# of games)	x correct, LRMC not (# of games)	One-tailed significance
AP	30	17	0.04
ESPN	29	16	0.04
Seed	32	19	0.05
RPI	37	22	0.03
Massey	30	21	0.13
Sagarin	24	11	0.02
KG	32	16	0.01
Sheridan	27	15	0.04
Vegas	20	16	0.31

Table 2. Significance of LRMC's performance against other game-by-game prediction methods, 1999-2000 through 2004-2005 seasons (378 total games).

Kvam and Sokol (2006)

LRMC - WD	2000	2005	2010	2019
Correct	41 (65%)	45 (70%)	39 (61%)	46 (69%)
Incorrect	22	19	25	21
Total Games	63	64	64	67

Predicting Game Wins

Data

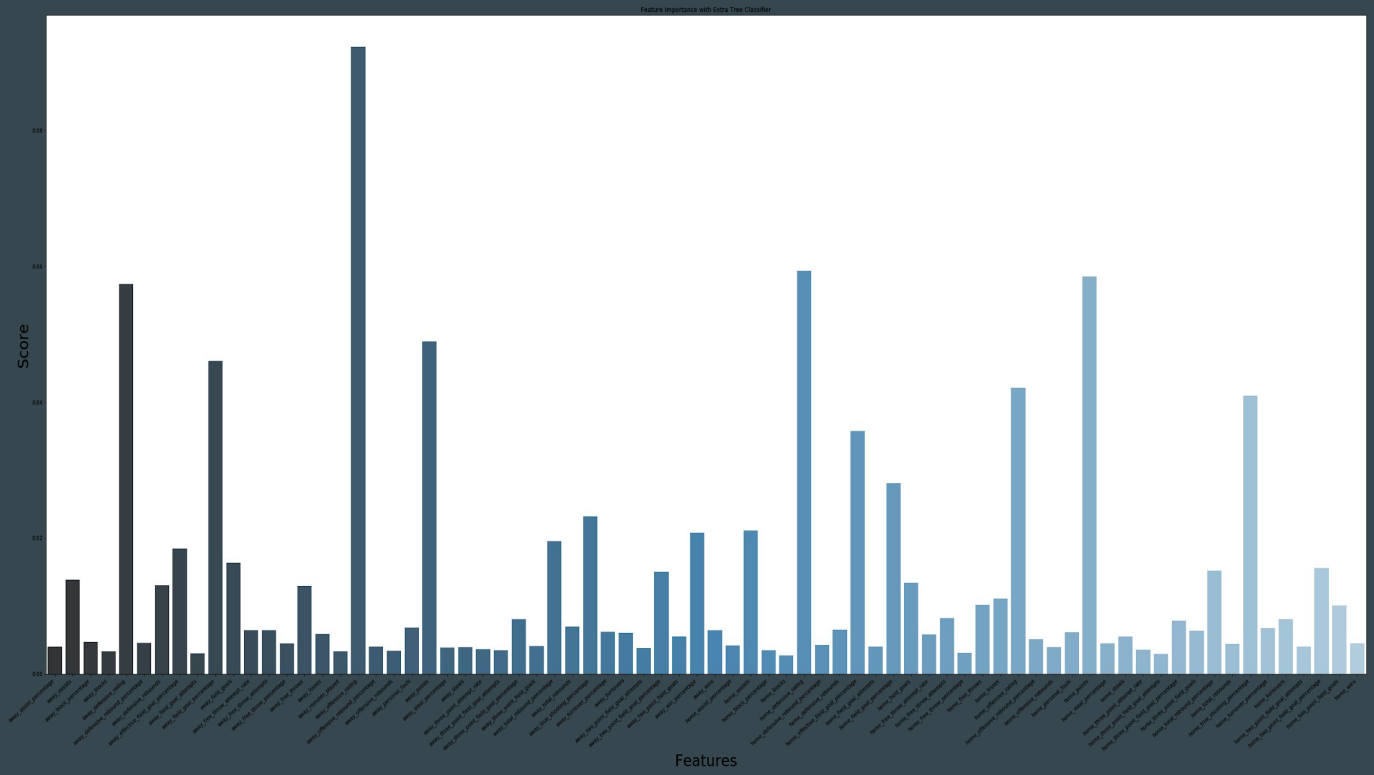
- Fairly large dataset
- All games played by all NCAA registered collegiate basketball teams in any season
- Looked at most recent 2018-2019 season
- Features (a few):
  - Field_Goal_Percentage
  - Assists
  - Turnover_Rate
  - Defensive Rating
  - Away_Minutes_Played (Mostly always 200)
  - Total_Rebound_Percentage
  - Free_Throw_Percentage
- Not all important

Feature Selection

- While our intuition from watching college basketball indicates Points, FG%, TOV would be the best features to predict team success, feature selection algorithms automatically select features most useful to the problem at hand

- In doing so we get the following benefits:
  1. Improves accuracy
  2. Reduces model complexity
  3. Trains Faster
  4. Reduced Overfitting

Feature Selection: Extra Tree Classifier



Feature Importance

1: Off Rating

2: Def Rating

3: Points

4: FG%

5: TS%

Feature Selection: RFE using Logistic Regression

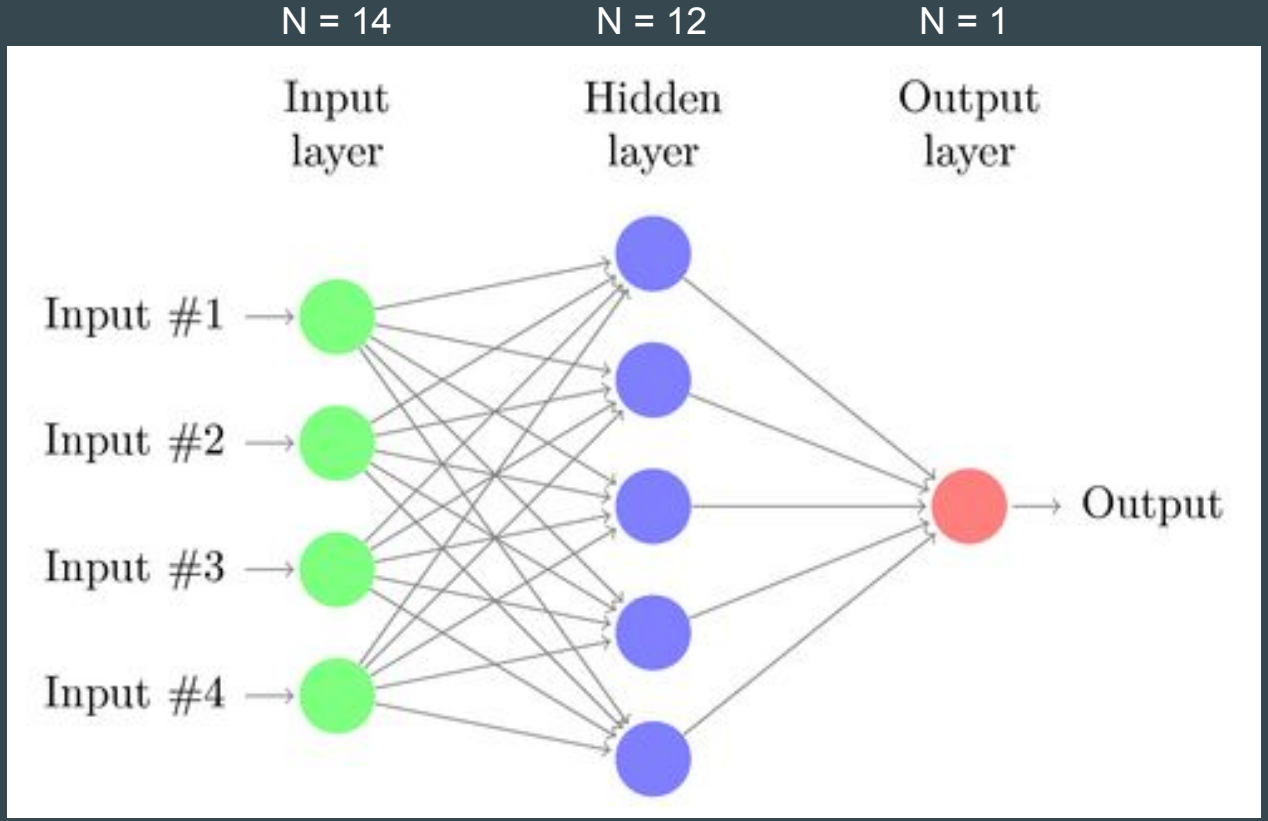
1. Defensive Rating
2. Offensive Rating
3. Points
4. Field Goals
5. Three Point Field Goals
6. Free Throws
7. Wins
8. Blocks
9. Turnovers
10. Losses

1. Two Point FG
2. Fouls
3. Offensive Rebounds
4. Defensive Rebound %
5. Total Rebounds
6. Assists
7. Off Rebound %
8. FT Attempts
9. Fouls
10. 3 Point FG Attempts

Neural Network (from scratch)

- Simple back-propagating neural network
- 3 layers: input, hidden, output
- Output mapped to a win for the **HOME** team or the **AWAY** team
- Used sigmoid activation function
- Probably have a huge problem of overfitting due to large dataset and sporadic data
- ~77% accuracy rate, with a lot of variance depending on the features selected

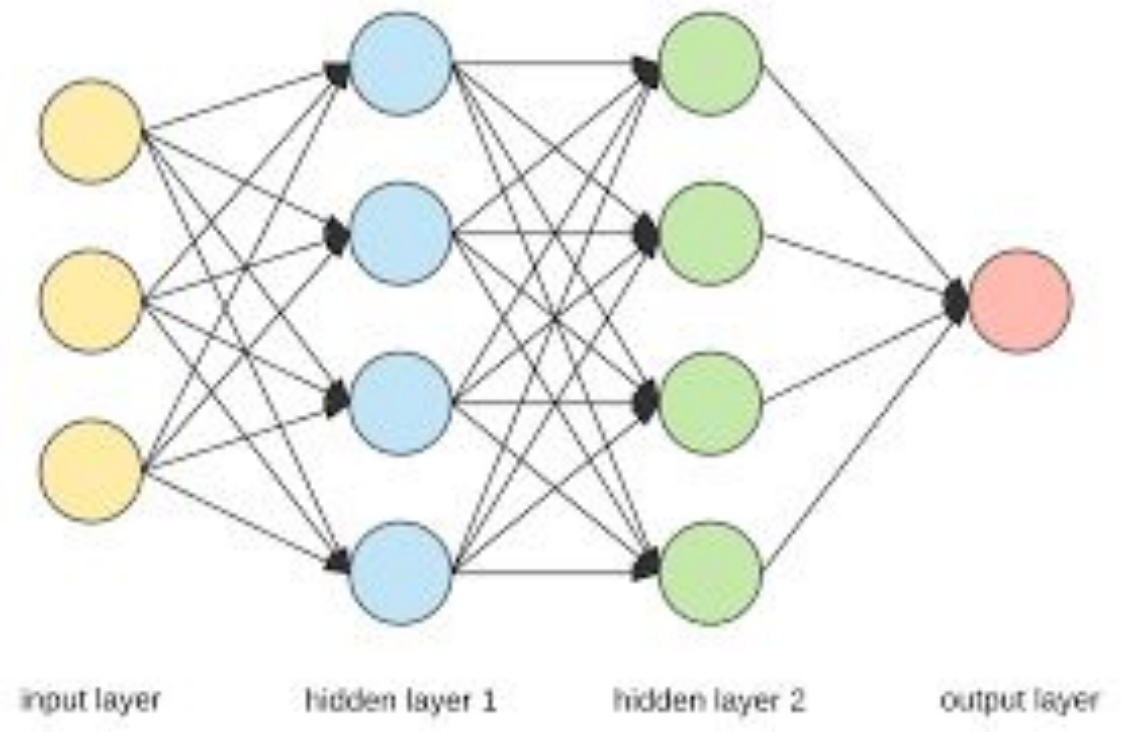
First Neural Network with Keras



Specifications on First Neural Network

- 14 input nodes in input layer
- 12 neurons in hidden layer (trial and error)
- 1 output node (Predicting win or loss)
- Rectifier (‘relu’) activation function on input and hidden layers
- Sigmoid activation function in the output layer
- Logarithmic loss function for binary classification (binary_crossentropy)
- Adam as efficient gradient descent algorithm (optimizer = ‘adam’)
- ~80 Percent Accuracy

Second Neural Network with Keras



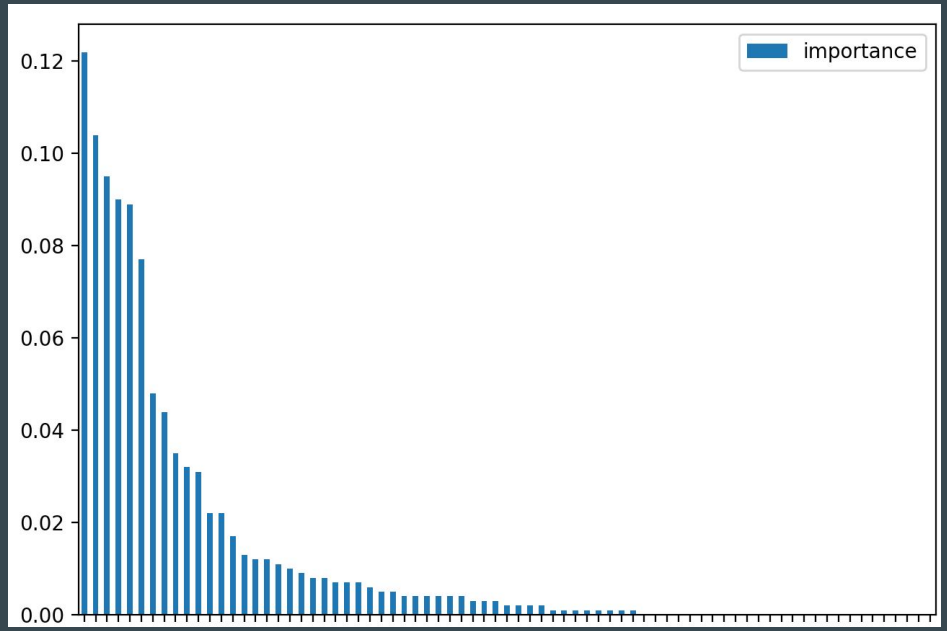
N = 14 N = 32 N = 32 N = 1

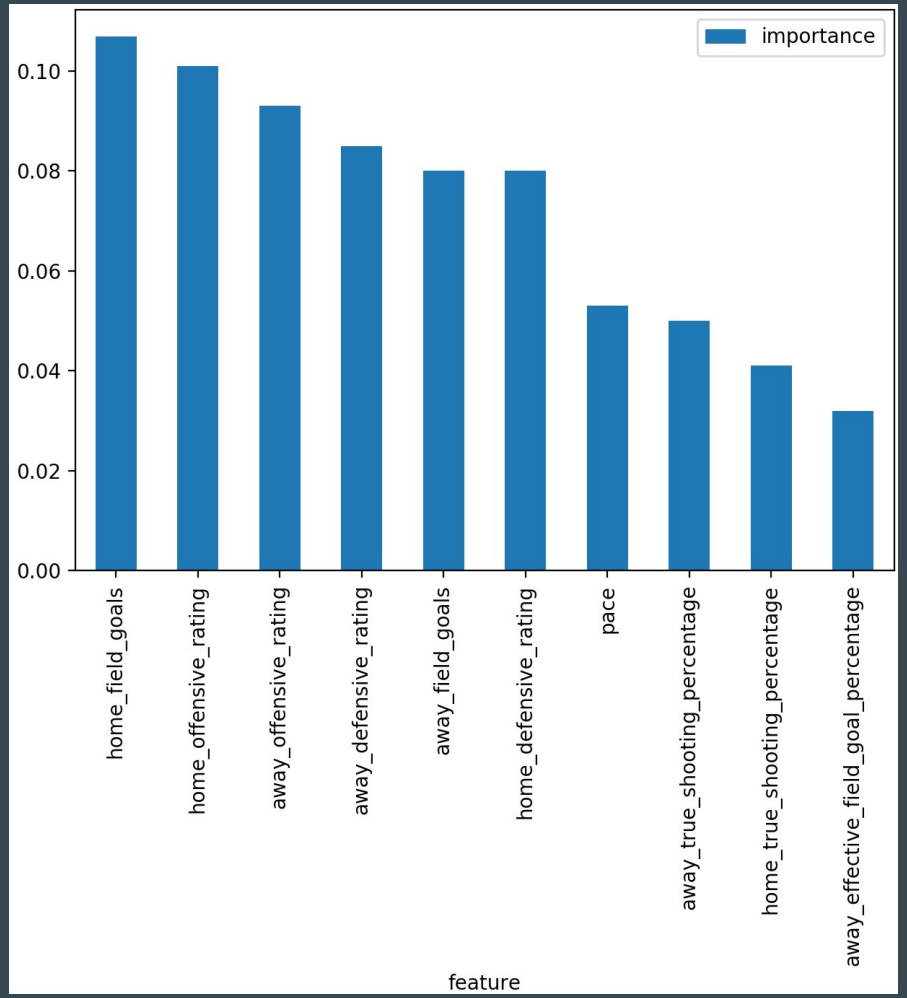
Specifications on Second Neural Network

- 14 input nodes in input layer
- Two Hidden Layers
- 32 neurons in first hidden layer
- 32 neurons in second hidden layer
- 1 output node (Predicting win or loss)
- Rectifier (‘relu’) activation function on hidden layers
- Sigmoid activation function in the output layer
- Logarithmic loss function for binary classification (binary_crossentropy)
- Stochastic gradient descent as optimizer (optimizer = ‘sgd’)
- ~85 percent accuracy

Random Forest Regressor

- Model Trained on 75 Features
- Feature Importance done automatically





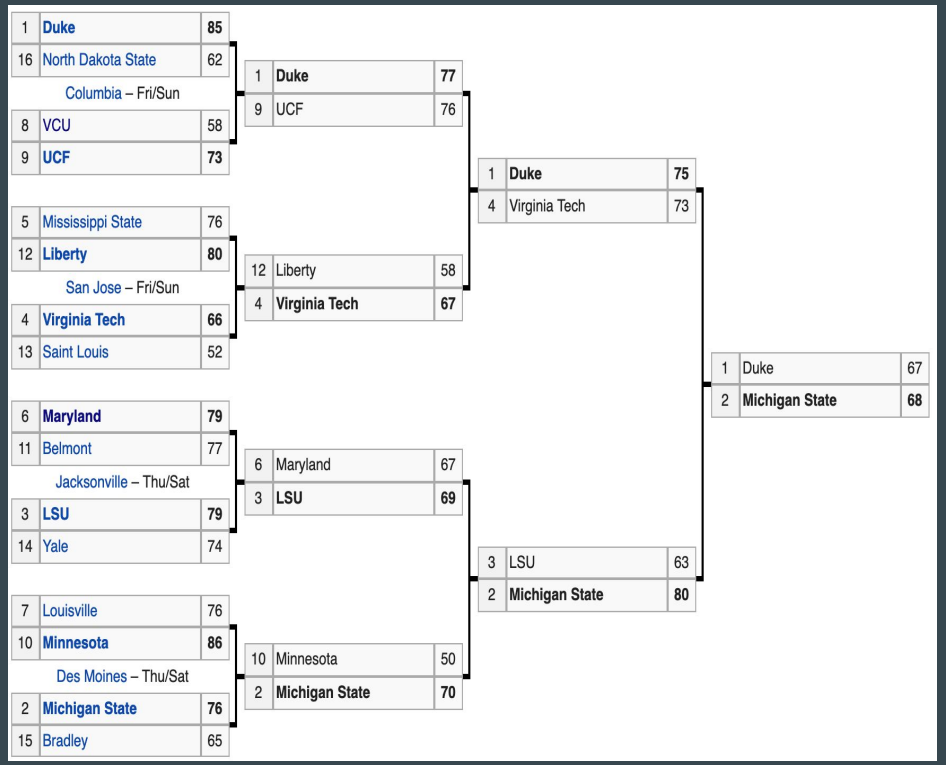
Random Forest Regressor Modeling

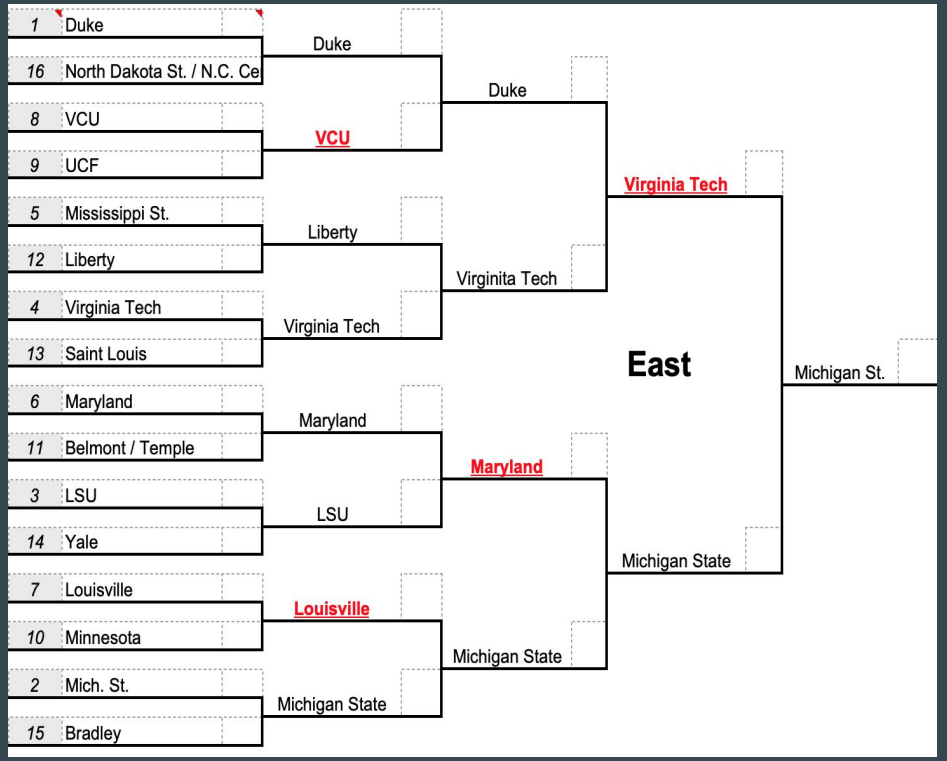
- Our model at first became great at taking a game's statistics and outputting a final score
- It's not that easy because we just don't have all the statistics of how a game is going to play out before it is played
    - However, this could potentially be used to detect point shaving and other forms of match fixing

How were we going to predict game outcomes?

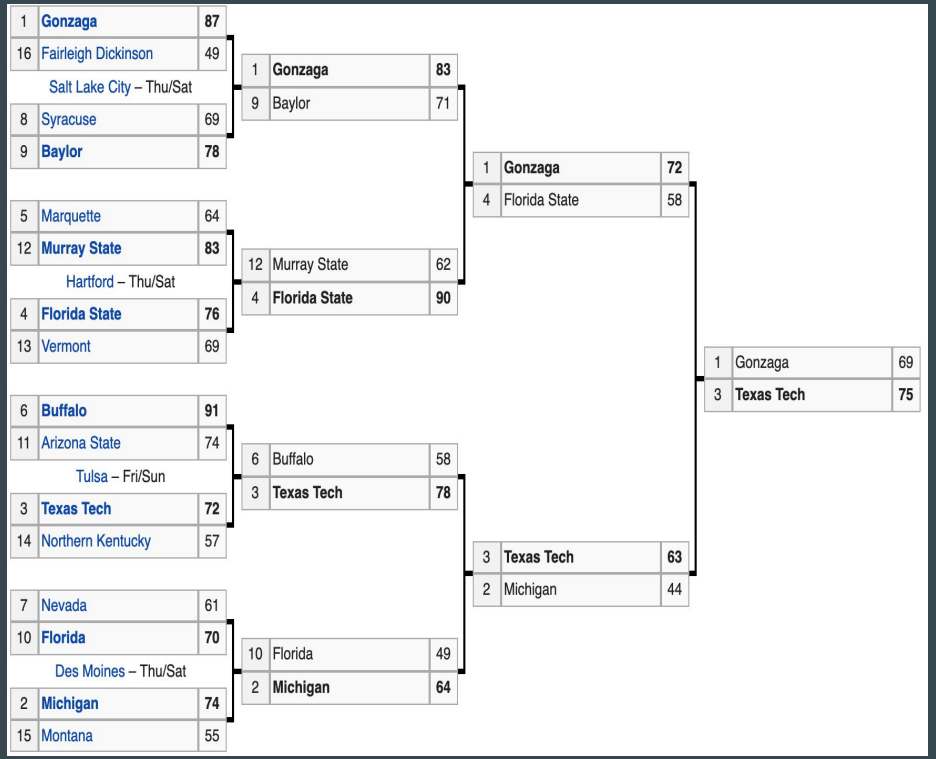
- Built a model of how each team played based on the average of their games and put those models up against each other in the algorithm

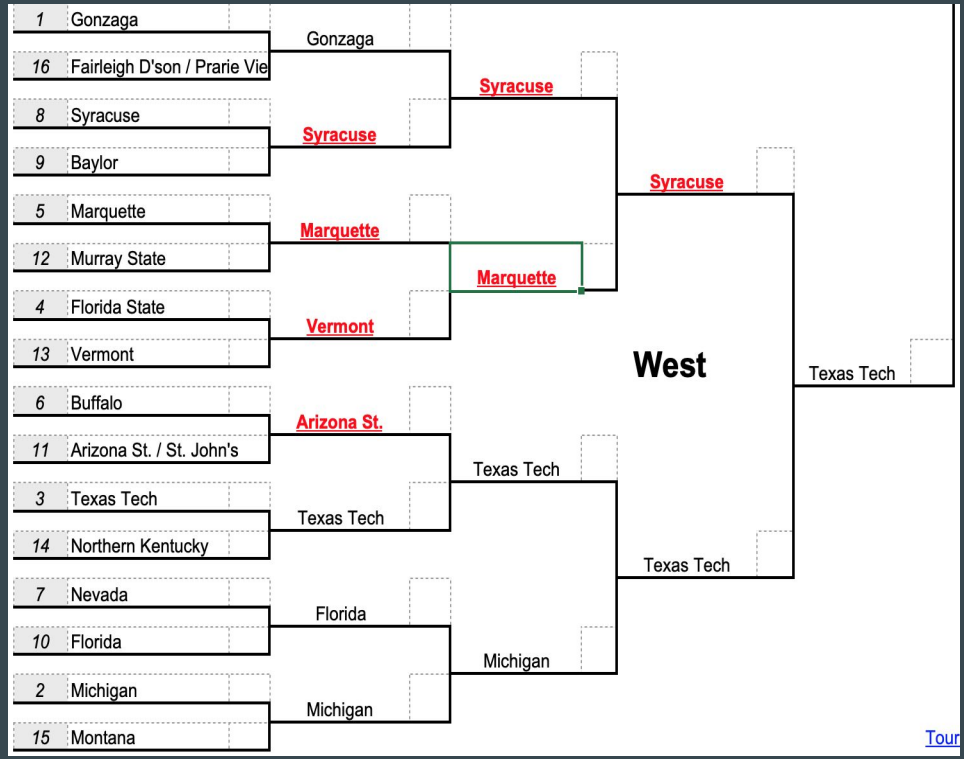
First Round Bracket Predictions (East)



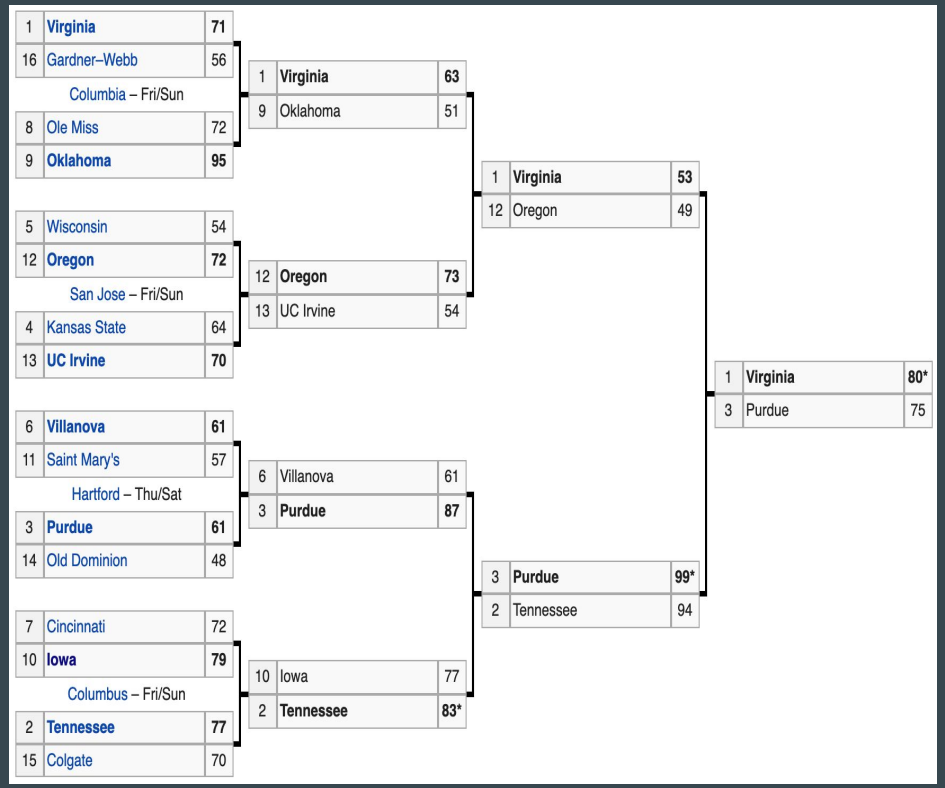


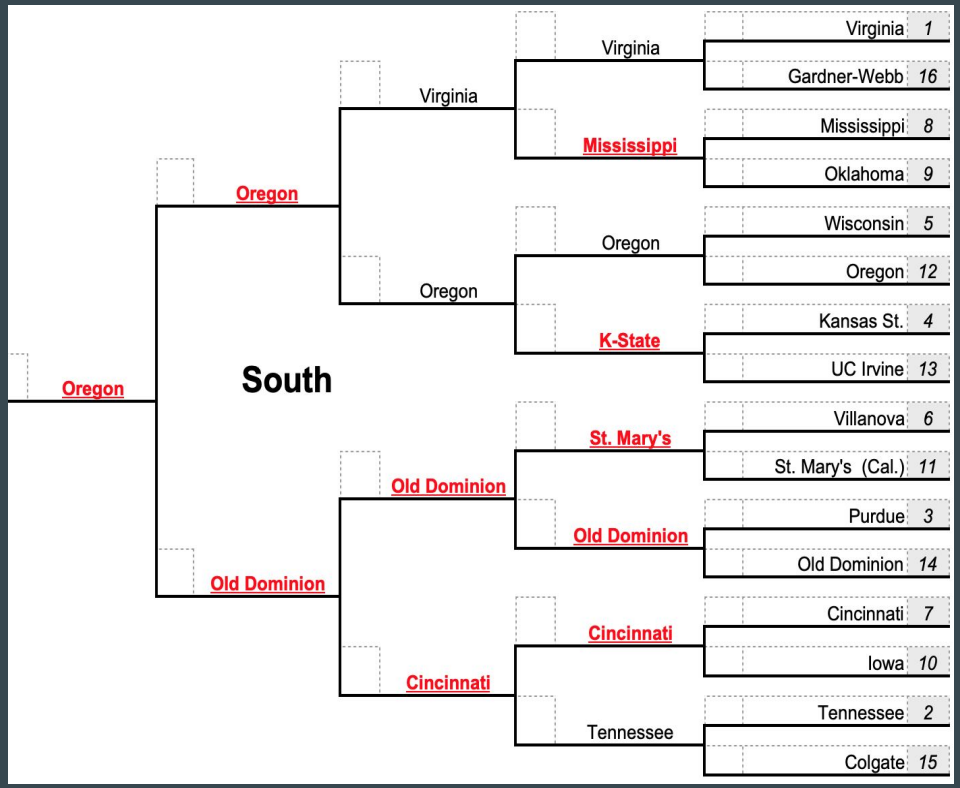
First Round Bracket Predictions (West)



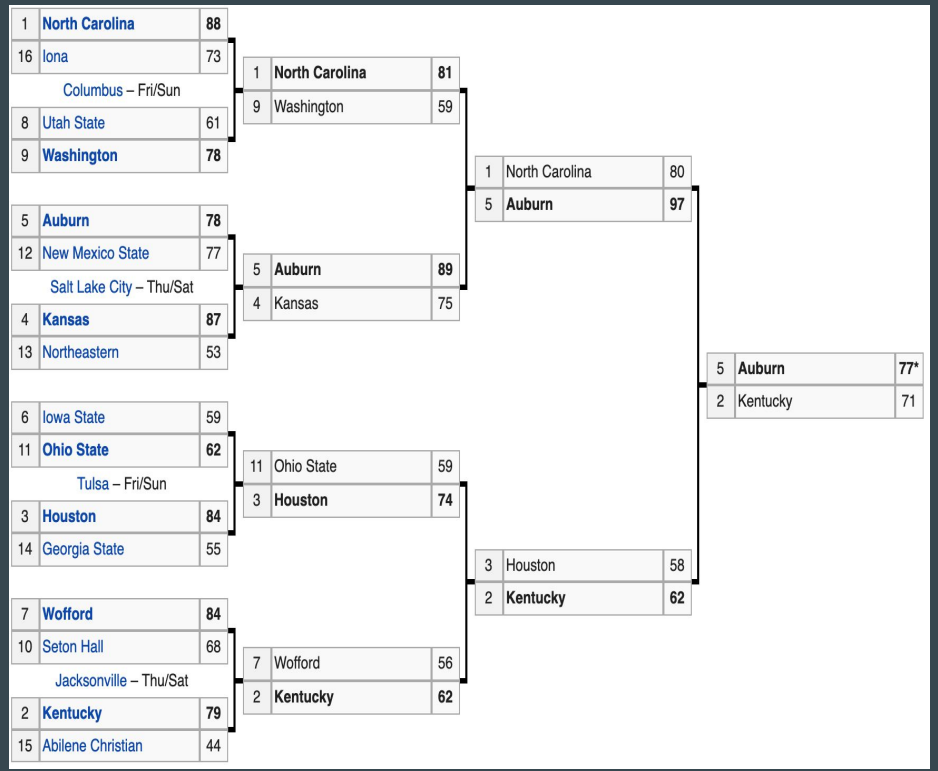


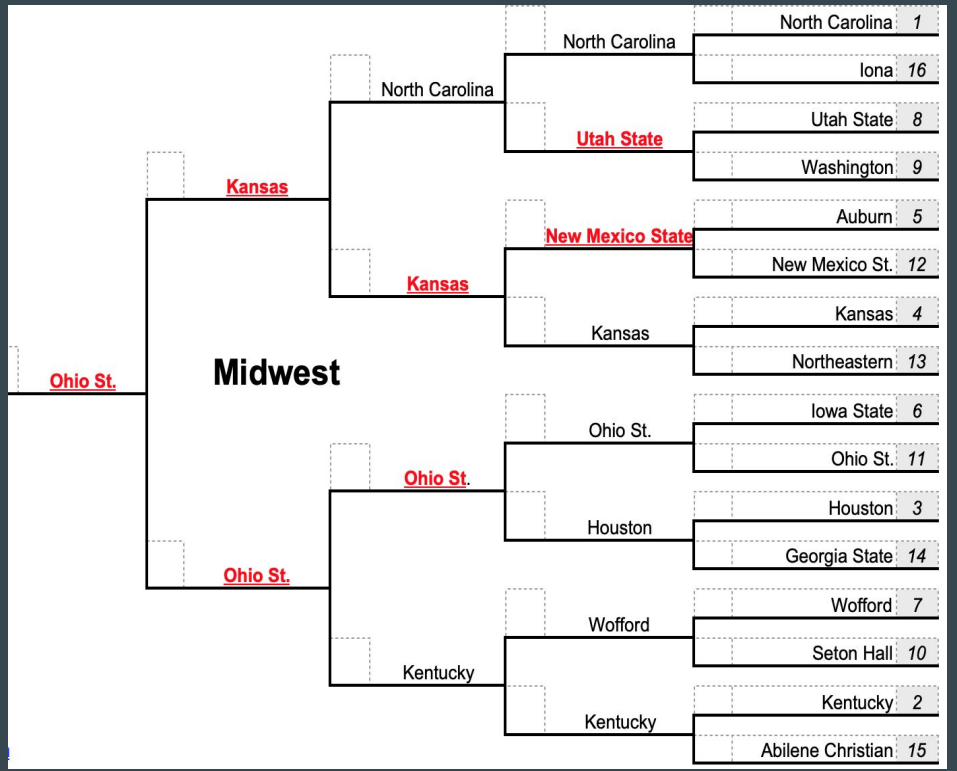
First Round Bracket Predictions (South)



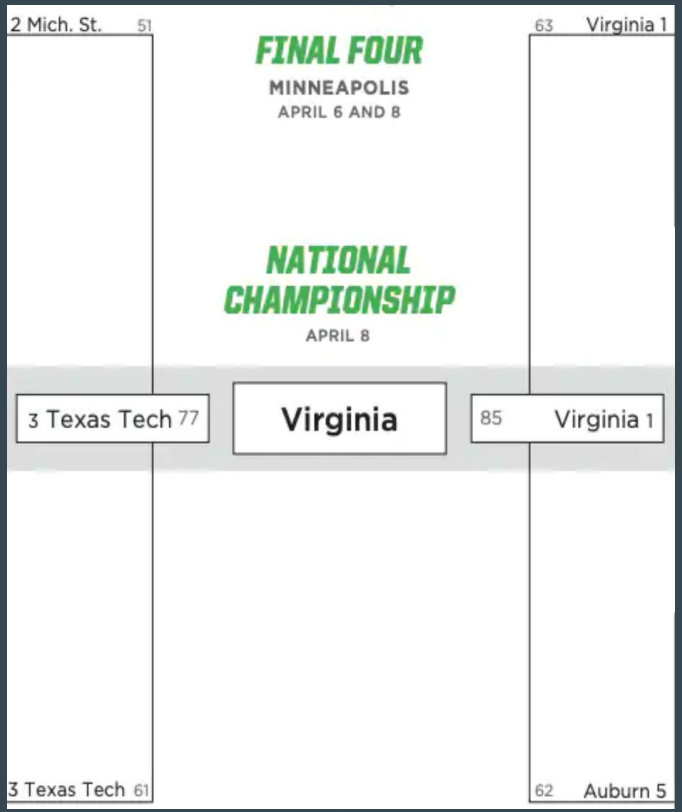


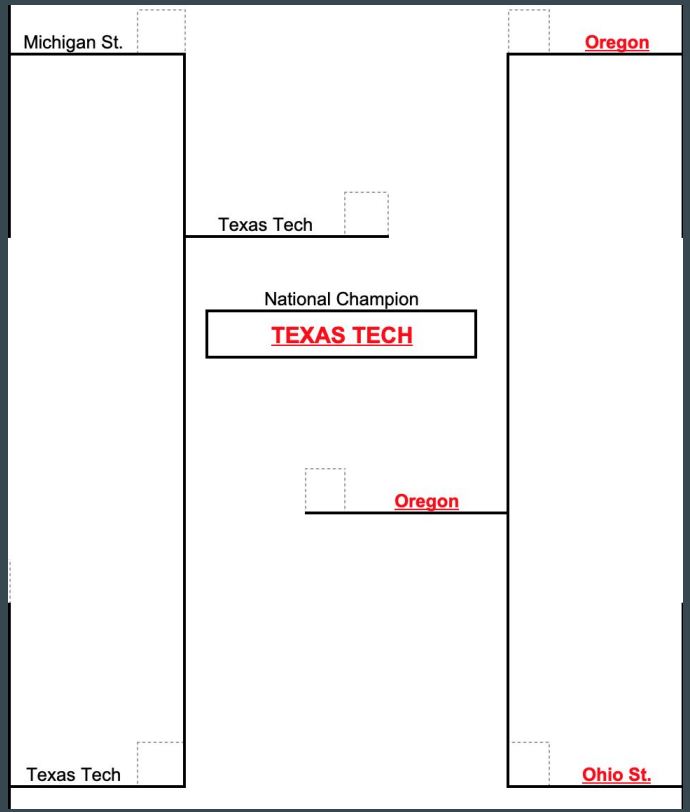
First Round Bracket Predictions (Midwest)





Final Four/Championship





In the future

- Perform more trial and error experimentation for neural network (testing different number of hidden layers/neurons)
- Add new features to input layer to incorporate basketball domain knowledge
- Add individual player statistics to model
- Develop model so that it also takes matchups into account
- Run model more than once for predicting tournament games

Questions

...