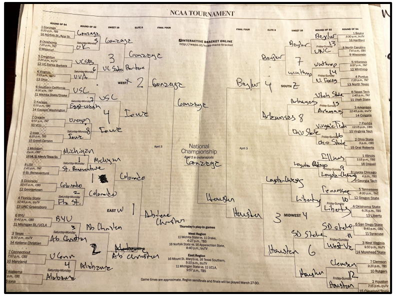
Predicting the NCAA Men's CBB Tournament

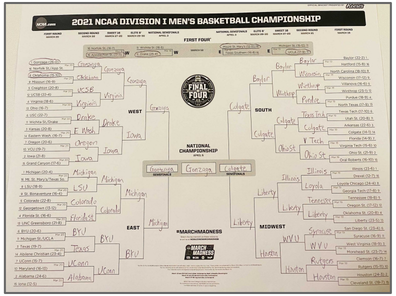
Sean Norris, Ren Tu, and Mikayla Pugel

What we did…

Tried to predict the NCAA Tournament…



Point Differential Model



Possession Model

..for a tournament where…

70,000,000 brackets are filled out each year

1 in 9.2 quintillion are the chances of a perfect bracket

0.025% Of people had this year's Final Four predicted correctly

How we did it...(Point Differential Model)

List of teams with aggregate season point differential

Team_1_ID	plus_minus
1101	312
1102	-330
1103	80

Schedule with season point differential and winner populated;
Team 1 and 2 positions were randomized to eliminate "home court" bias

Team_1_plusminus	Team_2_plusminus	Winner
53	128	-1
250	-18	1
89	-200	1

"X" Data

"Y" or "Target" Data

Training and Test data were split 80/20

"Three" lines of Machine Learning Code

```
logreg = LogisticRegression()
logreg.fit(x_train, y_train)
logreg_pred  = logreg.predict(x_test)
f1_score(y_test, logreg_pred, average = "weighted"
```

F1 Score - Logistic Regression: 73%

How we did it...(Possession Model)

Season-level differentials across possession-oriented features

Points Per Possession Difference	Opponent Points Per Possession Difference	Possessions Per Game Difference	Home Court Advantage	Points Per Possession Standard Deviation Difference	Bad Plays Per Possession Difference (e.g. turnovers, fouls, blocks)	Winner
2.1	-0.3	0.5	0	1.2	-1.9	1
-1.2	1.3	1.8	1	-0.7	2.5	0
0.5	0.8	-0.9	-1	2.1	-0.3	1

"X" Data

Training and Test data were split 80/20

"Y" or "Target" Data

F1 Score - Logistic Regression: 77%

F1 Score - Random Forest: 77%

F1 Score - Gradient Boosting: 77%

F1 Score - Naive Bayes: 75%

F1 Score - KNN: 73%

F1 Score - 5 Model Ensemble: 77%

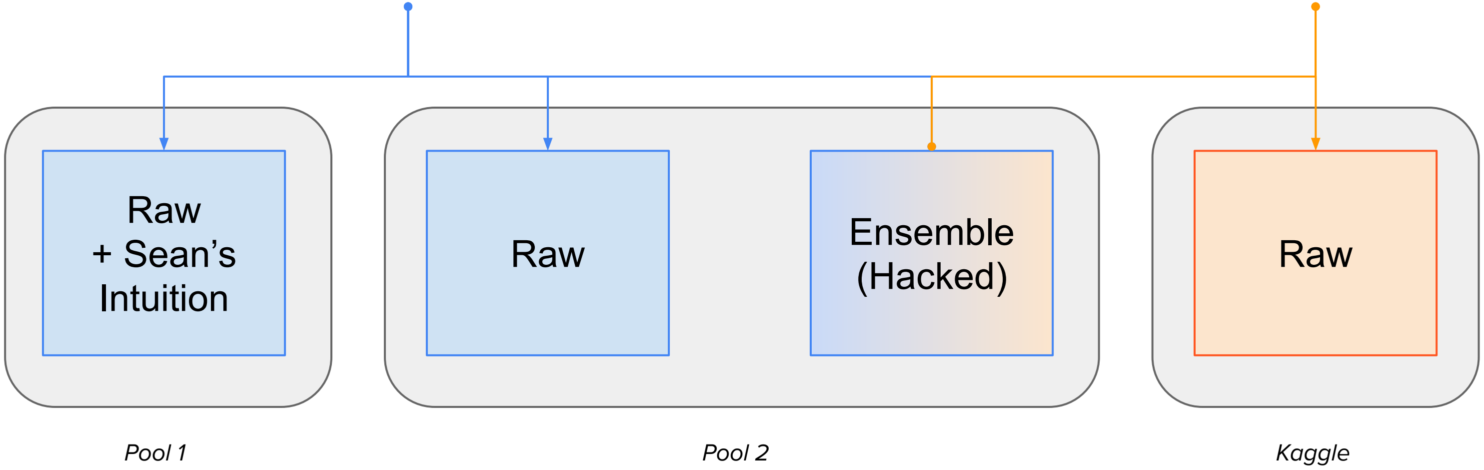
How we tested our models...

Point Differential Model

Fed annual point differential aggregations into a logistic regression classifier that trained on CBB outcomes from 2015 - 2020

Possession (Match-Up) Model

Fed annual differentials in several categories (e.g., possessions, etc.) into a logistic regression classifier that trained on CBB outcomes from 2010 - 2020



What Happened?

Baylor beat nearly everyone by double digits

- Round 1: **Won by 24**
- Round 2: **Won by 13**
- Sweet 16: **Won by 11**
- Elite 8: **Won by 9**
- Final Four: **Won by 19**
- Championship Game: **Won by 16**

...and no one had a perfect bracket

So...How did we do?

	Pool 1	Pool 2		Kaggle
	Point Differential + Sean's Intuition	*Raw Point Differential*	*Ensemble (Hacked)*	*Raw Possession Model*
	Finished **10th** of **18**	Finished **7th** of **29**	Finished **6th** of **29**	Finished **100th** of **1,200**
	30 of **63** correct **(47.6%)**	**33** of **63** correct **(52.4%)**	**30** of **63** correct **(47.6%)**	**28** of **63** correct **(44.4%)**
	Two of Final Four correct	Three of Final Four correct	Two of Final Four correct	One of Final Four correct
	Champion: ~~Gonzaga~~	Champion: ~~Gonzaga~~	Champion: ~~Gonzaga~~	Champion: ~~Gonzaga~~

Did we get lucky?

Raw Point Differential

Year	Kaggle Score
2021	0.64
2019	0.57

Raw Possession Model

Year	Kaggle Score
2021	0.60 (Top 10%)
2019	0.59

Kaggle log loss formula:

Team 1 Result * Log(Team 1 Probability) + Team 2 Result * Log(Team 2 Probability)

Lower is better in Kaggle Scores, this year's winner scored 0.55

How we could have done better...

- Adjust for strength of schedule, our models may have overvalued strong teams who played weaker competition
- Add a feature for likelihood of one Team 1 beating Team 2
  - This could take many forms
- Include seeding information (e.g., upsets are more likely in specific matchups)
  - Create a model purely on predicting upsets
- Adjusting point differentials for neutral sites, uneven number of games, and unbalanced schedules (e.g., home versus away)
- Putting greater weights on more recent performance (end of season/most recent seasons)
- Your thoughts?

Appendix

Power 5ish Point Differentials

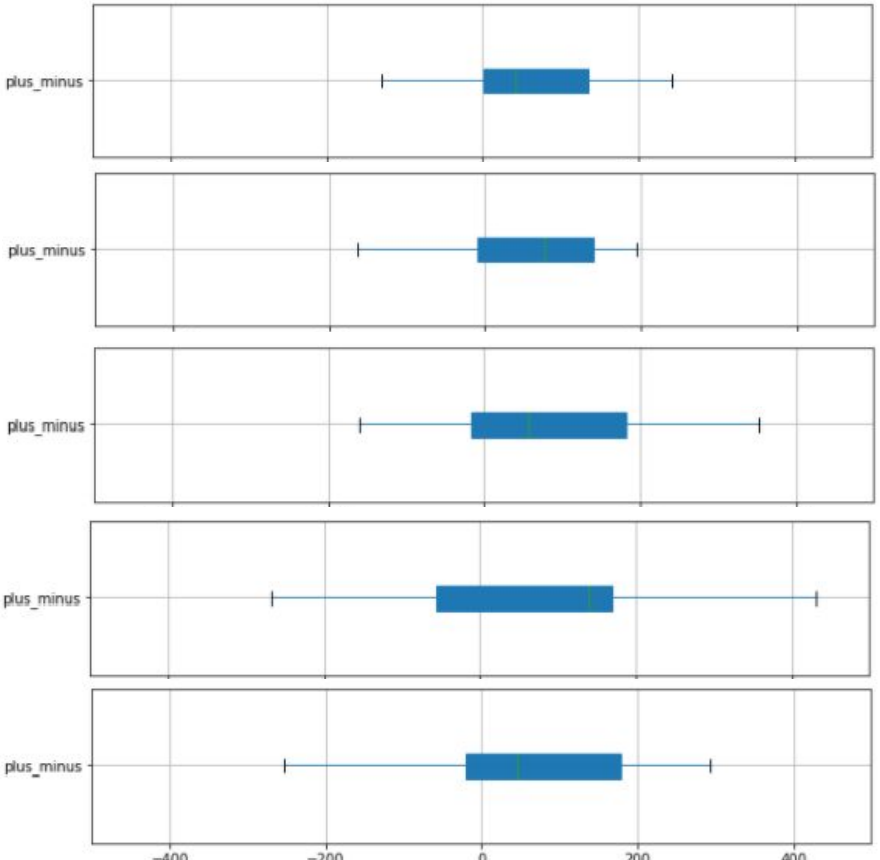
Big East

ACC

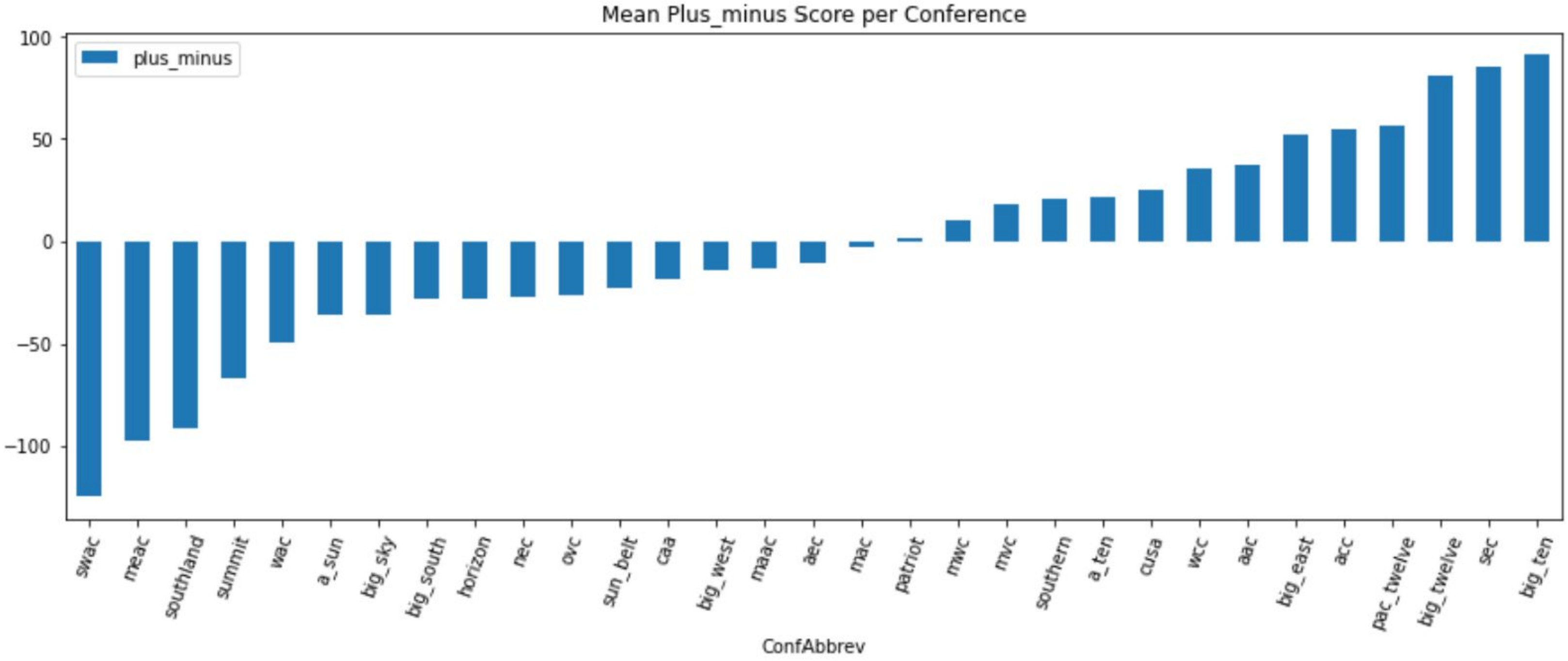
Big Ten

Big Twelve

Pac Twelve



Average Point Differential by Conference



Mean plus_minus per conference

ConfAbbrev	plus_minus
a_sun	-36.666667
a_ten	21.500000
aac	37.272727
acc	54.266667
aec	-11.000000
big_east	52.181818
big_sky	-36.272727
big_south	-28.818182
big_ten	91.214286
big_twelve	81.100000
big_west	-14.545455
caa	-19.000000
cusa	25.285714
horizon	-28.750000
maac	-13.909091
mac	-3.000000
meac	-97.333333
mvc	17.600000
mwc	9.636364
nec	-27.200000
ovc	-26.916667
pac_twelve	56.416667
patriot	1.300000
sec	85.214286
southern	20.600000
southland	-91.461538
summit	-66.777778
sun_belt	-22.916667
swac	-124.700000
wac	-49.444444
wcc	35.600000