Machine Learning with March Madness

Team March Madness

Brett Thompson, Grant Wooldridge, Kris Wasemiller & Scott Frazier

29 March 2021

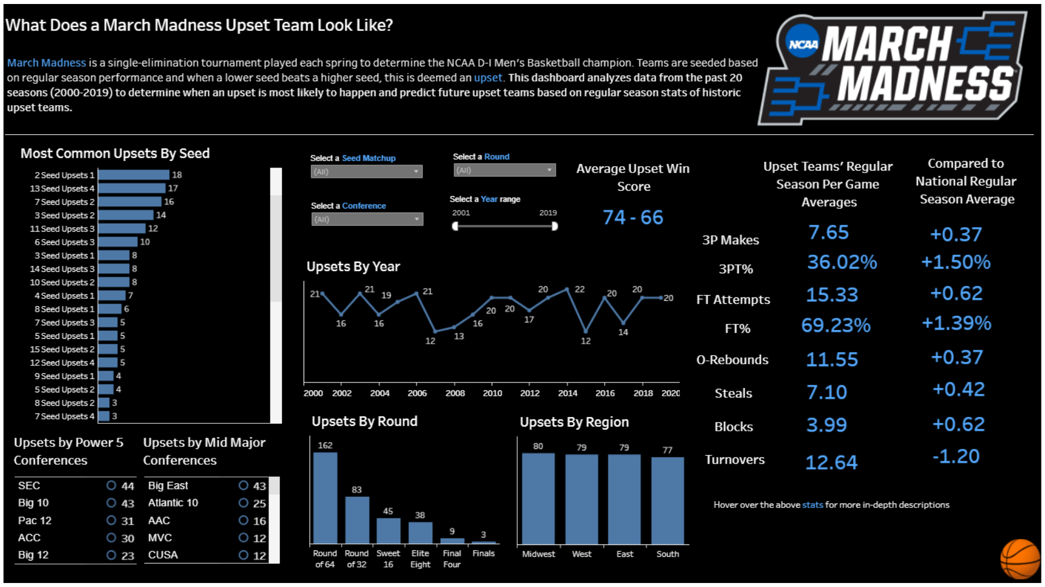
Table of Contents

Inspiration and Overview

For our project, we chose to look at March Madness. Every year around this time the sports world is abuzz about March Madness. Underdogs vs Giants, last second buzzer beaters, and legends that form from each tournament. And with every tournament, comes the tournament bracket challenge! No one has ever created the perfect bracket, so we thought it would be fun to create a machine learning model to help us better predict the men's tournament outcome and each matchup along the way.

We began to build our application around data collected for the 2021 season. Our goal was then to test the strength of our model against past tournaments going all the way back to 2016. Additionally, since the tournament didn't occur in 2020 we would make predictions on who would have been crowned the champion of that year. We decided to power our Flask app using Python, HTML/CSS, Javascript and Tableau.

We found some really cool Tableau visualizations as well as a Kaggle challenge that happens yearly that inspired us on how we would approach diving into the world of March Madness.



What Does a March Madness Upset Team Look Like?

March Madness is a single-elimination tournament played each spring to determine the NCAA D-I Men's Basketball champion. Teams are seeded based on regular season performance and when a lower seed beats a higher seed, this is deemed an upset. This dashboard analyzes data from the past 20 seasons (2000-2019) to determine when an upset is most likely to happen and predict future upset teams based on regular season stats of historic upset teams.

Select a Seed Matchup: (All)

Select a Round: (All)

Select a Conference: (All)

Select a Year range: 2001 – 2019

Average Upset Win Score

74 - 66

	Upset Teams' Regular Season Per Game Averages	Compared to National Regular Season Average
3P Makes	7.65	+0.37
3PT%	36.02%	+1.50%
FT Attempts	15.33	+0.62
FT%	69.23%	+1.39%
O-Rebounds	11.55	+0.37
Steals	7.10	+0.42
Blocks	3.99	+0.62
Turnovers	12.64	-1.20

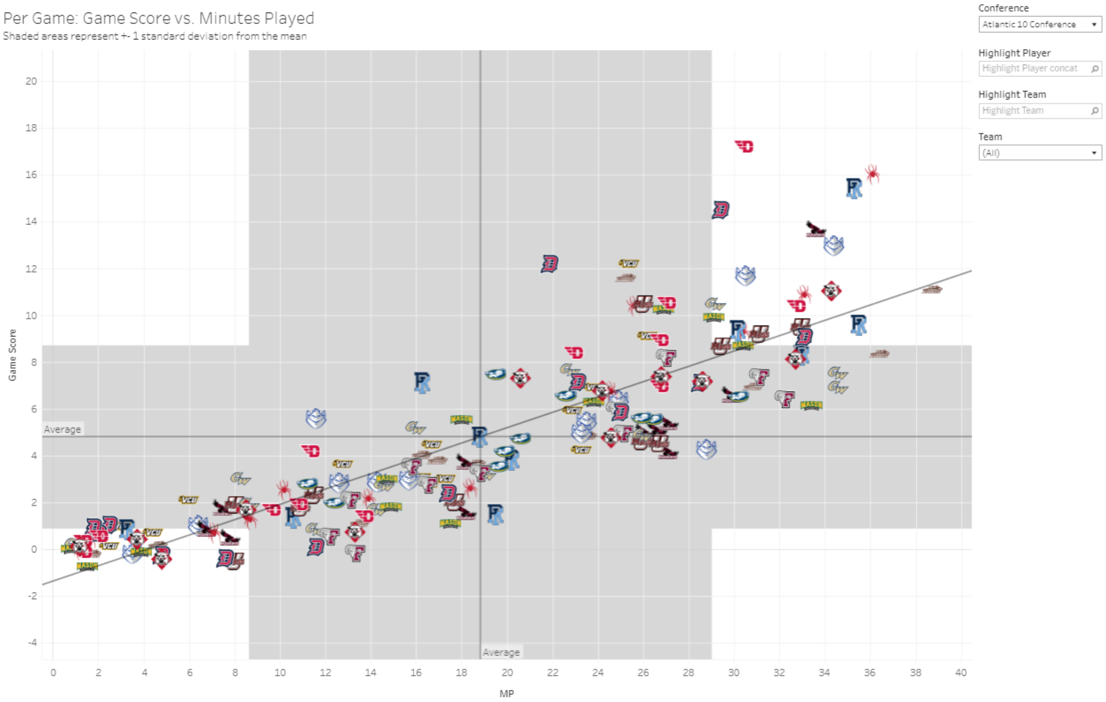
Hover over the above stats for more in-depth descriptions

Upsets by Power 5 Conferences

SEC	44
Big 10	43
Pac 12	31
ACC	30
Big 12	23

Upsets by Mid Major Conferences

Big East	43
Atlantic 10	25
AAC	16
MVC	12
CUSA	12



Per Game: Game Score vs. Minutes Played

Shaded areas represent +- 1 standard deviation from the mean

Conference: Atlantic 10 Conference

Highlight Player: Highlight Player concat

Highlight Team: Highlight Team

Team: (All)

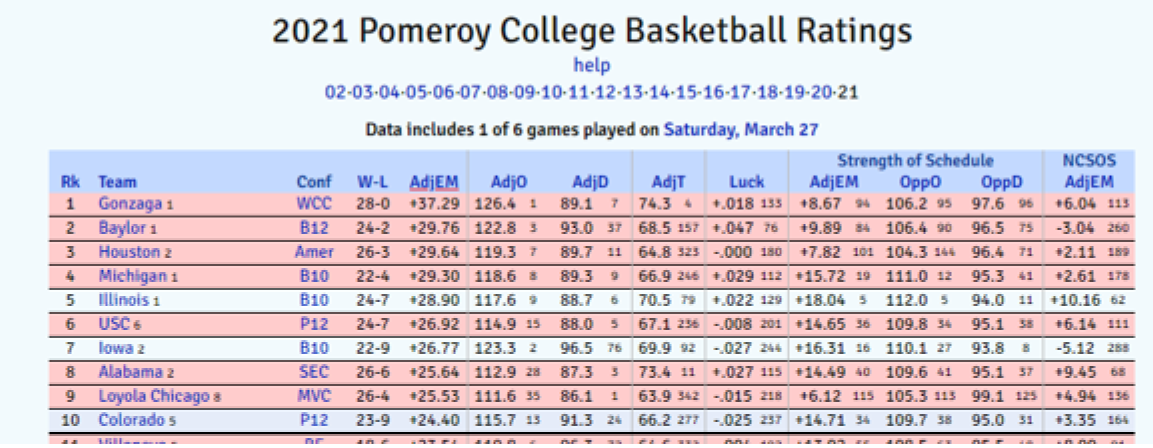
In our analysis of March Madness, we used 8 datasets. 6 of the datasets were used for Tableau visualizations while the others were used for machine learning. The 6 that were used for Tableau visualizations consisted of team names, detailed statistics in regular season play as well as tournament play, coach names, conferences by season and team, and conference names.

Questions to be asked of our data include: (i) Who will win the 2021 NCAA Tournament? (ii) How many points will they score? (iii) Who would have won the 2020 NCAA Tournament had it not been canceled due to COVID 19?

Data & Modeling Approach

Our original datasets required some data cleansing. The detailed statistics datasets, both regular season play and tournament play, had winning team ID and losing team ID that needed to be converted into a singular team ID for Tableau joining. It also had all of it's statistics listed by winning team and losing team. This wouldn't work for visualizations/Tableau unions so the data had to be broken out into new lines with singular titled statistics as well as additional columns to one-hot-encode wins/losses as well as regular/tournament play. Then it had to be joined back together creating a new dataset with all statistics for both regular season and tournament play. From there we let the other 4 datasets from Kaggle be joined to the newly created stats dataset to allow for things like team names, coach names, and conference names to be included in our Tableau visualizations.

For the machine learning data, we utilized two main sources; (1) “KenPom” data, and (2) Kaggle data from the “March Mania” machine learning competition. KenPom data is compiled throughout the season by Ken Pomeroy and hosted on his website. The data consists of advanced basketball analytics that ranks teams by their “adjusted efficiency margin” which is the difference between an individual team’s “adjusted offense” and “adjusted defense” metrics. These metrics are calculated as points allowed and points scored per 100 possessions. The idea of KenPom metrics are to provide the user the “caliber” of a team based on competition they have played throughout the season. The KenPom data on his website can be seen in the screenshot below:



2021 Pomeroy College Basketball Ratings

help

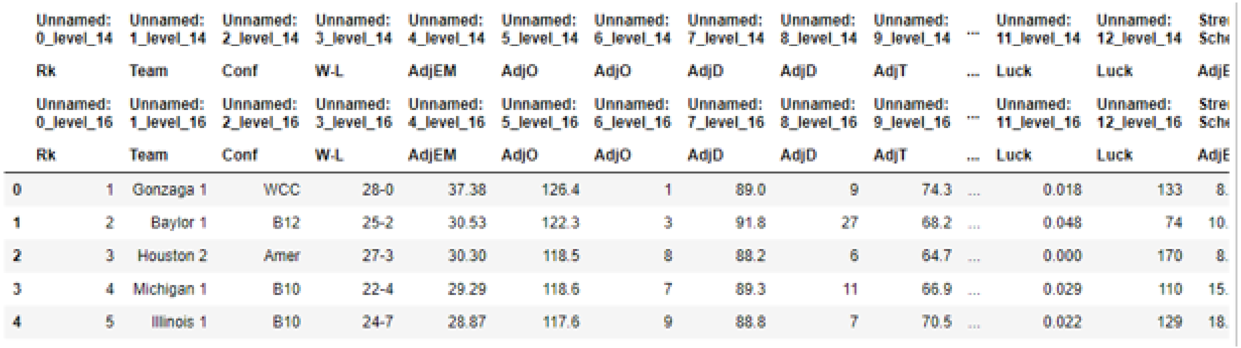
02·03·04·05·06·07·08·09·10·11·12·13·14·15·16·17·18·19·20·21

Data includes 1 of 6 games played on Saturday, March 27

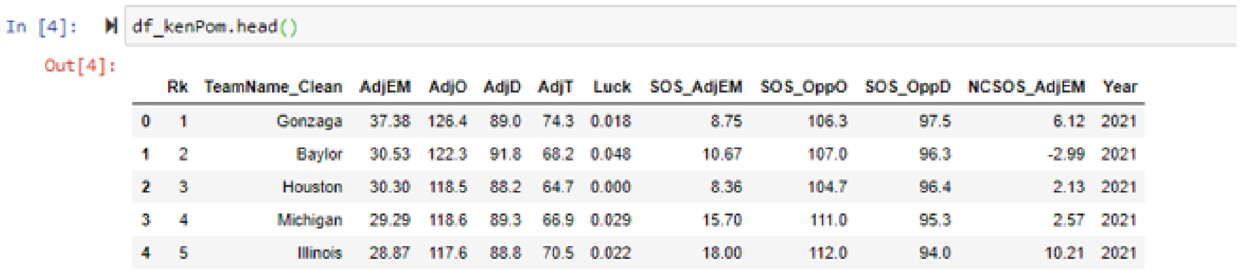
									Strength of Schedule			NCSOS
Rk	Team	Conf	W-L	AdjEM	AdjO	AdjD	AdjT	Luck	AdjEM	OppO	OppD	AdjEM
1	Gonzaga 1	WCC	28-0	+37.29	126.4 1	89.1 7	74.3 4	+.018 133	+8.67 94	106.2 95	97.6 96	+6.04 113
2	Baylor 1	B12	24-2	+29.76	122.8 3	93.0 37	68.5 157	+.047 76	+9.89 84	106.4 90	96.5 75	-3.04 260
3	Houston 2	Amer	26-3	+29.64	119.3 7	89.7 11	64.8 323	-.000 180	+7.82 101	104.3 144	96.4 71	+2.11 189
4	Michigan 1	B10	22-4	+29.30	118.6 8	89.3 9	66.9 246	+.029 112	+15.72 19	111.0 12	95.3 41	+2.61 178
5	Illinois 1	B10	24-7	+28.90	117.6 9	88.7 6	70.5 79	+.022 129	+18.04 5	112.0 5	94.0 11	+10.16 62
6	USC 6	P12	24-7	+26.92	114.9 15	88.0 5	67.1 236	-.008 201	+14.65 36	109.8 34	95.1 38	+6.14 111
7	Iowa 2	B10	22-9	+26.77	123.3 2	96.5 76	69.9 92	-.027 244	+16.31 16	110.1 27	93.8 8	-5.12 288
8	Alabama 2	SEC	26-6	+25.64	112.9 28	87.3 3	73.4 11	+.027 115	+14.49 40	109.6 41	95.1 37	+9.45 68
9	Loyola Chicago 8	MVC	26-4	+25.53	111.6 35	86.1 1	63.9 342	-.015 218	+6.12 115	105.3 113	99.1 125	+4.94 136
10	Colorado 5	P12	23-9	+24.40	115.7 13	91.3 24	66.2 277	-.025 237	+14.71 34	109.7 38	95.0 31	+3.35 164

The Kaggle datasets consisted of individual game, player, and team statistics going back to the 1984-85 season. Additionally, several other datasets such as “Team Spellings,” “Team IDs”, “Coaches”, etc… were provided to make compilation with other datasets easier.

To gather the KenPom data, we created a web-scraping script in Jupyter Notebook to quickly capture the HTML code provided on the KenPom website and make a dataframe out of the tables. The web-scraped data contained extra characters such as subscripts and superscripts for multiple columns. We cleaned this data to provide a condensed version of the columns we needed for analysis (AdjEM, AdjO, AdjD, AdjT, Luck, and Strength of Schedule metrics). The original KenPom dataframe as well as the clean KenPom dataframe used for analysis is shown below:



	Unnamed: 0_level_14	Unnamed: 1_level_14	Unnamed: 2_level_14	Unnamed: 3_level_14	Unnamed: 4_level_14	Unnamed: 5_level_14	Unnamed: 6_level_14	Unnamed: 7_level_14	Unnamed: 8_level_14	Unnamed: 9_level_14	...	Unnamed: 11_level_14	Unnamed: 12_level_14	Stre Sch
	Rk	Team	Conf	W-L	AdjEM	AdjO	AdjO	AdjD	AdjD	AdjT	...	Luck	Luck	AdjE
	Unnamed: 0_level_16	Unnamed: 1_level_16	Unnamed: 2_level_16	Unnamed: 3_level_16	Unnamed: 4_level_16	Unnamed: 5_level_16	Unnamed: 6_level_16	Unnamed: 7_level_16	Unnamed: 8_level_16	Unnamed: 9_level_16	...	Unnamed: 11_level_16	Unnamed: 12_level_16	Stre Sch
	Rk	Team	Conf	W-L	AdjEM	AdjO	AdjO	AdjD	AdjD	AdjT	...	Luck	Luck	AdjE
0	1	Gonzaga 1	WCC	28-0	37.38	126.4	1	89.0	9	74.3	...	0.018	133	8.
1	2	Baylor 1	B12	25-2	30.53	122.3	3	91.8	27	68.2	...	0.048	74	10.
2	3	Houston 2	Amer	27-3	30.30	118.5	8	88.2	6	64.7	...	0.000	170	8.
3	4	Michigan 1	B10	22-4	29.29	118.6	7	89.3	11	66.9	...	0.029	110	15.
4	5	Illinois 1	B10	24-7	28.87	117.6	9	88.8	7	70.5	...	0.022	129	18.



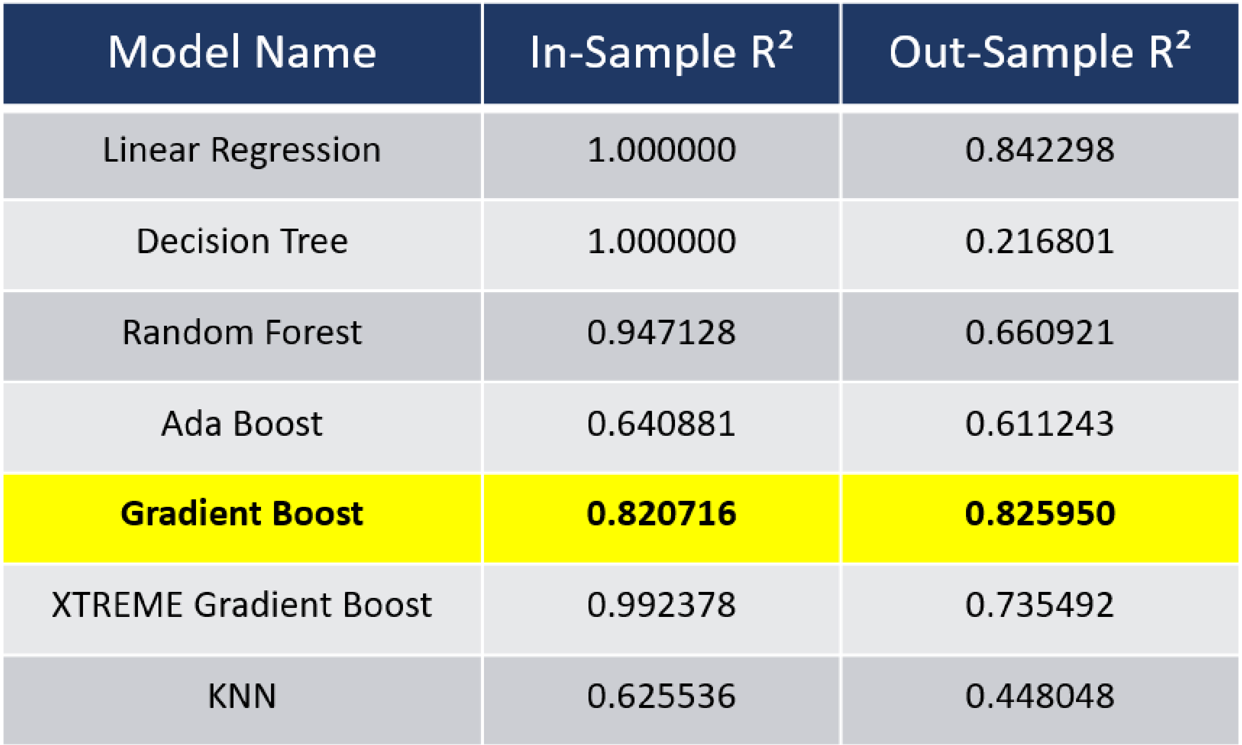
```
In [4]:  df_kenPom.head()
```

Out[4]:

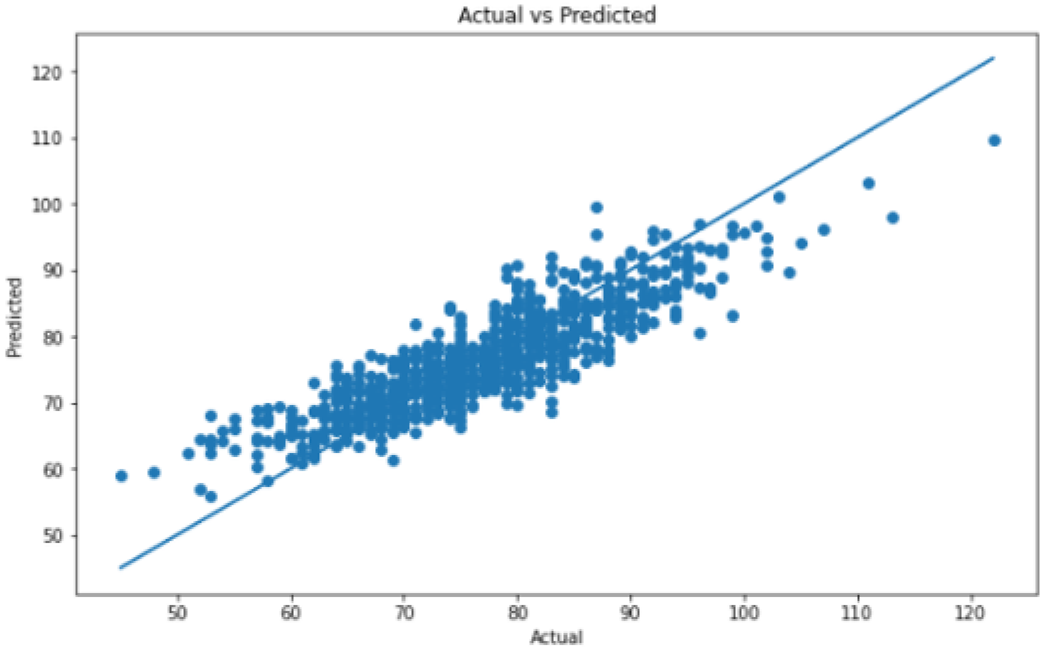
	Rk	TeamName_Clean	AdjEM	AdjO	AdjD	AdjT	Luck	SOS_AdjEM	SOS_OppO	SOS_OppD	NCSOS_AdjEM	Year
0	1	Gonzaga	37.38	126.4	89.0	74.3	0.018	8.75	106.3	97.5	6.12	2021
1	2	Baylor	30.53	122.3	91.8	68.2	0.048	10.67	107.0	96.3	-2.99	2021
2	3	Houston	30.30	118.5	88.2	64.7	0.000	8.36	104.7	96.4	2.13	2021
3	4	Michigan	29.29	118.6	89.3	66.9	0.029	15.70	111.0	95.3	2.57	2021
4	5	Illinois	28.87	117.6	88.8	70.5	0.022	18.00	112.0	94.0	10.21	2021

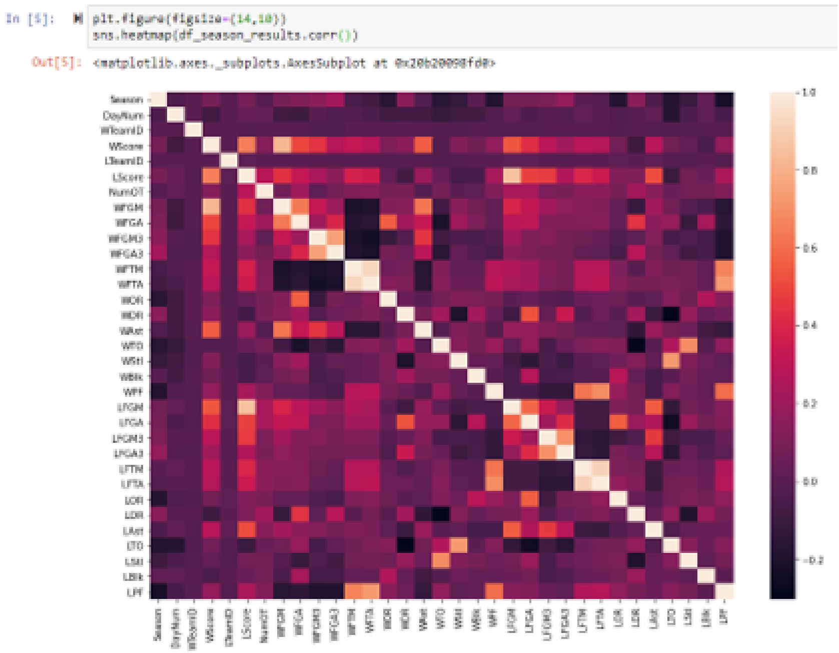
The KenPom data was then merged with our Kaggle data first on “Winning Team ID” and again on “Losing Team ID”. This gave us each winning and losing teams full season KenPom metrics.

After compiling KenPom metrics and game-by-game data into a single dataframe, further cleaning was necessary before training the model. All “made” baskets were removed from our dataset. This prevented our model from simply adding up total baskets made and predicting points scored off of season averages. We trained our model off our cleaned data to predict possible points scored. Our logic was the model would find “X” amount of points scored as well as “Y” amount of points allowed based on the opponents KenPom metrics. We trained our data on a total of 7 different possible models. The following table shows the R2 results off both the in-sample training data, as well as, the out-of-sample testing data:



Model Name	In-Sample R^2	Out-Sample R^2
Linear Regression	1.000000	0.842298
Decision Tree	1.000000	0.216801
Random Forest	0.947128	0.660921
Ada Boost	0.640881	0.611243
Gradient Boost	**0.820716**	**0.825950**
XTREME Gradient Boost	0.992378	0.735492
KNN	0.625536	0.448048





```
In [5]:  plt.figure(figsize=(14,10))
         sns.heatmap(df_season_results.corr())

Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x20b20098fd0>
```

We decided on the Gradient Boost model because the out of sample r2 was the best fit available. The linear regression model had the highest out-of-sample R2, but the in-sample training data appeared to be overfit as the r2 value was a perfect 1.000. When we had a trained model, we grouped our data game by game on “Team ID” and took an average of each team’s season statistics. This provided us with a single dataframe that contained each individual team’s KenPom metrics and their season averages for their respective “box score” statistics (Pts Scored, Pts Allowed, Assists, Rebounds, etc…) This process would allow us to run our trained model on two specific teams in order to predict a winner off the expected points scored by each team. To test the model’s effectiveness, it was run on the team matchups presented by the 2021 NCAA tournament bracket which were then simulated based on the models points predictions. Some results of the predictions are presented in the next section.

Results of Data Analysis

(i) Who will win the 2021 NCAA Tournament? How many points will they Score?

- Iowa Wins against Illinois: 80 - 76

(ii) Who would have won the 2020 NCAA Tournament had it not been canceled due to COVID 19?

- Michigan State Wins against BYU: 76 - 74

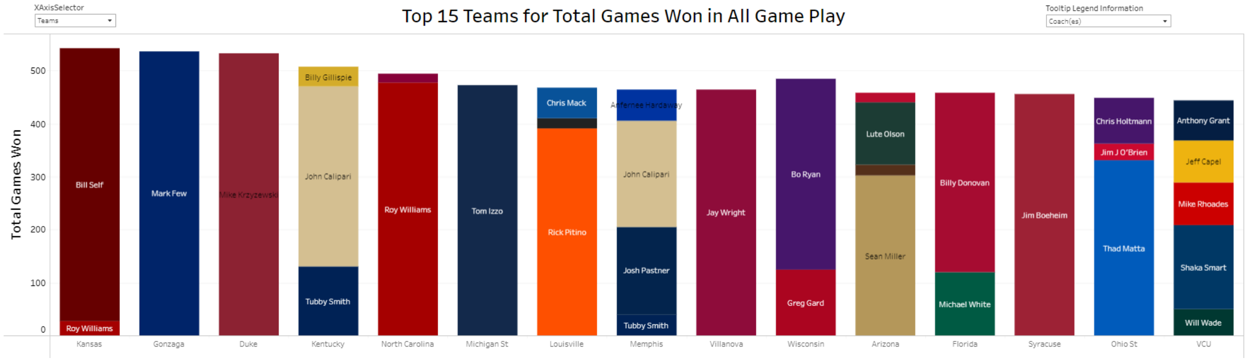
(iii) Model Results

- The model accurately predicted 23 out of 32 (72%) first round matchups correctly for the 2021 NCAA Tournament
- Since the model was finalized after the first and second round of the tournament had already been completed, we entered the future predictions into ESPN's 2nd chance tournament challenge. The 2nd chance tournament challenge includes the 16 remaining teams in the NCAA tournament and their matchups. As of 3/28, the model's "predicted" brackets sits in the 95th percentile and has correctly predicted all but 2 outcomes.

(iv) Exploring the Tableau visualizations

The main Tableau visualization is the Stats Dashboard. This visualization allows the user to manipulate season range, type of game, statistic viewed, the value of the statistic whether it be total or average, and the amount of top chosen values to view. From there the bar chart and line chart differ slightly.

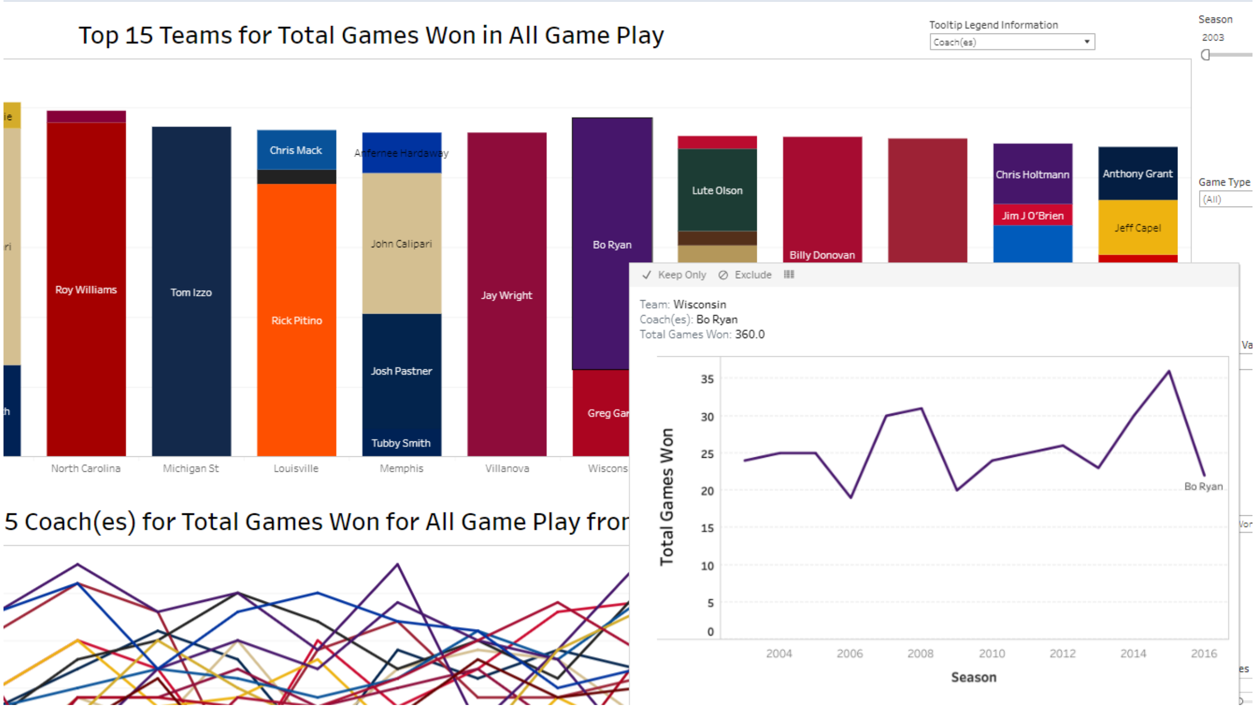
The bar chart has two additional drop downs to use: XAxisSelector and Tooltip Legend Information. The XAxisSelector allows the user to choose what value they want to see on the x axis, whether it be teams, coaches, or conferences. The Tooltip Legend Information drop down allows the user to choose what they will see in the tooltip as well as what coloring will be viewed in the visualization itself. An example could be the user selects teams as their x axis, and coaches as their tooltip legend information allowing them to see the teams as the main point of interest, but then broken out into sections of color identifying the coaches that contributed to that team.



XAxisSelector: Teams

Tooltip Legend Information: Coach(es)

Additionally, if the user hovered over a specific coach segment of a team's bar, the tooltip that pops up will display the coach's contribution over time for the chosen stat viewed.



Top 15 Teams for Total Games Won in All Game Play

Tooltip Legend Information: Coach(es)

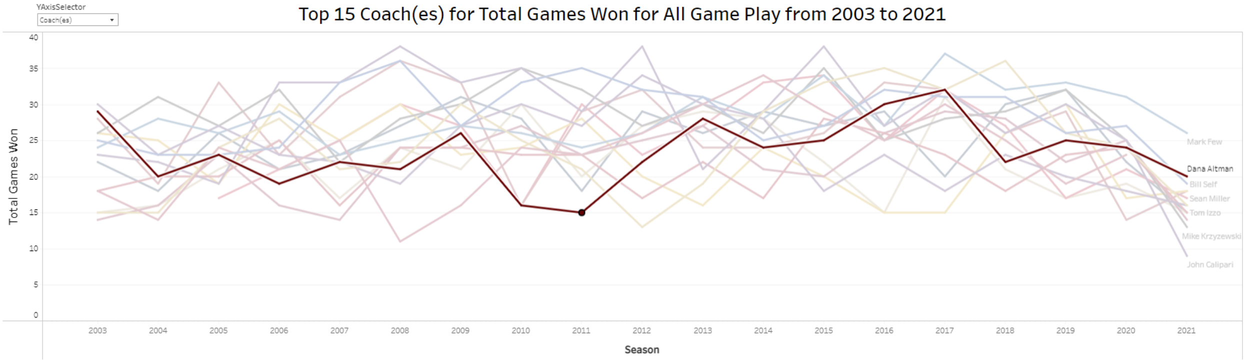
Season: 2003

Game Type: (All)

Team: Wisconsin
Coach(es): Bo Ryan
Total Games Won: 360.0

5 Coach(es) for Total Games Won for All Game Play fror

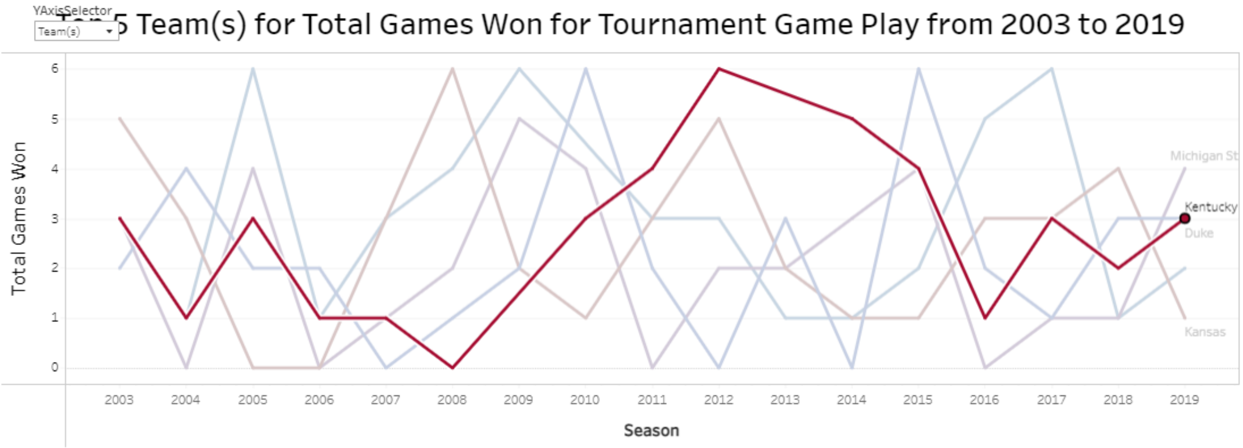
The line chart has one additional drop down to use, the YAxisSelector. Similar to the XAxisSelector from the bar chart visualization, the YAxisSelector lets the user select which value they want to see on the y axis, whether it be teams, coaches, or conferences.



YAxisSelector: Coach(es)

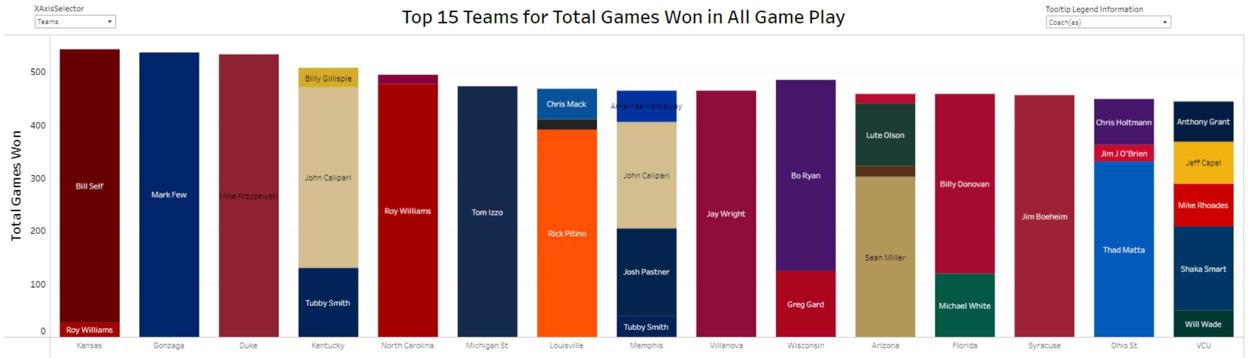
Top 15 Coach(es) for Total Games Won for All Game Play from 2003 to 2021

In exploring the data, we made some interesting findings! For example, did you know that while Kentucky won the 2012 NCAA tournament, they didn't even make the tournament in 2013! We found this by discovering a slight limitation to the line chart visualization. When filtered to tournament wins only by team, you'll see that Kentucky has a smooth down curve from 2012 to 2014 making it look like they had 5.5 tournament wins in 2013.



This would be impossible! Further research provided evidence showing that they didn't even make the tournament in 2013, but followed back up in 2014 by nearly winning it all once more.

We also noticed another slight limitation that led us to another discovery. The bar chart works exactly as it should, however you'll notice in the screenshot below, Wisconsin shows that it had 485 wins from 2003 to 2021, when in fact it only had 463. This is because during the 2016 season Wisconsin had 2 head coaches. Bo Ryan started out the season leading Wisconsin to a 7-5 record only to retire in December due to allegations of him having an affair. The job was given to Greg Gard mid season because of this. As a result, you'll see the 2016 season wins factored in twice, once for Bo Ryan and another for Greg Gard, overstating the total wins during the 2016 season for Wisconsin.



Limitations/Bias

The model uses known KenPom data to make predictions. While historical data is helpful, the model would be limited in predicting 2022 outcomes. It is possible to calculate KenPom data throughout the season, or to use a "rolling" number of games to determine a team's current KenPom statistics. However, given how much a college basketball team changes season-to-season, the model has been limited to use season ending KenPom statistics.

There are components that regress to the season averages for points scored. While that is expected, the model does not capture “blow out wins” or “blow out losses.” As in some other scoring prediction models, the “point-spread” of each game (amount of points the favored team is projected to win by based on probability) was not factored into our model.

The model may be biased to teams with a higher than average “Tempo.” i.e. fast-paced or high scoring teams seem to outperform actual outcomes when run through the model regardless of the caliber defense they play against. By back-testing the model against actual results in previous tournament years we found that high tempo teams were often predicted to score at or above their season average points against “stronger” teams. The actual result showed many of these games were 20+ point losses as opposed to close, high scoring games.

Future Work Recommendations

Currently, the scoring prediction model can be described as limited and complex. While it does an average job of predicting outcomes, there is plenty of room for improvement. Future work may include pulling in additional data from previous years (pre-2016) as those were not included in our initial analysis.

Creating a bracket that would auto-complete based on the winners of each individual matchup. Additionally, historically “seed” performance probability could be added to the model to refine predictions for first round matchups. For example, a #1 seed has lost to a #16 seed only 1 out of 144 times (0.69%) in the history of the NCAA tournament. Our model does not capture that data and views every game as “equal.”

Adding individual player information to the game-by-game information to train the model to capture the “importance” of individuals on a specific team. This might give some insight to what people can expect if a “star player” gets injured before a future matchup.

Lastly, the ideal goal for future work would be to further refine features in order to reduce model complexity and increase explainability.

Works Cited

2021 Pomeroy College Basketball Rankings "2021 Pomeroy College Basketball Ratings". *Kenpom.Com*, 2021, https://kenpom.com/. Accessed 21 Mar 2021.

March Machine Learning Mania 2021 – NCAAM | Kaggle "March Machine Learning Mania 2021 - NCAAM | Kaggle". *Kaggle.Com*, 2021, https://www.kaggle.com/c/ncaam-march-mania-2021/data. Accessed 28 Mar 2021.