Leading the March to Madness

PREDICTING THE 2018 NCAA MEN'S BASKETBALL TOURNAMENT

Edrian "Ed" Salinas | General Assembly Data Science Immersive | April 10, 2018

Contents

SUMMARY

For my Capstone Project in the Data Science Immersive at General Assembly, I chose to look into predicting the holiest of holies of late Winter Collegiate Sports, the 2018 National Collegiate Athletic Association (NCAA) Men's Basketball Tournament. This is an annual contest that pits the best of the best in Division I College Basketball with single elimination games. Navigating through the three week long tournament in March and early April takes a great deal of skill, matchup, and luck.

This sporting event has also sprouted an annual contest on Kaggle.com. This year's contest was sponsored by Google Cloud with a final takeaway of $50,000 total to be given to the top three submitted predictions scored by lowest log loss.

Kaggle provides the data which needs to be explored and cleaned up to create our features for the final model. I used a combination of basketball analytics taken from the National Basketball Association (NBA), the Elo Rating system, and Ken Pomeroy's proprietary college basketball statistics to create the features of the model.

I performed several models on the final features which resulted in a variety of results and submitted them to Kaggle. What I found truly surprised me.

BACKGROUND

The NCAA Men's Basketball Tournament has been going on for 80 years since the very first tournament was held in March 1939 when there were only 8 teams playing in single game elimination. Now there are 68 teams in the tournament which is played over three weekends from mid-March through the first weekend in April. The tournament has become a multimillion dollar industry and is often lovingly called March Madness.

The tournament begins with the NCAA Selection Committee. They meet on a Sunday in mid-March called Selection Sunday. Of the 68 teams that are selected by the committee to be in the tournament, 32 are given automatic bids. These teams win their conference tournaments which are usually held the weekend of Selection Sunday with some games going right into the afternoon that the Selection Committee is meeting. The rest of the teams are selected by the committee and are given at-large bids. The committee takes into account final rankings like the Ratings Percentage Index (RPI) published weekly, Strength of Schedule (SOS), meaningful wins against teams with a high SOS or RPI, and of course, meaningful losses versus the same.

When they have completed the selection process, they publish a bracket broken up into four regions: West, Midwest, South, and East. Each region has 16 teams each ranked from 1 to 16 with 1 being the top teams in the country going down from there. Within each region the #1 seed plays the #16 seed, the #2 seed plays the #15 seed, etc. Four more teams are selected and are pitted against four other teams that play the Tuesday and Wednesday night after Selection Sunday. These are considered play-in games to get into the tournament. Two of the games feature four 16-seeded teams and the other two feature four 11-seeded teams. Thus begins the tournament!

DATA

The primary portion of the data is provided by Kaggle and is broken up in the following files:

- Cities - List of cities and states that have Division I teams
- Conferences - List of all the Division I conferences across the country since 1985.
- ConferenceTourneyGames - Conference Tournament Games (ACC Tournament/SEC Tournament, etc.)
- Events 2010-2017 - Play by play for all games from 2010 to 2017
- GameCities - Cities where games have been played since 2010
- MasseyOrdinals - Weekly team rankings from various sources such as Pomeroy, Sagarin, ESPN, RPI, etc.
- NCAA Tourney Compact and Detailed Results - The primary Test set for the model
- NCAA Tourney Seed Round Slots - represents bracket structure of the Tournament
- NCAA Tourney Seeds - Seeds for all teams in every tournament since 1985
- NCAA Tourney Slots - Mechanism by which teams are paired against each other
- Players 2010-2017 - Data on all the players that have played in Division I college basketball games
- Regular Season Compact and Detailed Results - Stats on every game played in the regular season
- Seasons - Identifies the different college basketball seasons
- Secondary Tourney Compact Results - Results from NIT or other post season tournaments
- Secondary Tourney Teams - Teams that have played in these secondary tournaments
- Team Coaches - All coaches and the teams they've coached
- Team Conferences - All the conferences in Division I
- Teams - List of teams
- Team Spellings - Long form spellings of each Team
- KenPom – Not provided by Kaggle; Ken Pomeroy's college basketball statistics on the front page of his site (Kaggle requires any additional data to be free to all)

EXPLORATORY DATA ANALYSIS

To begin the data analysis, I took a look at the Regular Season Detail Results data. This information includes every regular season game played by year, the day that the game was played, the winning team, the losing team, the score for each team, and several in-game statistics that pertain to basketball such as field goals made/attempted, three point field goals, free throws, offensive and defensive rebounds, assists, turnovers, steals, blocks, and fouls. To help break down these statistics to create features for the final model, I used NBA Analytics from the NBA Stuffer Website and their Team Evaluation Statistics. While the NBA and College games are different, the analytics stats should help make for a good predictive model.

For Possessions, I used their basic formula using the statistics that are found in the Season dataset:

$$Poss = 0.96[(FG_{Att}) + TO + 0.44(FT_{Att}) - OR]$$

Where:

$$\text{FG}_{Att} = \text{ Field Goal Attempts}$$

$$\text{TO} = \text{Turnovers}$$

$$\text{FT}_{Att} = \text{ Free Throw Attempts Throw Attempts}$$

The next set of metrics form what is called the Four Factors: Effective Field Goal Rate, Turnover Rate, Offensive Rebounding Percentage, and Free Throw Rate. These are represented by the following equations respectively:

$$FG_{Eff} = \frac{FG_{Made} + 0.5(3PtFG_{Made})}{FG_{Att}}$$

$$TO_{Rate} = \frac{TO}{FG_{Made} + 0.44(FT_{Att}) + TO}$$

$$OR_{Per} = \frac{OR}{OR + DR_{Opp}}$$

$$FT_{Rate} = \frac{FT_{Att}}{FG_{Att}}$$

Where:

$$\text{FG}_{Made} = \text{ Field Goals Made}$$

$$\text{3Pt FG}_{Made} = \text{ Three Point Field Goal Made}$$

$$\text{FG}_{Att} = \text{ Field Goal Attempts}$$

$$TO = Turnovers$$

$$OR = Offensive\ Rebounds$$

$$FT_{Att} = Free\ Throw\ Attempts$$

$$DR_{Opp} = Opponent\ Defensive\ Rebounds$$

“While these are the four essential factors that help decides wins and to losses, none of these factors carry equal weight. Dean Oliver has found the following weights as follows:”[1]

1. Shooting (40%)
2. Turnovers (25%)
3. Rebounding (20%)
4. Free Throws (15%)

For our calculations the equation for the Four Factor metric is:

$$FF = 0.4\left(FG_{Eff}\right) + 0.25(TO_{Rate}) + 0.20(OR_{Per}) + 0.15(FT_{Rate})$$

Another metric used is the Player Impact Estimate (PIE) from Rusty LaRue. “PIE measures a player’s overall statistical contribution against the total statistics in games they play in. Basically it’s giving you a percentage showing how much of a positive or negative impact a player had on a game.”[2] For the purpose of this model, PIE is represented by the following equation taking into account the entire team:

$$PIE = \frac{Team[Points + FG_{Made} + FT_{Made} - FG_{Att} - FT_{Att} + DR + 0.5(OR) + Ast + Stl + 0.5(Blk) - PF - TO]}{Total[Points + FG_{Made} + FT_{Made} - FG_{Att} - FT_{Att} + DR + 0.5(OR) + Ast + Stl + 0.5(Blk) - PF - TO]}$$

Other statistics that we can use include Offensive Efficiency (number of points scored per 100 possessions), Defensive Efficiency (number of points a team allows per 100 opposing team possessions), Assist Ratio (percentage of a team’s possessions that ends in an assist), Defensive Rebounding Percentage (team’s ability to get defensive rebounds), and Free Throw Percentage:

$$Off_{Eff} = 100(\frac{Score}{Poss})$$

$$Def_{Eff} = 100\left(\frac{Score}{Poss}\right) or\ just\ the\ opposite\ of\ Off_{Eff}$$

[1] "NBA Analytics 101 Primer | NBAstuffer." https://www.nbastuffer.com/analytics-101/. Accessed 6 Apr. 2018

[2] "Player Efficiency Stats - More than 94': A basketball blog - Rusty LaRue." 17 May. 2014, http://www.rustylarue.com/more-than-94/player-efficiency-stats. Accessed 6 Apr. 2018

$$Ast_{Ratio} = \frac{100(Ast)}{FG_{Att} + 0.44(FT_{Att}) + Ast + TO}$$

$$DR_{Per} = \frac{DR}{DR + OR}$$

$$FT_{Per} = \frac{FT_{Made}}{FT_{Att}}$$

The next statistic we use is the Elo Rating. The Elo Rating system was originally invented as an improved chess rating system, but is also used as a rating system for multiplayer competition including basketball which is how we will use it.

The Elo Rating is calculated per game from the following for Team A(R_A) and Team B (R_B) for each game played between each other:

$$E_A = \frac{1}{1 + 10^{\frac{(R_B - R_A)}{400}}}$$

$$E_B = \frac{1}{1 + 10^{\frac{(R_A - R_B)}{400}}}$$

Nate Silver has calculated his own algorithm for the Elo rating that we will be using as a metric in our features. These can be calculated with the following equations:[3]

$$R_0 = 1300$$

$$R_{i+1} = K(S_{Team} - E_{Team})R_i$$

Where R is the Elo Rating, S=1 if the team wins and 0 if the team loses. E represents the expected win probability in Nate's formula and is defined by:

$$E_{Team} = \frac{1}{1 + 10^{\frac{(Opp\ Elo - Team\ Elo)}{400}}}$$

In chess, K is a constant, but Nate changes K to handle margin of victory and is defined by:

$$K = 20\frac{(MOV_{Winner} + 3)^{0.8}}{7.5 + 0.006(elo\_difference_{winner})}$$

Where:

$$elo\_difference_{winner} = winning\ elo - losing\ elo$$

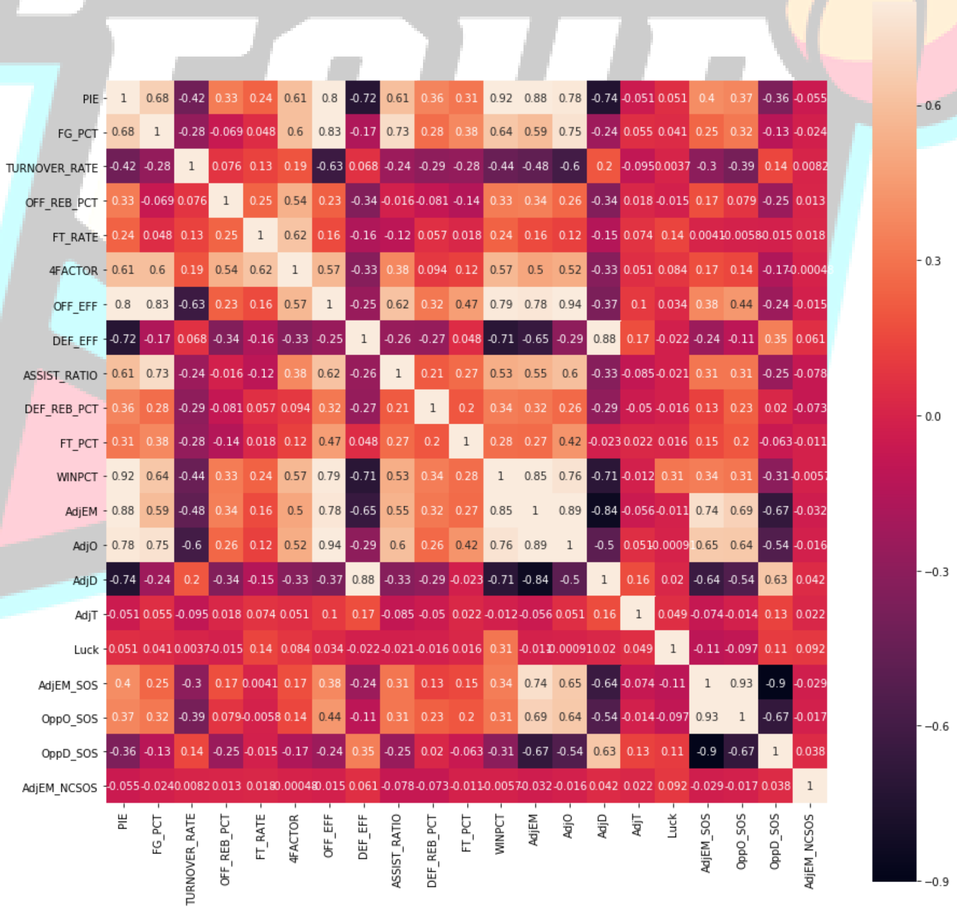
[3] "Replicating Nate Silver's Elo Algorithm." https://www.ergosum.co/nate-silvers-nba-elo-algorithm/. Accessed 6 Apr. 2018

Nate also takes into account home advantage by increasing the rate of the home team by 100 as in $R_{Home} = R_{Team} + 100$. In between seasons Nate handles this by reverting each team towards a mean of 1505 as in the following formula

$$R_{S=i+1} = 0.75R_{s=i} + (0.25)1505$$

The rest of the features that were used in our machine learning were scraped from Ken Pomeroy's website. He uses his own proprietary algorithms to rank all 350+ Division I Basketball programs including:

- Adjusted Efficiency Margin
- Adjusted Offensive Efficiency
- Adjusted Defensive Efficiency
- Adjust Tempo (Possessions per 40 minutes)
- Luck Rating
- Strength of Schedule Ratings for conference and non-conference play
- Adjusted Offensive Efficiency of Opposing Offenses
- Adjust Defensive Efficiency of Opposing Defenses
- Are we ready to model yet? Almost. First let's look at a correlation matrix of our features so far.



	PIE	FG_PCT	TURNOVER_RATE	OFF_REB_PCT	FT_RATE	4FACTOR	OFF_EFF	DEF_EFF	ASSIST_RATIO	DEF_REB_PCT	FT_PCT	WINPCT	AdjEM	AdjO	AdjD	AdjT	Luck	AdjEM_SOS	OppO_SOS	OppD_SOS	AdjEM_NCSOS
PIE	1	0.68	-0.42	0.33	0.24	0.61	0.8	-0.72	0.61	0.36	0.31	0.92	0.88	0.78	-0.74	-0.051	0.051	0.4	0.37	-0.36	-0.055
FG_PCT	0.68	1	-0.28	-0.069	0.048	0.6	0.83	-0.17	0.73	0.28	0.38	0.64	0.59	0.75	-0.24	0.055	0.041	0.25	0.32	-0.13	-0.024
TURNOVER_RATE	-0.42	-0.28	1	0.076	0.13	0.19	-0.63	0.068	-0.24	-0.29	-0.28	-0.44	-0.48	-0.6	0.2	-0.095	0.0037	-0.3	-0.39	0.14	0.0082
OFF_REB_PCT	0.33	-0.069	0.076	1	0.25	0.54	0.23	-0.34	-0.016	-0.081	-0.14	0.33	0.34	0.26	-0.34	0.018	-0.015	0.17	0.079	-0.25	0.013
FT_RATE	0.24	0.048	0.13	0.25	1	0.62	0.16	-0.16	-0.12	0.057	0.018	0.24	0.16	0.12	-0.15	0.074	0.14	0.0041	-0.0058	-0.015	0.018
4FACTOR	0.61	0.6	0.19	0.54	0.62	1	0.57	-0.33	0.38	0.094	0.12	0.57	0.5	0.52	-0.33	0.051	0.084	0.17	0.14	-0.17	-0.00048
OFF_EFF	0.8	0.83	-0.63	0.23	0.16	0.57	1	-0.25	0.62	0.32	0.47	0.79	0.78	0.94	-0.37	0.1	0.034	0.38	0.44	-0.24	-0.015
DEF_EFF	-0.72	-0.17	0.068	-0.34	-0.16	-0.33	-0.25	1	-0.26	-0.27	0.048	-0.71	-0.65	-0.29	0.88	0.17	-0.022	-0.24	-0.11	0.35	0.061
ASSIST_RATIO	0.61	0.73	-0.24	-0.016	-0.12	0.38	0.62	-0.26	1	0.21	0.27	0.53	0.55	0.6	-0.33	-0.085	-0.021	0.31	0.31	-0.25	-0.078
DEF_REB_PCT	0.36	0.28	-0.29	-0.081	0.057	0.094	0.32	-0.27	0.21	1	0.2	0.34	0.32	0.26	-0.29	-0.05	-0.016	0.13	0.23	0.02	-0.073
FT_PCT	0.31	0.38	-0.28	-0.14	0.018	0.12	0.47	0.048	0.27	0.2	1	0.28	0.27	0.42	-0.023	0.022	0.016	0.15	0.2	-0.063	-0.011
WINPCT	0.92	0.64	-0.44	0.33	0.24	0.57	0.79	-0.71	0.53	0.34	0.28	1	0.85	0.76	-0.71	-0.012	0.31	0.34	0.31	-0.31	-0.0057
AdjEM	0.88	0.59	-0.48	0.34	0.16	0.5	0.78	-0.65	0.55	0.32	0.27	0.85	1	0.89	-0.84	-0.056	-0.011	0.74	0.69	-0.67	-0.032
AdjO	0.78	0.75	-0.6	0.26	0.12	0.52	0.94	-0.29	0.6	0.26	0.42	0.76	0.89	1	-0.5	0.051	-0.00091	0.65	0.64	-0.54	-0.016
AdjD	-0.74	-0.24	0.2	-0.34	-0.15	-0.33	-0.37	0.88	-0.33	-0.29	-0.023	-0.71	-0.84	-0.5	1	0.16	0.02	-0.64	-0.54	0.63	0.042
AdjT	-0.051	0.055	-0.095	0.018	0.074	0.051	0.1	0.17	-0.085	-0.05	0.022	-0.012	-0.056	0.051	0.16	1	0.049	-0.074	-0.014	0.13	0.022
Luck	0.051	0.041	0.0037	-0.015	0.14	0.084	0.034	-0.022	-0.021	-0.016	0.016	0.31	-0.011	-0.00091	0.02	0.049	1	-0.11	-0.097	0.11	0.092
AdjEM_SOS	0.4	0.25	-0.3	0.17	0.0041	0.17	0.38	-0.24	0.31	0.13	0.15	0.34	0.74	0.65	-0.64	-0.074	-0.11	1	0.93	-0.9	-0.029
OppO_SOS	0.37	0.32	-0.39	0.079	-0.0058	0.14	0.44	-0.11	0.31	0.23	0.2	0.31	0.69	0.64	-0.54	-0.014	-0.097	0.93	1	-0.67	-0.017
OppD_SOS	-0.36	-0.13	0.14	-0.25	-0.015	-0.17	-0.24	0.35	-0.25	0.02	-0.063	-0.31	-0.67	-0.54	0.63	0.13	0.11	-0.9	-0.67	1	0.038
AdjEM_NCSOS	-0.055	-0.024	0.0082	0.013	0.018	-0.00048	-0.015	0.061	-0.078	-0.073	-0.011	-0.0057	-0.032	-0.016	0.042	0.022	0.092	-0.029	-0.017	0.038	1

Figure 1: Correlation Heat map of final features

Here are the main takeaways from Figure 1:

- PIE and Winning Percentage are heavily correlated
- NBA Stuffer's Offensive Efficiency and Ken Pomeroy's Adjusted Offensive Efficiency are correlated which makes intuitive sense.
- Strenth of Schedule and Opponent Offensive Efficiency are Correlated
- Strength of Schedule and Opponent Defensive Efficiency are negatively correlated

NOW it's time to model.

MODELING

Predictive modeling for the NCAA Tournament is relatively easy in the sense that it is a classic binary classification problem. A binary classification problem is one in which there are only two outcomes that you are trying to predict: a win or a loss. There are several models that could be used for these types of problems so with the small data set (season data for 13 years for over 350+ teams).

We need to do a quick aside on the scoring for Kaggle. It's based on the Log Loss Function defined by the following equation:

$$-\frac{1}{N}\sum_{i=1}^{N}[y_i logp_i + (1-y_i)\log(1-p_i)]$$

For March Madness there are 63 games so the final equation becomes:

$$-\frac{1}{63}\sum_{i=1}^{63}[y_i logp_i + (1-y_i)\log(1-p_i)]$$

where y_i is the binary indicator and p_i is the model probability of assigning label of win(1) or loss(0) to each game. A perfect model would have a log loss of zero while the opposite would be true for less ideal models that have much larger values of log loss.

The initial model that was put together for the competition was done rather quickly and only included Ken Pomeroy statistics from 2014-2017 and did not include the Elo Rating. The Gridsearch parameters for the model were also limited due to time. The model that was chosen was Logistic Regression with C=1 and penalty of l2. The best log loss was 0.44 with training accuracy of 81.78% and 77.78%.

The final submission score for this model was 0.698145 with a ranking of 736. Obviously that was not very good so I was definitely taken out of the final prize. However, with a little more time, more data from Ken Pomeroy's website and the Elo Rating I was able to fine tune some models and work to see how I could have gotten a better score with this additional data and more hyperparameter turning.

I ended up doing very extensive Gridsearches of several different type of classifier models. This included K-Nearest Neighbors, Decision Trees, Random Forest, Gradient Boosting, XGBoost, Logistic Regression, and Single Vector Machines (SVM) which also happens to be the order from worst to best as shown in Table 1.

Models	Test Log Loss	Best Log Loss	Training Accuracy	Test Accuracy	Auc Roc Score
Logistic Regression	0.34	0.4	84.38%	85.58%	0.86
SVM	0.35	0.4	84.01%	85.10%	0.85
XGBoost	0.36	0.41	82.93%	84.13%	0.84
Original	0.44	0.44	81.78%	77.78%	-
Random Forest	0.53	0.51	79.93%	74.04%	0.74
Gradient Boost	0.48	0.52	87.02%	76.92%	0.77
Decision Trees	0.76	0.54	69.95%	71.15%	0.71
KNN	0.69	0.6	100%	73.08%	0.73

Table 1: Evaluation Metrics for different models

As can be seen Logistic Regression and SVM had the same "best" log loss of all the models that were being evaluated. It's actually amazing how Logistic Regression most of the time, in my albeit limited experience, seems to yield the best results. However, SVM also had really good results. So which of the two do we choose?

The AUC ROC store for Logistic Regression was slightly higher than SVM at 0.86 versus 0.85. So what is the AUC ROC score?

The AUC ROC score is actually better represented by the Area Under the Receiver Operating Characteristic curve (AUROC)[4]. To understand the AUROC curve, the concept of the confusion matrix must be understood. When we make a binary prediction like this problem predicting wins and losses, there can be four types of incomes:

- We predict 1 while we should have the class is actually 1. This is called a True Positive (TP) where the model correctly predicts that the class is positive (1).
- We predict 1 while we should have the class is actually 0. This is called a False Positive (FP) where the model incorrectly predict that the class is positive (1)
- We predict 0 while we should have the class is actually 1. This is called a False Negative (FN) where the model incorrectly predict that the class is negative (0).

[4] "classification - What does AUC stand for and what is it? - Cross ...." https://stats.stackexchange.com/questions/132777/what-does-auc-stand-for-and-what-is-it. Accessed 6 Apr. 2018

- We predict 0 while we should have the class is actually 0. This is called a True Negative (TN) where the model correctly predicts that the class is negative (0)

The confusion matrix for the Logistic Regression Model is shown in Table 2:

		Predicted Class	
		Win	**Loss**
Actual Class	**Win**	84 True Positives (TP)	20 False Negatives (FN)
	Loss	10 False Positives (FP)	94 True Negatives (TN)

Table 2: Confusion Matrix for Logistic Regression Model

This was evaluated against the test set for the model which was 20% of the total dataset. We can see that, among 84 games, the Logistic Regression model accurately picked who would win those games. On the other end, among 94 games, the model accurately picked who would lose those games. However, it also predicts losses in 20 games that should have been wins for those teams and predicted wins in 20 games that were actually losses.

We can compute two different metrics from the confusion matrix which will factor into the AUROC curve.

- True Positive Rate (TPR) also known as sensitivity defined as $\frac{TP}{TP+FN}$. This metric corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points. The higher the TPR, the fewer positive data points that will be missed.
- False Positive Rate (FPR) is defined as $\frac{FP}{FP+TN}$. This metric corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points. The higher the FPR, the more negative points will be misclassified.

To find the AUROC curve we plot TPR vs. FPR, and resulting curve is called the ROC curve. For the Logistic Regression Model, the AUROC curve is seen in Fig. 2.

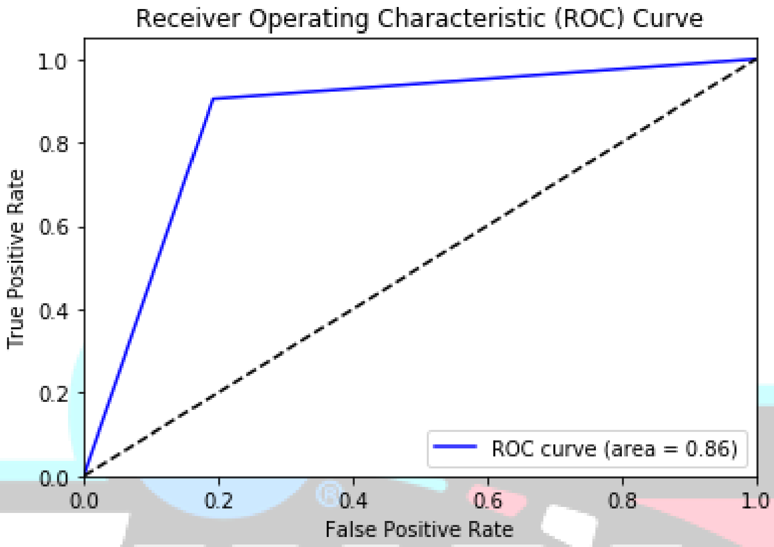


Figure 2: AUROC curve for Logistic Regression Model

The confusion matrix for SVM is shown in Table 3:

		Predicted Class	
		Win	Loss
Actual Class	Win	85 True Positives (TP)	19 False Negatives (FN)
	Loss	12 False Positives (FP)	92 True Negatives (TN)

Table 3: Confusion Matrix for SVM Model

And the AUROC curve in Fig. 3:

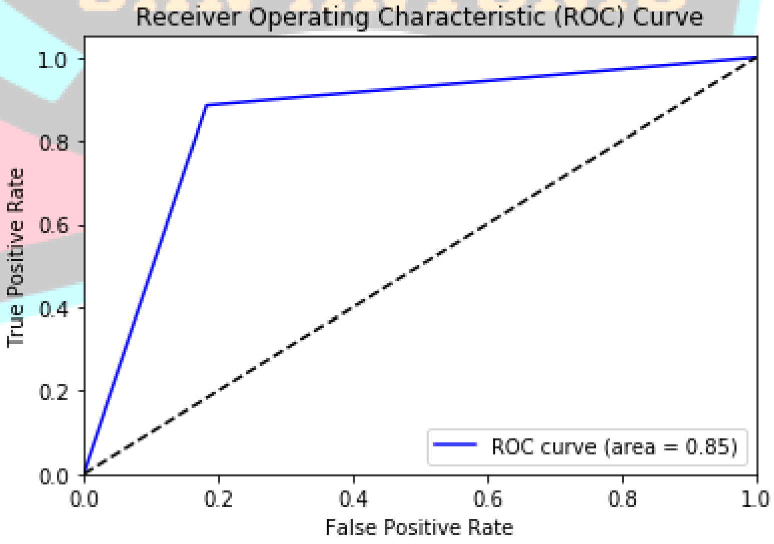


Figure 3: AUROC curve for SVM Model

The AUROC for the Logistic Regression Model was 0.86 while the AUROC for the SVM Model was 0.85. In this case, the Logistic Regression Model wins out as the better model with the following parameters based on the test results: C=1000, fit_intercept = False, and penalty = l1.

Models	AUROC Score
Logistic Regression	0.86
SVM	0.85
XGBoost	0.84
Gradient Boost	0.77
Random Forest	0.74
KNN	0.73
Decision Trees	0.71
Original	-

Table 4: AUROC scores for the various models

Table 4 shows the AUROC score for all the models evaluated from best to worst.

CONCLUSION

All but the original models were done as the NCAA Tournament was underway, but none of the actual wins and losses were used to evaluate the models during the time they were evaluated. The dataset remained the same with data through the 2018 Regular Season. As mentioned earlier, after the Final Four was completed on April 2nd and Villanova beat Michigan convincingly, I was able to go back and get the Kaggle scores (Log Loss) based on the actual wins and losses over the course of the 2018 Tournament. The final Kaggle scores can be found in Table 5.

Models	Kaggle Score
SVM	0.400627
Logistic Regression	0.415799
XGBoost	0.435129
Gradient Boost	0.453843
Top Score	0.531942
Random Forest	0.556503
Decision Trees	0.569473
KNN	0.697064
Original	0.698145

Table 5: Kaggle Scores (Log Loss) for the actual 2018 NCAA Men's Tournament Results

These were the results that I found incredibly surprising. The top score for the tournament was 0.531942. Had I submitted any one of my top four models, I would have beaten the best score by quite a bit. Again, there was no leakage in the model that I could tell, since the only data used was final regular season data for the last 13 years. No actual data from the tournament games as they were played was used. I basically should have held out and done a better analysis of the games before the tournament started and submitted my best model.

In any case, this was an amazing and fun thought experiment on predicting something that I truly love to watch over the course of March and early April. I certainly plan on entering this tournament next year and beyond!

Bring on March Madness 2019!!

References

1. A&M Men’s Basketball, November 17, 2017, accessed April 5 , 2018, http://12thman.com/news/2017/11/17/mens-basketball-hogg-helps-aggies-down-ucsb-in-home-opener.aspx
2. 2018 NCAA Final Four Logo, accessed April 5, 2018, https://www.primesport.com/d/ncaa-mens-final-four
3. "Google Cloud & NCAA® ML Competition 2018-Men's | Kaggle." https://www.kaggle.com/c/mens-machine-learning-competition-2018. Accessed 6 Apr. 2018
4. "classification - What does AUC stand for and what is it? - Cross ...." https://stats.stackexchange.com/questions/132777/what-does-auc-stand-for-and-what-is-it. Accessed 6 Apr. 2018.
5. "Making Sense of Logarithmic Loss." 14 Dec. 2015, http://www.exegetic.biz/blog/2015/12/making-sense-logarithmic-loss/. Accessed 6 Apr. 2018
6. "NBA Analytics 101 Primer | NBAstuffer." https://www.nbastuffer.com/analytics-101/. Accessed 6 Apr. 2018.
7. “A&M Men’s Basketball, November 17, 2017”. Accessed April 5 , 2018, http://12thman.com/news/2017/11/17/mens-basketball-hogg-helps-aggies-down-ucsb-in-home-opener.aspx
8. "Pomeroy." https://kenpom.com/. Accessed 6 Apr. 2018
9. “FiveThirtyEight-Elo-Ratings.” https://www.kaggle.com/lpkirwin/fivethirtyeight-elo-ratings. Accessed 6 Apr. 2018.
10. “Elo Rating System.” https://en.wikipedia.org/wiki/Elo_rating_system. Accessed 6 Apr. 2018.

11. “Replicating Nate Silver’s Elo Algorithm.” https://www.ergosum.co/nate-silvers-nba-elo-algorithm/. Accessed 6 Apr. 2018.