Predicting March Madness

Using Ridge & LASSO Regression to Predict Points Scored

Raymond Romaniuk (6047088)

Brock University

MATH 4F90

Due: April 25, 2021

Introduction

The National Basketball Association (NBA) is regarded as one of the four major sports leagues in North America, alongside the National Football League (NFL), National Hockey League (NHL) and Major League Baseball (MLB). Basketball is a team sport played between two teams with five players from each team on the playing surface, known as the court, at any given time. The objective of the game is to shoot the basketball through the opposing teams hoop and accumulate more points than your opponent. Points can be scored in two separate categories, field goals and free throws. A field goal is scored during live play and can be worth either two or three points, called a two or three pointer. The amount of points a field goal is worth is dictated by the position on the court the shooting player shot the ball from. If it was from behind the three-point line (see Figure 1) the shot is worth three points and if it is from inside the three-point line the shot is only worth two points. The second method of scoring is by free throws, these are worth one point each. A free throw occurs when a player commits an infraction, known as a foul, on the opposition and their opponent is awarded a given number of free throws based on the severity of the foul. Free throws take place from the free throw line with no infringement allowed by the opposition. The NBA being a significant focal point in North American sports leads fans to gravitate not just to the NBA, but also to National Collegiate Athletic Association Division I Basketball, NCAA Basketball or College Basketball for short, where the majority of future NBA players come from.

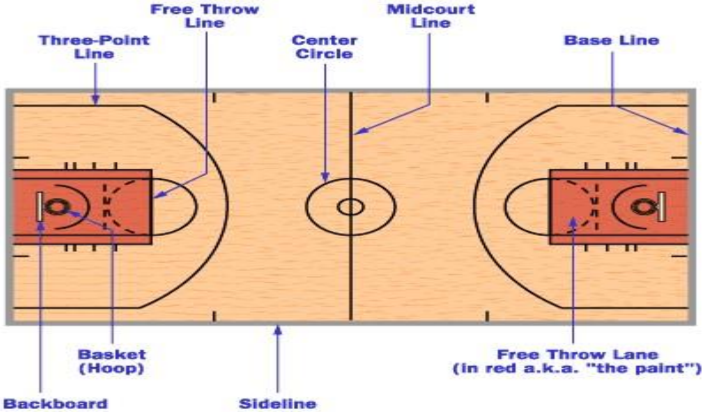


Figure 1: Diagram of important basketball court locations

This project aims to explore College Basketball, specifically the NCAA Division I Men's Basketball Tournament, known as March Madness. The March Madness Tournament is the culmination of each College Basketball season and brings 68 of the top teams in the country together to form a bracketed tournament and compete for the distinction of being the best team in Division I College Basketball. Teams gain entry to the tournament through two avenues, automatic bids and at large bids. Of the 68 teams, 32 of them receive automatic bids into the tournament. To receive an automatic bid a team must win their respective conference's playoff tournament. There are 32 conferences, so these 32 teams are the 32 conference champions. The 36 at large bids are comprised of teams who did not win their conference's playoff tournament. These 36 teams are chosen by the Selection Committee who selects the 36 teams they believe are most deserving of competing in the tournament. Teams are then seeded from 1 to 68 and split into four different regions, East, West, South and Midwest. There are 16 teams allocated to each region and seeded from 1 to 16 within the region. The teams are then bracketed by their seed and the opening round consists of games with the first seed playing the sixteenth seed, second playing fifteenth and so on. Noticeably 16 teams per region does not divide the total 68 teams evenly. An eight-team play-in round, called the First Four, is contested prior to the First Round between the four lowest ranked automatic bid teams and the four lowest ranked at large bid teams. The lowest ranked automatic bid teams are usually seeded lower, overall, than the lowest ranked at large bid teams, since they are representing weaker conferences. The four automatic bid teams thus play each other for one of two available 16 seed spots, whereas the four at large bid teams play each other for one of two available 11 seed spots. March Madness consists of six rounds and a total of 67 games.

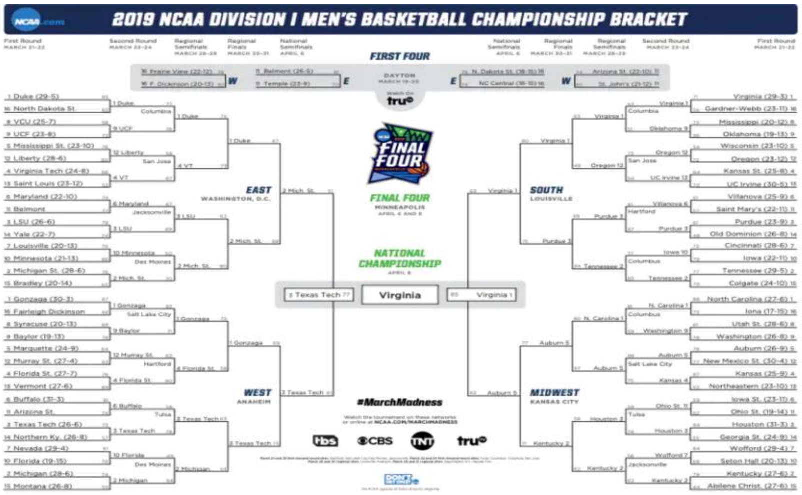


Figure 2: 2019 March Madness Tournament Bracket

The goal of the project is to use data from past March Madness tournament's and accurately predict how many points a team will score against a given opponent and ultimately determine who is most likely to win each game. The variables in the dataset are predominantly regular season and playoff totals, averages and rankings. These data points are combined with March Madness seeding and game results to round out the dataset. Results from the 2020-21 season will also be gathered as the season progresses to test the final model on this year's tournament. The model is designed with the goal of predicting the winner of as many games as possible in the tournament.

About the Data

Using the Beautiful Soup package in Python, NCAA Men's Basketball data was scraped for the 2017-18, 2018-19, 2019-20 and 2020-21 seasons. This data came from four different sources: ESPN, Fox Sports, the NCAA and the Pomeroy College Basketball Ratings. These four data sources provide a plethora of input variables that can be considered for our prediction models.

In total there are 120 variables, including: AP Poll rankings (pre-season and final), Win-Loss Totals (overall, in conference, home, away, neutral and against top 25 opponents), points for/against (overall and in conference), three pointer totals, field goal totals, free throw totals, strength of schedule (overall and non-conference), a variety of analytics from Ken Pomeroy (offensive and defensive efficiency (for team and their opponents), adjusted tempo and a luck rating) and the results of each March Madness game (score and seeding). These inputs are available for both the team of interest and their opponent in any given game. This means that each game accounts for two observations in the data set. Table 1 shows a sample of our data sets structure. For our analysis we will use each team's points scored in a given game as the output variable and aim to predict this as accurately as possible based on their opponent and their own regular season statistics.

Team	Wins	Loses	AP Rank	Opponent	Opp Wins	Opp Loses	Opp AP Rank
Gonzaga	26	0	1	Baylor	22	2	3
Baylor	22	2	3	Gonzaga	26	0	1

Table 1: Sample of our data set structure.

March Madness consists of 67 games each year, four games in the "First Four" play-in round, 15 games in each of the four regions and three games in the "Final Four". This means we will have 134 observations for each tournament. Unfortunately, since the 2020 tournament was cancelled, we only have the necessary data for the 2018 and 2019 tournaments. Thus, in total we have 268 observations to feed into our analysis. It is important to note that the only missing values in our dataset are found in the AP Poll (pre-season and final) input, this occurs because only the top 25 teams are included in those rankings and leaves teams not ranked blank. This will need to be taken into account when we perform our analysis.

March Madness provides college basketball fans with an opportunity to see how teams from different conferences fair against one another and creates interesting scenarios where the best team from one conference could potentially be dominated by a mid level team in one of the top conferences. It is evident that the points scored in March Madness games (see Figure 3) has a much wider point range, with mean 70.35 and standard deviation 11.92, than the average points per game teams scored before the tournament (see Figure 4), mean 76.92 and standard deviation 5.45. This is a good example of the potential disparity between conferences as some conferences don't have the money, recruiting power and brand names to compete with the college basketball blue bloods. It also appears that upon a quick inspection teams, on average, tend to score less points in March Madness than they do in games prior to it.

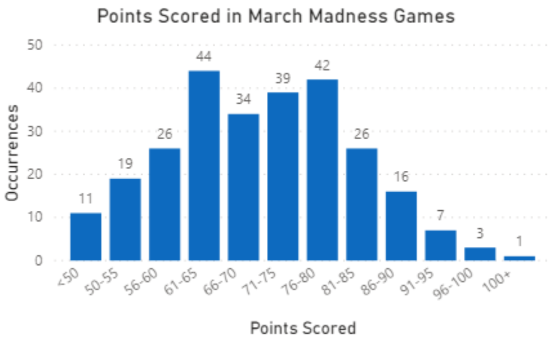


Figure 3: A bar chart showing the distribution of points scored by teams in 2018 and 2019 March Madness tournament games.

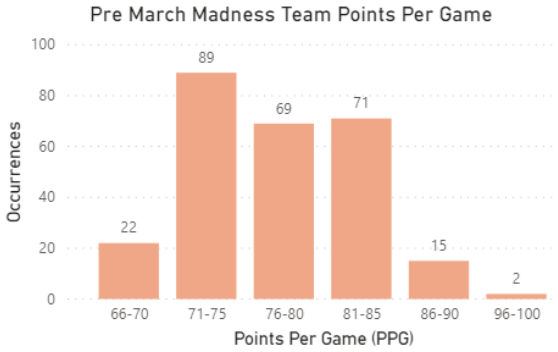


Figure 4: A bar chart showing the distribution of average points scored in the regular season by teams competing in the 2018 and 2019 March Madness tournaments. Notice the range of average points scored in this figure is much narrower than points scored in individual March Madness games.

It is worth noting that points per game before March Madness is positively correlated with points scored in a March Madness game, with a correlation coefficient of approximately 0.3804 (see Figure 5). Logically one would assume that it's obvious that the more a team scores, the more points they will score in March Madness. While that makes sense, we also must consider that some of those teams with high points per game are teams coming from weaker conferences. For example, entering the 2019 tournament Wofford was scoring approximately 96 points per game, but in March Madness they only averaged 70 points between their two games, these games are the two major outliers in Figure 5. It is good to see that there don't seem to be many of these cases and the logical assumption that higher points per games results in more points scored seems to hold.

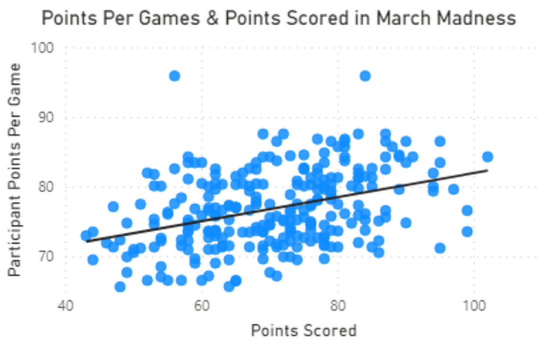


Figure 5: Overlaying a trend line on the scatter plot of points scored in March Madness games and their average points scored in the regular season, we see that these two variables are positively correlated.

We've mentioned the difficulty level between different conferences, but how does strength of schedule (a team that plays more difficult opponents has a higher strength of schedule) affect the number of games a team wins. As expected, Figure 6 illustrates that teams that win more games in a single tournament (maximum six wins), tend to have a higher strength of schedule. From this we can gather that, at least in 2018 and 2019, teams from better conferences tended to make it further in the tournament than teams from weaker ones. The significant outlier, at four games won, is the Cinderella story Loyola Chicago team that reached the Final 4 as an 11th seed in their region

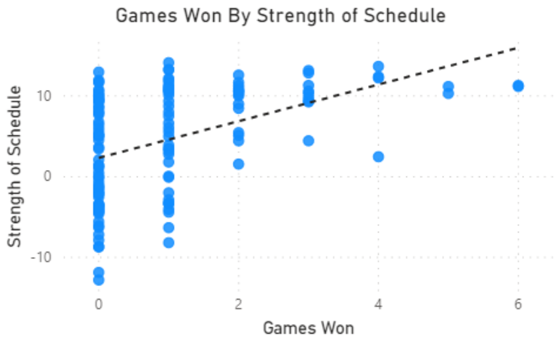
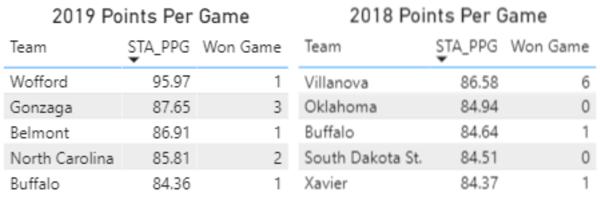


Figure 6: Scatter plot of regular season strength of schedule with March Madness tournament games won. The overlayed trend line depicts a positive correlation between strength of schedule and games won.

A statement that many coaches emphasize to their teams is that *the best offense is a good defense.* Is that the case when it comes to March Madness? It appears that teams with the top 5 least points against per game (see Figure 8) hold a slight advantage over teams with the top 5 points per game (see Figure 7). Between the 2018 and 2019 tournament top defensive teams reached the Final Four three times and won a combined 23 games versus top offensive teams who only won a combined 16 games. This is a fun comparison to make, but not the best for predictions since both scoring and not getting scored on are strictly situational and head-to-head a top offensive team may be able to simply overpower a defensive minded opponent.



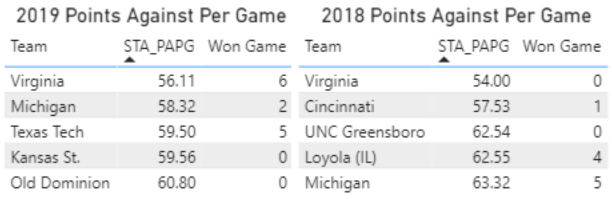
2019 Points Per Game

Team	STA_PPG	Won Game
Wofford	95.97	1
Gonzaga	87.65	3
Belmont	86.91	1
North Carolina	85.81	2
Buffalo	84.36	1

2018 Points Per Game

Team	STA_PPG	Won Game
Villanova	86.58	6
Oklahoma	84.94	0
Buffalo	84.64	1
South Dakota St.	84.51	0
Xavier	84.37	1

Figure 7: Top five offensive teams (by points per game) entering the 2018 and 2019 March Madness tournaments and how many games each team won in that years tournament.



2019 Points Against Per Game

Team	STA_PAPG	Won Game
Virginia	56.11	6
Michigan	58.32	2
Texas Tech	59.50	5
Kansas St.	59.56	0
Old Dominion	60.80	0

2018 Points Against Per Game

Team	STA_PAPG	Won Game
Virginia	54.00	0
Cincinnati	57.53	1
UNC Greensboro	62.54	0
Loyola (IL)	62.55	4
Michigan	63.32	5

Figure 8: Top five defensive teams (by points against per game) entering the 2018 and 2019 March Madness tournaments and how many games each team won in that years tournament. Top defensive teams seem more likely to advance deeper into the tournament than their offensive counterparts.

There do not appear to be any variables that are strongly correlated with points scored. Points For, Adjusted Offensive Efficiency and Opponent Points Against Per Game have the largest absolute coefficients at 0.417378, 0.434684 and 0.408683 respectively. These are reasonable since teams that score more, have a higher offensive efficiency and are playing teams that allow more points would be expected to score more than an average team.

OLS Regression

We will begin our statistical exploration by creating a simple linear regression model using all our available inputs. With the help of Python's Scikit-learn (Sklearn) library.

Before we run our regression, we will standardize the data. This is important because our variables are on a variety of different scales, for example wins and field goals attempted are on two different scales so it will not be intuitive as to which variable is more important to the model based off coefficients alone. To standardize the data, we will subtract the variables mean value from each observation and divide by the standard deviation.

A multiple linear regression model is of the form $\boldsymbol{Y} = \boldsymbol{\beta} \cdot \boldsymbol{X} + \boldsymbol{\varepsilon}$, where $\boldsymbol{Y}$ is a vector containing the observations of points scored, $\boldsymbol{\beta}$ is a vector containing the estimated coefficients for each of our variables, $\boldsymbol{X}$ is a matrix of all the independent variable values for each observation and $\boldsymbol{\varepsilon}$ is the random error of the model. The formula for the i^{th} observation can be written as follows, $y_i = \boldsymbol{\beta} \cdot \boldsymbol{x_i} + \varepsilon_i$, where y_i is the i^{th} observation of points scored, $\boldsymbol{\beta}$ is a vector containing the estimated coefficients for each of our variables, $\boldsymbol{x_i}$ is a vector of all the independent variable values for the i^{th} observation and ε_i is the error value of the i^{th} observation.

Inputting our standardized observations, we get the maximum and minimum coefficients displayed below in Table 2.

Variable	Coefficient Value	Variable	Coefficient Value
Opponent Adjusted Efficiency Margin	99.20	Opponent Strength of Schedule	-177.67
Opponent Average Opponents Offensive Efficiency	95.52	Adjusted Efficiency Margin	-144.48
Adjusted Offensive Efficiency	88.40	Opponent Average Opponents Defensive Efficiency	-93.24
Opponent Adjusted Defensive Efficiency	69.17	Adjusted Defensive Efficiency	-76.76
Losses	64.71	Opponent Adjusted Offensive Efficiency	-73.70

Table 2: This table contains the top five maximum and minimum coefficients calculated by our simple OLS regression using all our available data.

Nine out of the ten variables listed in Table 2 are advanced metrics, indicating that for the model using the full dataset the variables with the most impact are not things that can be counted while watching a game. Since college basketball can be unpredictable at times these underlying variables that are not necessarily visible to the human eye could be a significant proponent of the unpredictability.

The one variable included in Table 2 that is not an advanced statistic is Losses. The coefficient of the Losses variable suggests that teams that lost more games prior to March Madness score more in tournament games than teams with less losses. Although this may not be initially intuitive it reflects the issue mentioned earlier that teams from weaker conferences tend to struggle more against the tougher competition. It is realistic that teams with more losses score more points in the tournament since, in this case, losses are a testament to the number of quality teams in the top conferences that can be considered contenders to win the tournament.

Plugging our observations back into our newly created model to predict each team's points scored we find that we have a mean squared error of approximately 40.65. We don't have anything to compare this value to yet, but it is worth noting that our model likely overfits the data, so plugging the same data back in runs the risk of the model being fit to the data's nuances. Figure 9 displays how our models predicted points scored compare to the actual points scored.

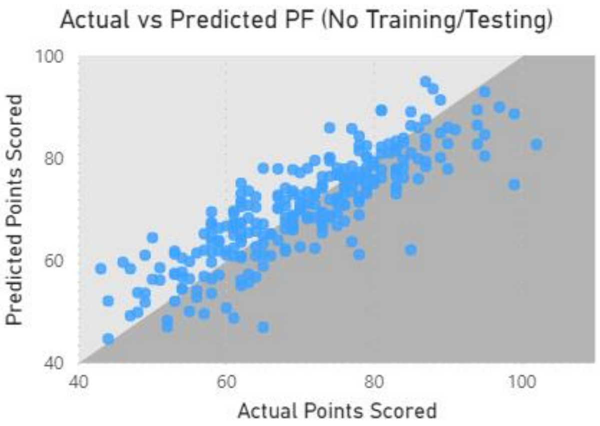


Figure 9: A scatter plot comparing our predictions of each teams points scored with the amount of points the team scored for our model using the full dataset.

Now that we have our predictions for the points each team will score, we can see how well our model did in predicting the outcome of each game. The model was able to correctly predict 121 (90.30%) of the 134 games between both tournaments. Predicting 90% of the games correctly is respectable, but, again, since our model is using all the data, and making its predictions on games that it had already used in its creation, it may have taken on the nuances of the data and overfit to the specific games that it learned from.

To take care of our potential overfitting issue we will split the data into training and testing datasets. By doing this we will no longer be making predictions on data that we used to create the model.

Our training data will contain a randomly selected 100 games (100/134=74.63% of the data) and our testing data will contain the remaining 34 games (34/134=25.37%).

Again, we will standardize our training data and use Sklearn to run a linear regression model. Doing so we receive the maximum and minimum coefficients in Table 3.

Variable	Coefficient Values	Variable	Coefficient Values
Opponent Adjusted Efficiency Margin	142.72	Opponent Strength of Schedule	-198.28
Opponent Average Opponents Offensive Efficiency	111.42	Opponent Average Opponents Defensive Efficiency	-108.41
Opponent Adjusted Defensive Efficiency	90.15	Opponent Adjusted Offensive Efficiency	-101.54
Strength of Schedule	82.75	Points Against Per Game	-51.89
Points Against (Conference Games)	60.43	Points Against Per Game (Conference Games)	-49.84

Table 3: This table contains the top five maximum and minimum coefficients calculated by our simple OLS regression using our 100 training games as the inputted data.

From Table 3 we can see that not all the top five maximum and minimum coefficients are the same between the model created using the full dataset and the model created using the training data. Notably Adjusted Efficiency Margin, Adjusted Defensive Efficiency, Adjusted Offensive Efficiency and Losses were replaced in their respective top fives by Points Against Per Game, Points Against Per Game (Conference), Strength of Schedule and Points Against (Conference). The training model seems to put more emphasis on a teams points against, where the model of the full dataset put more emphasis on the advanced metrics. Both of the model's top fives, however, still contain primarily advanced metrics.

Similar to our first model we have a questionable variable contributing positively to points scored in the tournament. Previously this variable was Losses, but in this model, it is replaced by Points Against (Conference Games). Again, one would assume that a team that allows more points against their conference opponents would be a weaker team and therefore score less in March Madness games. The argument that the team is likely representing a tougher conference, as mentioned for the positive effect of Losses, provides justification for teams that appear to be weaker defensively fairing well in the tournament. An example of a situation that could promote this trend is from the 2018 tournament where the fifth seeded Clemson defeated twelfth seeded New Mexico State by a score of 79-68. In 2018 Clemson was only the fourth best team in their conference and had a record of 23-9 relative to New Mexico State's 28-5.

Ultimately, the positive coefficient on Points Against (Conference Games) may be caused by teams that put more emphasis on offense than defense and believe that scoring the most points possible gives them the best chance to win.

As one would expect, there seems to be a difference in the distribution of the coefficient values between the two models. From Figures 10 and 11, we can see that the coefficients of the training model seem to disperse further from zero than the coefficients of the original model using the full dataset. Where the coefficients of the original model are more closely clustered around zero creating a somewhat normal distribution, the coefficients in the training model are further from zero and do not look to be normally distributed.

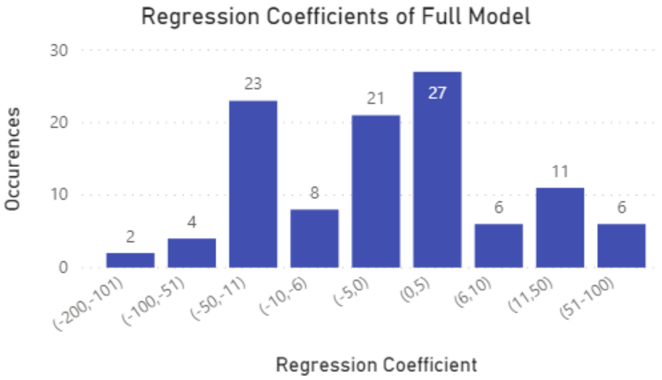


Figure 10: A bar chart of the coefficients from the original model using the full dataset. Coefficients seem to be clustered around zero, creating a somewhat normally distributed look. Approximately 53.7% of the coefficients are negative.

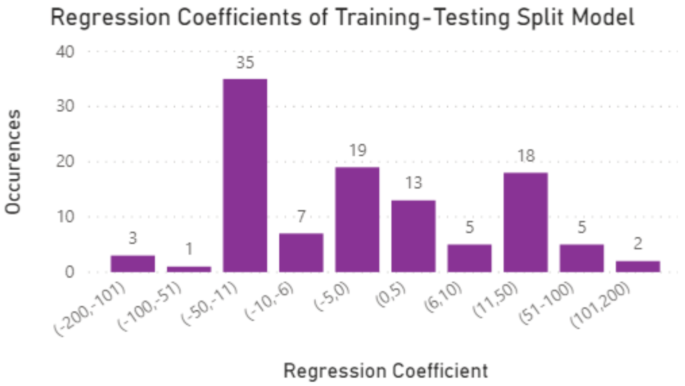


Figure 11: A bar chart of the coefficients from the model created using the training data. Coefficients seem to vary more than the original model and are not as clustered around zero. Approximately 60.2% of the coefficients are negative.

To test the accuracy of our new model, we now standardize our testing data with the same mean and standard deviations used from the training data and plug it into our training model to check the accuracy of our model. The mean squared error for this model is approximately 496.64, about twelve times that of our original model. Even with this drastic increase we should not be worried since our previous model was making predictions on data that was used in its creation and may be overfitting. Figure 12 shows how the predictions made on our testing dataset compare to the observed values.

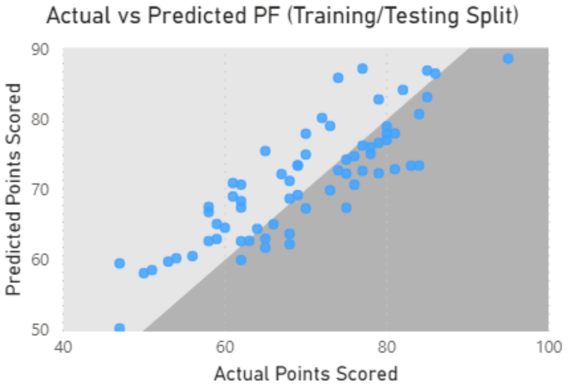


Figure 12: A scatter plot comparing our predictions of each teams points scored with the amount of points the team scored for our training-testing model.

Again, we can compare how our model did in predicting the outcome of each game. Our randomly selected training model was able to correctly predict the outcome of 29 (85.29%) of the 34 games in the testing dataset, an approximately five percentage point decrease.

To try to improve upon our model we will next use a ridge regression approach to adjust the coefficient weights to give the more influential variables more impact on our predictions and will hopefully be able to improve upon our initial simple linear regression models.

By adjusting the weights of the variables to increase the effect on points scored of the most impactful variables we introduce a bias-variance trade-off. Where the simple linear regression model aims to minimize variance without the presence of bias, the bias-variance trade-off allows us to try to reduce the variance even more at the expense of introducing bias into our model. The potential inclusion of bias into our model may be able to positively impact the accuracy of our predictions. The bias-variance trade-off is derived as seen below.

$$
\begin{aligned}
& E((y-\hat{y})^2) \\
&= E[(y-E(\hat{y})+E(\hat{y})-\hat{y})^2] \\
&= E\left[\left(\left(y-E(\hat{y})\right)+(E(\hat{y})-\hat{y})\right)\cdot\left(\left(y-E(\hat{y})\right)+(E(\hat{y})-\hat{y})\right)\right] \\
&= E\left[\left(y-E(\hat{y})\right)^2+\left(y-E(\hat{y})\right)\cdot(E(\hat{y})-\hat{y})+(E(\hat{y})-\hat{y})\cdot\left(y-E(\hat{y})\right)+(E(\hat{y})-\widehat{y)}^2\right] \\
&= E\left[\left(y-E(\hat{y})\right)^2+2\cdot\left(y-E(\hat{y})\right)\cdot(E(\hat{y})-\hat{y})+(E(\hat{y})-\hat{y})^2\right] \\
&= E\left[\left(y-E(\hat{y})\right)^2\right]+2\cdot E\left[\left(y-E(\hat{y})\right)\cdot(E(\hat{y})-\hat{y})\right]+E[(E(\hat{y})-\hat{y})^2] \\
&= E\left[\left(y-E(\hat{y})\right)^2\right]+2\cdot E\left[\left(y-E(\hat{y})\right)\right]\cdot E[(E(\hat{y})-\hat{y})]+E[(E(\hat{y})-\hat{y})^2] \\
&= E\left[\left(y-E(\hat{y})\right)^2\right]+2\cdot E\left[\left(y-E(\hat{y})\right)\right]\cdot 0+E[(E(\hat{y})-\hat{y})^2] \\
&= E\left[\left(y-E(\hat{y})\right)^2\right]+E[(E(\hat{y})-\hat{y})^2] \\
&= E[Bias(\hat{y})^2]+Var(\hat{y}) \\
&= \ Bias(\hat{y})^2+Var(\hat{y})
\end{aligned}
$$

Here y is the fixed value of the true regression formula that we are attempting to estimate, and it is independent of $\hat{y}$.

A model with low bias and high variance will tend to have a low mean squared error on the training data, however it will likely not perform as well on the testing data since the model has overfit the training data and increased the model complexity.

A model with high bias and low variance will tend to perform worse on the training data, as it oversimplifies the model, thus underfitting the data. Figure 13 shows a depiction of the change in prediction error as bias decreases and variance increases from The Elements of Statistical Learning textbook (Hastie et al. 220). As variance increases the model complexity also increases. The ridge and

LASSO regression models will attempt to create models that minimize the training error (i.e. find the minimum of the red curve in Figure 13).

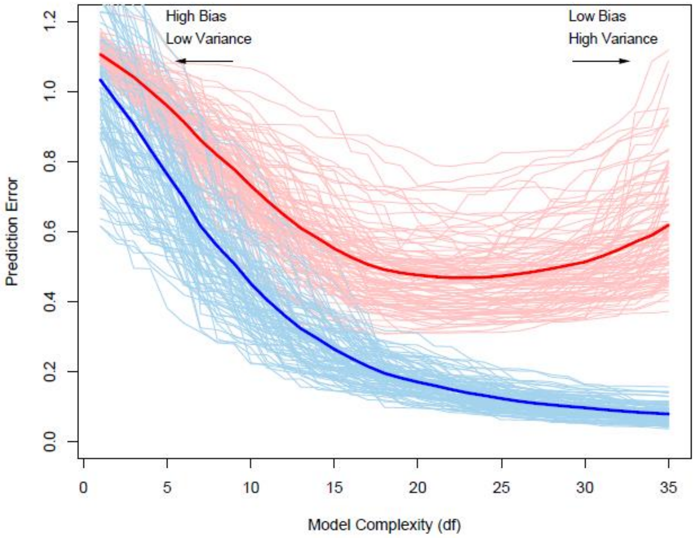


Figure 13: Line graph showing the change in training and testing error as model complexity increases. The red curves represent the testing error, while the blue curves represent the training error. The light red and blue curves represent the errors for 100 training sets of size 50 and the solid curves represent the expected training and testing errors.

In addition to the bias-variance trade-off we will touch on the effective degrees of freedom of our models. The effective degrees of freedom typically represent the number of parameters included in a regression model. The more parameters included in the model the higher the model's complexity, since there are a larger number of variables that our predictions are dependent on.

In the case of ridge regression, we have the formula, below, to calculate its effective degrees of freedom.

$$df(\lambda) = tr[\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X} + \lambda\mathbf{I})\boldsymbol{X}^T]$$

$$= tr[\boldsymbol{H}_\lambda]$$

$$= \sum_{j=1}^{p} \frac{d_j^2}{d_j^2+\lambda}$$

In the formula we initially have the trace of the hat matrix for ridge regression, where ***X*** represents our matrix of independent variables, λ represents the constant value in our penalty term (introduced in the next chapter) and ***I*** represents the identity matrix. This is then rewritten as the summation over the total number of parameters of the squared singular values of ***X*** divided by the squared singular values of ***X*** plus our constant λ.

In the case of LASSO regression, we don't have an exact formula to calculate the effective degrees of freedom of the model, but it can be estimated as follows:

$$df(\hat{y}) = \frac{1}{\sigma^2}\sum_{i=1}^{N} Cov(\widehat{y_i}, y_i)$$

where $Cov(\widehat{y_i}, y_i)$ is the sampling covariance of the predicted value and actual value, and σ^2 is the variance of the model.

Using this bias-variance trade-off property, we will attempt to create an even more accurate prediction model with ridge regression, along with introducing LASSO regression in our next chapter.

Ridge & LASSO Regression

First, we will create a ridge regression model. Similar to ordinary least squares, ridge regression will estimate the coefficients of each variable by minimizing the residual sum of squares, however, now a penalty term will be introduced that will shrink the regression coefficients, and in the case of LASSO regression, coefficients that make a minimal contribution to the model will be zero. This is because a small change in the derivative of our β is the same for all β's, thus small β's receive a relatively large penalty and are set to zero. Additionally, due to the absolute value in the LASSO penalty term being non-differentiable at zero, there exists no simple solution and estimates must be found numerically. Figure 14 shows the difference between ridge and LASSO regression penalty terms graphically.

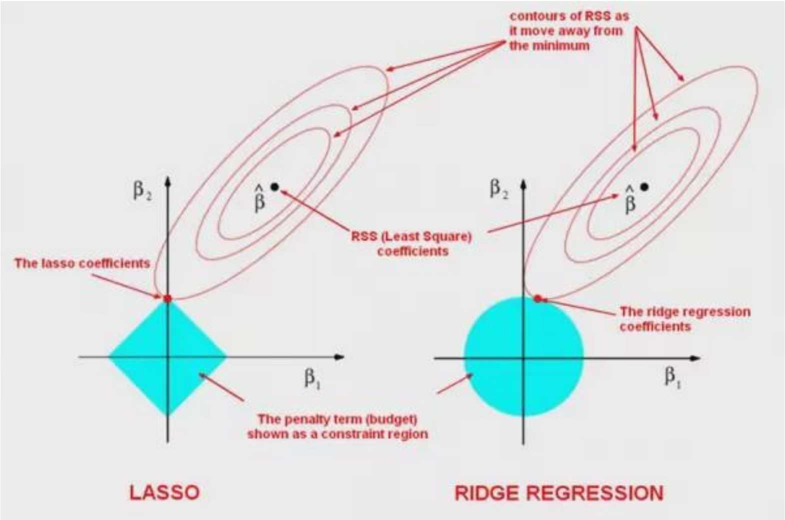

contours of RSS as it move away from the minimum
β₂
β̂
RSS (Least Square) coefficients
The lasso coefficients
β₁
The penalty term (budget) shown as a constraint region
β₂
β̂
The ridge regression coefficients
β₁
LASSO
RIDGE REGRESSION

Figure 14: Graphical representation of the difference between ridge and LASSO regression penalty terms.

The penalty will hopefully help our model minimize unnecessary noise from less influential variables and create more accurate estimates. An increase in the constant λ in the penalty term leads to an increase in the bias of our model allowing the most influential variables to have the greatest coefficients and thus, also, decreasing the variance in our predicted model.

The equation used to estimate the coefficients of our ridge regression model can be written as follows:

$$\hat{\beta}^{ridge} = \min_{\beta}\left\{\sum\nolimits_{i=1}^{N}(y_i - \sum\nolimits_{j=0}^{p} x_{ij} \cdot \beta_j)^2 + \lambda \cdot \sum\nolimits_{j=0}^{p} \beta_j^2\right\}$$

where N is the number of observations in the data set, y_i is the i^{th} observation of points scored, p is the number of parameters in our model, x_{ij} is the i^{th} observation of the j^{th} parameter, β_j is the estimated coefficient of the j^{th} parameter and λ (lambda) is the chosen constant of our penalty term (if $\lambda = 0$ we are performing a simple linear regression). The last term is our penalty term and contains an L_2 penalty. This formula for calculating the ridge regression coefficients aims to minimize the sum of the difference between our observed values and predicted value, and the sum of our squared estimated coefficients. This is an example of the bias-variance trade-off with the addition of bias from the penalty term to counteract the variance from the first term that may cause the model to overfit the data. The $\lambda \cdot \sum_{j=0}^{p} \beta_j^2$ term is the ridge regression penalty term. The derivation of the ridge regression solution is included below.

$$(\boldsymbol{y} - \boldsymbol{X}\beta)^T(\boldsymbol{y} - \boldsymbol{X}\beta) + \lambda\beta^T\beta$$

$$= \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{X}\beta - \beta^T\boldsymbol{X}^T\boldsymbol{y} + \beta^T\boldsymbol{X}^T\boldsymbol{X}\beta + \lambda\boldsymbol{I}\beta^T\beta$$

$$= \boldsymbol{y}^T\boldsymbol{y} - \beta^T\boldsymbol{X}^T\boldsymbol{y} - \beta^T\boldsymbol{X}^T\boldsymbol{y} + \beta^T\boldsymbol{X}^T\boldsymbol{X}\beta + \beta^T\lambda\boldsymbol{I}\beta$$

$$= \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{X}^T\beta^T\boldsymbol{y} + \beta^T(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})\beta$$

Minimize in terms of β by taking the partial derivative

$$\frac{\partial(\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{X}^T\beta^T\boldsymbol{y} + \beta^T(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})\beta)}{\partial\beta}$$

$= 0 - 2\boldsymbol{X}^T\boldsymbol{y} + 2(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})\beta$ since $\frac{\partial a^TCa}{\partial a} = (C + C^T)a = 2Ca$ when C is symmetric)

Set partial derivative equal to zero and solve for β

$$0 = -2\boldsymbol{X}^T\boldsymbol{y} + 2(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})\beta$$

$$2\boldsymbol{X}^T\boldsymbol{y} = 2(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})\beta$$

$$\boldsymbol{X}^T\boldsymbol{y} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})\beta$$

$$\therefore\ \hat{\beta}^{ridge} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1} \cdot \boldsymbol{X}^T\boldsymbol{y}$$

To create our ridge and LASSO models we will again use the SkLearn library with our previously created and standardized training and testing datasets. We will first run a ridge regression with the λ value in the penalty being one. Table 4 illustrates the maximum and minimum coefficients in the model.

Variable	Coefficient Values	Variable	Coefficient Values
Opponent Adjusted Defensive Efficiency	6.91	Adjusted Offensive Efficiency Rank	-4.70
Opponent Wins (Conference)	5.36	Opponent Neutral Site Wins	-4.53
Average Opponents Offensive Efficiency Rank	4.95	Opponent Adjusted Efficiency Margin	-4.53
Adjusted Offensive Efficiency	4.66	Opponent Non-Conference Strength of Schedule Rank	-4.43
Points For	4.57	Opponent Non-Conference Strength of Schedule	-3.83

Table 4: This table contains the top five maximum and minimum coefficients calculated by our ridge regression model with λ=1.

Noticeably our maximal and minimal coefficients have substantially decreased in magnitude from our previous OLS model. This is expected since the penalty term in the ridge regression model aims to minimize our λ multiplied by the square of the coefficients and the smaller the coefficients the lesser the penalty.

As in our previous models the maximum and minimum coefficients contain predominantly advanced statistics. This continues to support our narrative that games may be unpredictable due to some underlying variables not immediately visible to the human eye.

The maximum and minimum non-advanced statistics in this model are: Opponent Wins (Conference), Points For and Opponent Neutral Site Wins. The positive or negative impact of Opponent Wins (Conference), Points For and Opponent Neutral Site Wins (Neutral Site Wins is important because all March Madness games are played at a neutral site, thus the better a team fairs prior to the tournament would indicate their comfort level during the tournament) reasonably indicate the effect we would expect them to have on points scored in a tournament game.

Worth noting is that Opponent Non-Conference Strength of Schedule and Opponent Non-Conference Strength of Schedule Rank both have a significant negative impact on a team's points scored. This is peculiar since an opponent who plays difficult non-conference games would be expected to have a high Opponent Non-Conference Strength of Schedule and low Opponent Non-Conference Strength of Schedule Rank, so they would be expected to be on different ends of the maximal-minimal coefficient spectrum. I believe that this may have occurred because schools choose many of their non-conference opponents and this subset of NCAA basketball teams, may create an odd outcome once the data is standardized.

With this ridge regression model, we can perform another set of predictions on our testing dataset. This ridge regression model (with λ=1) has a mean squared error of approximately 85.98, a substantial improvement from the 496.64 of the OLS model on the testing data. Figure 15 illustrates our predictions compared to the observed points scored.

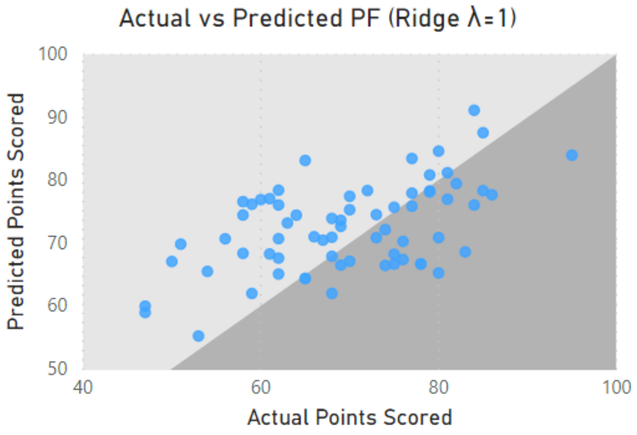


Figure 15: A scatter plot comparing our predictions of each teams points scored with the amount of points the team scored for our ridge regression model with λ=1.

Putting the predicted points scored together to form our predictions we find that our model predicted 27 (79.41%) of the 34 games. Our first ridge regression model performed slightly worse than our OLS model, however we have not yet optimized the λ value in our model, so that is what we will explore next.

To find the optimal λ value we ran ridge regressions for all λ's in the range 0 to 20, at three decimal place accuracy, and calculated the mean squared error for each model. The optimal λ minimizes the mean squared error and we found it to be 7.582. Figure 16 shows the mean squared error for each value of λ and Figure 17 focuses on our minimized mean squared error.

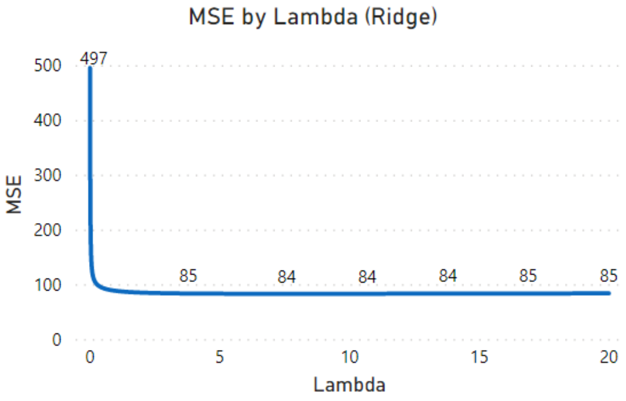


Figure 16: A line graph demonstrating the change in mean squared error as λ increases for ridge regression.

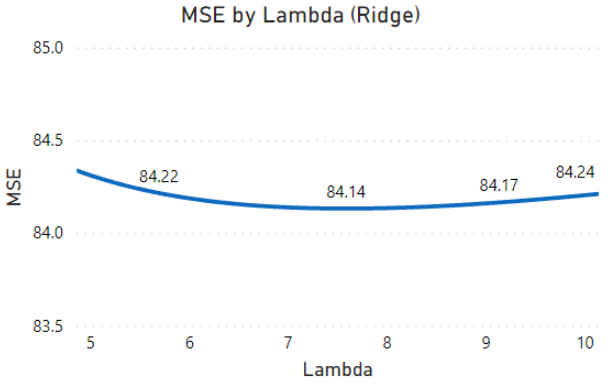


Figure 17: A line graph focusing on the turning point in Figure 16 where we find our minimum mean squared error for ridge regression.

Running our ridge regression with the optimal λ of 7.582 we have the maximum and minimum coefficients outlined in Table 5.

Variable	Coefficient Values	Variable	Coefficient Values
Opponent Adjusted Defensive Efficiency	3.82	Adjusted Offensive Efficiency Rank	-2.76
Adjusted Offensive Efficiency	2.84	Opponent Non-Conference Strength of Schedule	-2.43
Points Against	2.75	Automatic Bid	-2.42
Points For	2.60	Opponent Neutral Site Wins	-2.33
Adjusted Opponents Offensive Efficiency Rank	2.49	Opponent Adjusted Tempo Rank	2.29

Table 5: This table contains the top five maximum and minimum coefficients calculated by our ridge regression model with λ=7.582.

Again, we notice that our model contains predominantly advanced statistics along with four non-advanced statistics. Notably the Automatic Bid variable has entered the top five minimum coefficients indicating teams that earn an automatic entry into the tournament, by winning their conference playoffs, score less points than those who earn an At-Large Bid who are considered to be the best teams in the country, which is what one would expect.

One may observe from Table 5 that interestingly Adjusted Offensive Efficiency and Adjusted Offensive Efficiency Rank are at opposite ends of the maximum magnitude spectrum. This is expected since a higher quality team would be expected to have a high Adjusted Offensive Efficiency and low Adjusted Offensive Efficiency Rank.

Using our new ridge regression with λ=7.582 to create predictions for our testing dataset we have a mean squared error of approximately 84.14. This model was able to predict the outcome 27 (79.41%) of the 34 games. Figure 18 depicts the actual points scored against the number of points that our optimal ridge regression model predicted.

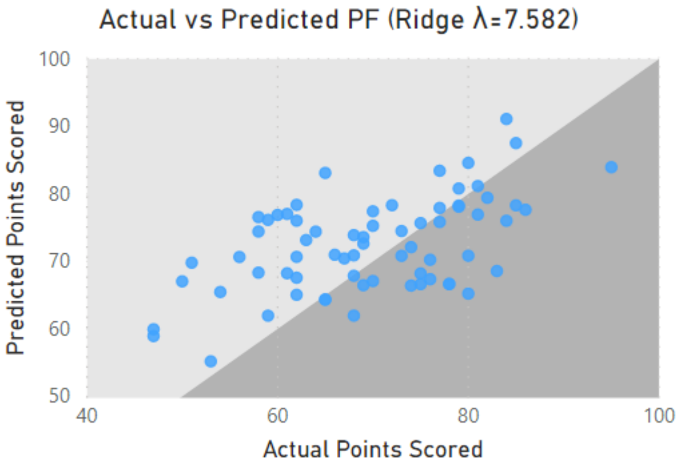


Figure 18: A scatter plot comparing our predictions of each teams points scored with the amount of points the team scored for our ridge regression model with λ=7.582. Notably this figure looks very similar to Figure 15 where λ=1.

Ultimately there is minimal difference between our ridge regression for λ=1 and λ=7.582. The mean squared errors only differ by approximately two (85.98 and 84.14 respectively) and both models predicted 27 games correctly with similar plots of actual points scored compared to predicted points scored.

Now that we have an optimal ridge regression model, we will introduce the least absolute shrinkage and selection operator (LASSO) regression to try to improve on our current accuracy. Where the penalty term for ridge regression aimed to minimize the λ value multiplied by the sum of squared regression coefficients (L_2 penalty), the LASSO model aims to minimize a penalty term with λ multiplied by the sum of the absolute regression coefficients (L_1 penalty).

The equation used to estimate the coefficients of our LASSO regression model can be written as follows:

$$\hat{\beta}^{LASSO} = \min_{\beta}\left\{\sum_{i=1}^{N}\left(y_i - \sum_{j=0}^{p} x_{ij}\cdot\beta_j\right)^2 + \lambda\cdot\sum_{j=0}^{p}\left|\beta_j\right|\right\}$$

where all the variables are the same as our ridge regression apart from the change from L_2 penalty to L_1 penalty.

Earlier we were able to derive the ridge regression solution, but there does not exist an analytic solution for LASSO regression. The computation of the LASSO regression solution is a quadratic programming problem and requires an efficient algorithm to compute the set of solutions.

To create our LASSO models, we will again enlist the help of the SkLearn library and our previous training and testing datasets. Running a LASSO regression with λ=1 as a starting point we get the maximum and minimum coefficients in Table 6.

Variable	**Coefficient Values**	**Variable**	**Coefficient Values**
Points For	3.32	Average Opponents Defensive Efficiency	-1.64
Opponent Points Against Per Game	1.67	Opponent Adjusted Tempo Rank	-1.21
Opponent Adjusted Defensive Efficiency	1.47	Adjusted Offensive Efficiency Rank	-1.16
Opponent Conference Wins	0.37	Opponent Non-Conference Strength of Schedule	-0.96
Points Against	0.36	Opponent Neutral Site Wins	-0.95

Table 6: This table contains the top five maximum and minimum coefficients calculated by our LASSO regression model with λ=1.

The maximum and minimum coefficients of our first LASSO regression model, for the first time, contain an equal amount of advanced and non-advanced statistics. Most of these ten variables appear in our previous models, this time with the introduction of Opponent Conference Wins. Facing an opponent with many wins in conference play most likely indicates they come from a weaker conference and thus more points will likely be scored on them in March Madness.

In our previous OLS and ridge regression model each variable had its own non-zero coefficient. While minimizing the penalty a LASSO regression may set the coefficients of any number of variables to zero. Our LASSO model with λ=1 only has thirteen non-zero variable coefficients, meaning approximately 88% of our variables were given a coefficient of zero and do not affect our predictions. Ten of our thirteen non-zero coefficients are listed in Table 6 with the remaining three being: Adjusted Tempo (0.20), Field Goals Attempted (0.05) and Non-Conference Strength of Schedule Rank (-0.26).

Using our new LASSO regression model to make predictions on our testing dataset we have a mean squared error of approximately 75.90, our lowest mean squared error thus far. Our model also predicted 29 (85.29%) of the 34 testing games correctly, which tied for the most testing games predicted correctly with the OLS model. Figure 19 shows our models predictions compared to the actual number of points scored.

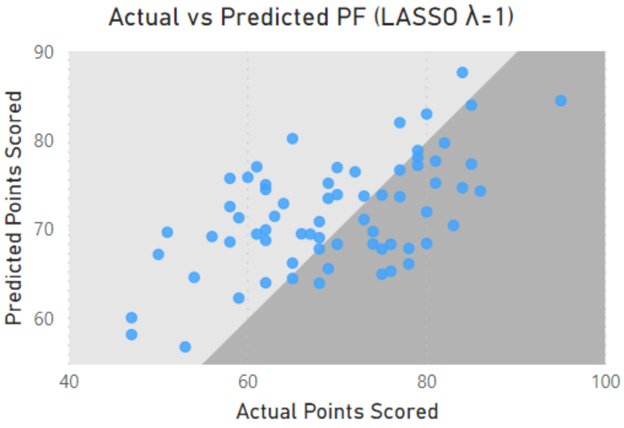


Figure 19: A scatter plot comparing our predictions of each teams points scored with the amount of points the team scored for our LASSO regression model with λ=1.

Similar to our ridge regression exploration, we can optimize our λ value to minimize the mean squared error of our model and hopefully improve the prediction accuracy in the process. Again, running our LASSO regression for λ values in the range 0 to 20 we can find our optimal λ, with it eventually being 1.246. Figures 20 and 21 show how the mean squared error changes with each increase of λ. Note that SkLearn's estimates for LASSO regression are inaccurate for small positive λ's due to "numerical reasons" as noted in its documentation (*Sklearn.linear_model.Lasso)*. This is apparent since LASSO mean squared error for λ=0 does not match our previously calculated mean squared error for our OLS model.

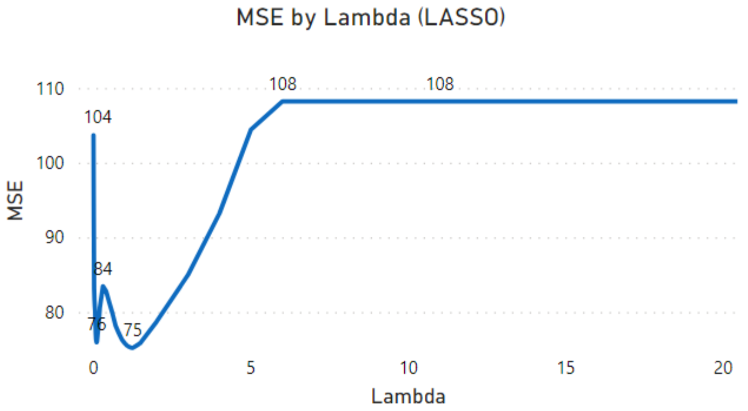


Figure 20: A line graph demonstrating the change in mean squared error as λ increases for LASSO regression.

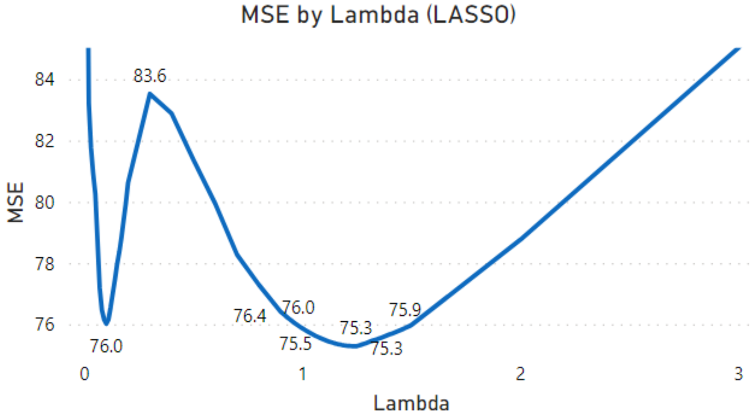


Figure 21: A line graph focusing on the turning point in Figure 20 where we find our minimum mean squared error for LASSO regression.

Now that we have the optimal λ of 1.246 that minimizes our models mean squared error we find the maximum and minimum coefficients in Table 7.

Variable	**Coefficient Values**	**Variable**	**Coefficient Values**
Points For	3.15	Average Opponents Defensive Efficiency	-1.38
Opponent Points Against Per Game	1.54	Opponent Adjusted Tempo Rank	-1.19
Opponent Adjusted Defensive Efficiency	1.38	Adjusted Offensive Efficiency Rank	-1.14
Points Against	0.48	Opponent Non-Conference Strength of Schedule Rank	-0.77
Adjusted Tempo	0.06	Opponent Neutral Site Wins	-0.70

Table 7: This table contains the top five maximum and minimum coefficients calculated by our LASSO regression model with λ=1.246.

Our optimal LASSO model contains twelve of the thirteen variables from our model with λ=1. The one variable that no longer has a non-zero coefficient is Field Goals Attempted, which was already only approximately 0.5. The two variables not included in Table 7 are: Opponent Conference Wins (0.00002) and Non-Conference Strength of Schedule Rank (-0.10).

Creating our predictions for points scored on our testing data we have a minimum mean squared error of approximately 75.31, only slightly less than the approximately 75.90 from our original LASSO model. We also find that we again predict 29 (85.29%) of the 34 test games correctly. Figure 22 compares our predicted points scored to the observed number of points scored.

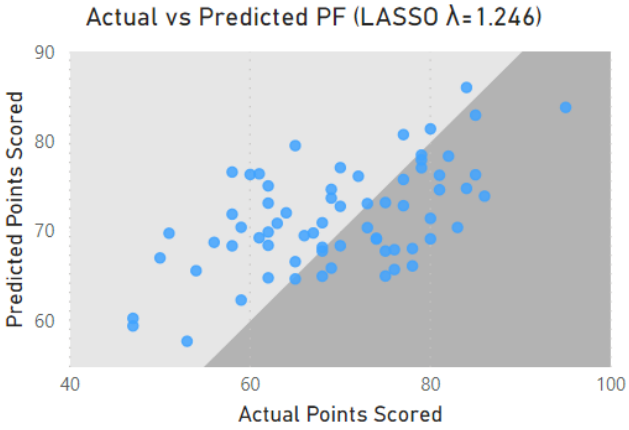


Figure 22: A scatter plot comparing our predictions of each teams points scored with the amount of points the team scored for our LASSO regression model with λ=1.246. Note this graph is quite similar to Figure 19 where λ=1.

In both our ridge and LASSO regression explorations in this section our optimal model was similar to our initial models with λ=1. Both of our LASSO regression models predicted the same number of games correctly as our OLS model with 29 correct predictions, which is currently our maximum amount. Our optimal LASSO regression model currently holds the lowest mean squared error of approximately 75.31.

Next, we will explore cross-validation, to get a better estimate of the true accuracy of our models. Previously we used a single training-testing split to create our models, now we will use several training-testing splits that will indicate whether our initial training-testing split was overestimating, underestimating or was reasonably accurate.

Cross-Validation

In statistics cross-validation is a type of resampling method where we draw multiple samples from a data set and fit a model to each of our samples. We then analyze our fitted models and can create a model that averages the results of our group of fitted models to try to optimize our prediction accuracy.

Cross-validation is an extension of our previous approach of splitting the data into training and testing sets. Previously, we only split the data into two random samples and could potentially have our predictions skewed by a poor training-testing split. Increasing the number of random samples allows us to avoid our model being influenced by one poor split.

The cross-validation method we will use to create our random samples is *k*-fold cross-validation. In *k*-fold cross-validation the data is split into *k* random folds/samples. These *k* folds are then used to fit *k* models with one of the folds left out of each model to make predictions on later. One model, for example, could use the first fold as its testing set and the model would be fit using the other *k* – 1 folds.

We will use *k*-fold cross-validation with *k* = 5 to determine the optimal λ value for our ridge and LASSO regression models. To do this we first fit ridge and LASSO models to our folds to find the minimum mean squared error for each of the five samples. We fit our models for all λ values between 0 and 150 for ridge regression and between 0 and 20 for LASSO regression, with three decimal places of precision.

Next, we will use our fitted models to make predictions on the data that was left out for each of the five samples. Figure 23 shows the accuracy, based on mean squared error, of our five samples for each value of λ for our ridge regression.

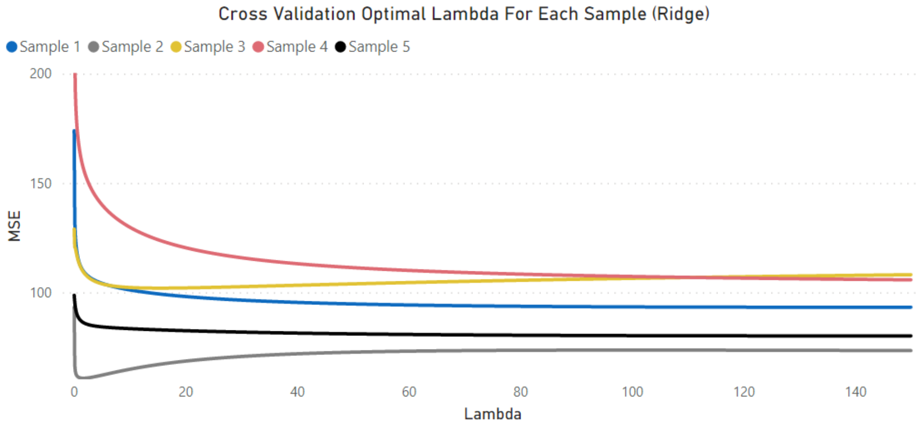


Figure 23: A line graph showing the change in mean squared error as λ increases for our five samples using ridge regression.

We see that, although there is slight variance between samples, the mean squared errors follow similar trends for each of the five estimates. Our optimal λ value occurs at the minimum average mean squared error of these five samples. Figures 24 and 25 show our optimal λ value found using *k*-fold cross-validation for our ridge regression model.

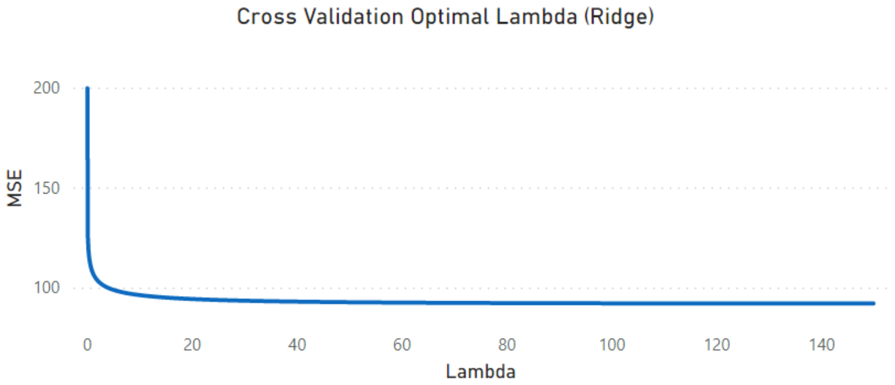


Figure 24: A line graph showing the average mean squared error of our five samples as λ increases, for ridge regression.

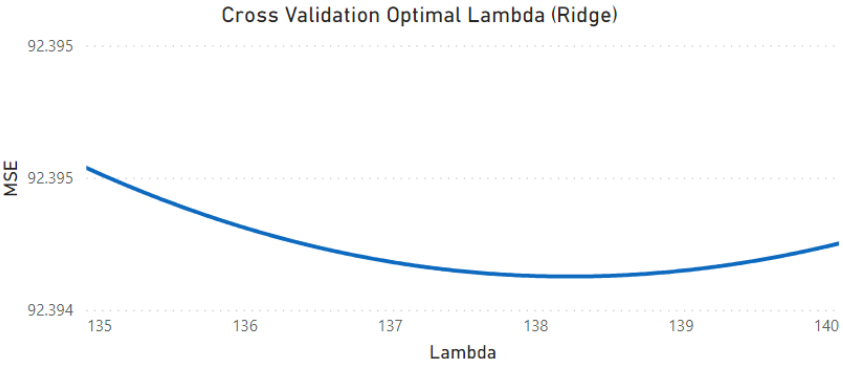


Figure 25: A line graph showing the minimum (138.230) of the average mean squared error of our five samples as λ increases, for ridge regression.

We find the optimal λ value for our ridge regression model, by *k*-fold cross-validation, to be 138.230. We can now fit our optimal ridge regression model. Table 8 shows the maximal and minimal coefficients of our optimal model.

Variable	**Coefficient Values**	**Variable**	**Coefficient Values**
Opponent Adjusted Defensive Efficiency	1.26	Opponent Neutral Site Wins	-1.10
Points For	1.03	Opponent Adjusted Tempo Rank	-0.96
Adjusted Offensive Efficiency	0.99	Adjusted Offensive Efficiency Rank	-0.94
Points Against	0.98	Opponent Non-Conference Strength of Schedule Rank	-0.76
Opponent Points Against Per Game	0.95	Opponent Points For	-0.71

Table 8: This table contains the top five maximum and minimum coefficients calculated by our optimal ridge regression model with λ=138.230.

There appears to be an equal distribution of advanced and non-advanced statistics in our group of maximal and minimal coefficients. Each of these ten variables seem to be reasonable predictors of points scored.

Our next chapter will explore the predictions made by our optimal ridge regression model, but first we must follow similar steps to find our optimal LASSO regression model.

As we did with our ridge regression model, we will fit LASSO regression models to our *k* = 5 samples for all λ values between 0 and 20, with three decimal accuracy. These models can then be used to create predictions on the data that was not included when training each model. Figure 26 shows the accuracy, based on mean squared error, of our five samples for each value of λ for our LASSO regression.

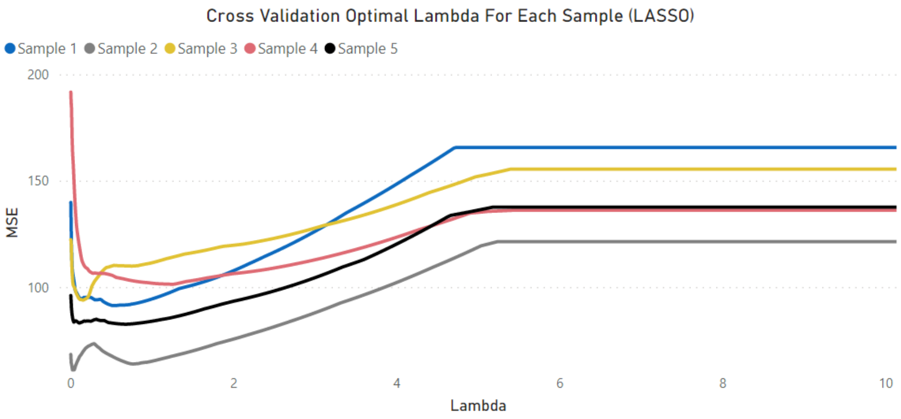


Figure 26: A line graph showing the change in mean squared error as λ increases for our five samples using LASSO regression.

Similar to what we saw with ridge regression, the mean squared errors follow similar trends for the five estimates, with some variance due to differences in the random samples. Our optimal λ value occurs at the minimum average mean squared error of these five samples. Figures 27 and 28 show our optimal λ value found using *k*-fold cross-validation for our LASSO regression model.

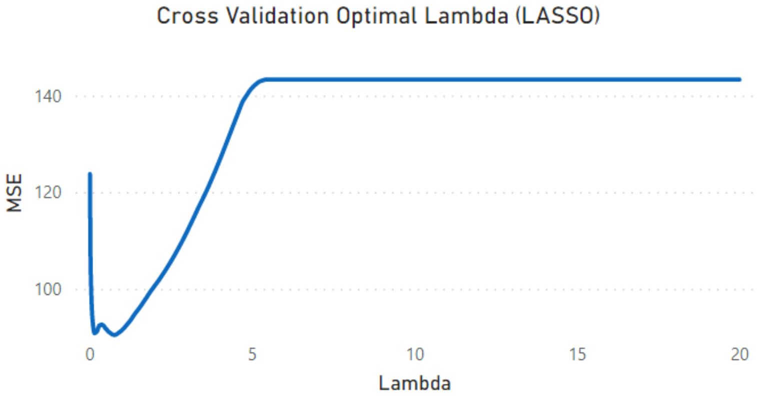


Figure 27: A line graph showing the average mean squared error of our five samples as λ increases, for LASSO regression.

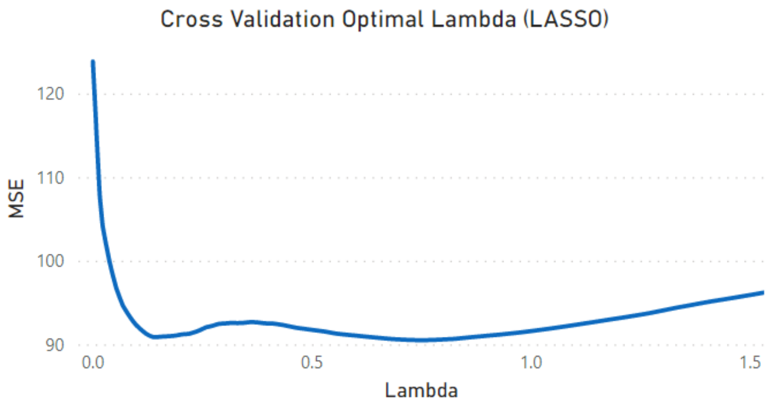


Figure 28: A line graph showing the minimum (0.748) of the average mean squared error of our five samples as λ increases, for LASSO regression.

We find the optimal λ value for our LASSO regression model, by *k*-fold cross-validation, to be 0.748. We can now fit our optimal LASSO regression model. Table 9 shows the maximal and minimal coefficients of our optimal model.

Variable	**Coefficient Values**	**Variable**	**Coefficient Values**
Opponent Adjusted Defensive Efficiency	2.64	Opponent Adjusted Tempo Rank	-1.61
Adjusted Offensive Efficiency	1.78	Adjusted Offensive Efficiency Rank	-1.35
Points For	1.69	Opponent Non-Conference Strength of Schedule Rank	-1.11
Points Against	0.54	Opponent Neutral Site Wins	-0.74
Adjusted Tempo	0.51	Average Opponents Defensive Efficiency	-0.39

Table 9: This table contains the top five maximum and minimum coefficients calculated by our optimal LASSO regression model with λ=0.748.

Overall, our optimal LASSO regression model, using *k*-fold cross-validation, has seventeen non-zero coefficients. This is a slight increase from the twelve non-zero coefficients in our optimal LASSO model found using our random training-testing split. This new optimal model has more non-zero coefficients than our previous model since it has a smaller λ value.

The variables included in our new optimal LASSO regression model not listed in Table 9 are: Opponent Points Against Per Game (0.36), Opponent Conference Wins (0.02), Three Point Attempts (0.01), Luck Rank (0.003), Opponent Adjusted Offensive Efficiency Rank (-0.01), Three Point Percentage (-0.15) and Non-Conference Strength of Schedule Rank (-0.17). Table 9 indicates that 70% of the maximal and minimal coefficients of our optimal LASSO regression model are advanced statistics and overall, the model is comprised of 59% advanced statistics.

Now that we have accounted for the potential influence of a poor training-testing split on our optimal models, we can use our new optimal ridge and LASSO regression models to predict the 2021 March Madness tournament.

Results

Finally, we can test the accuracy of our new optimal ridge and LASSO regression models by creating predictions for the 2021 March Madness tournament. Along with our two models I've also filled out a bracket of my choices for comparison. Each of our three brackets will be entered into the CBS Sports Bracket Challenge for comparison with the general public's brackets.

Prior to creating the predictions for this year's tournament, a data issue appeared in the 2020-21 data. Due to the impact of COVID-19 this season, teams played fewer games than in previous tournaments. In the case of Colgate they only played 15 games entering the tournament, while the average in past tournaments was approximately 33 games.

Since our data is standardized based on data from previous tournaments, count statistics like Points For and Three Point Attempts would be lower than usual and our predictions would be underestimated.

To combat this potential underestimation, we took the average games played of teams participating in previous tournaments and adjusted the count statistics of this year's participants to be in terms of the past years average. The issue is that all teams now have the same number of Games Played, but since Games Played is not included in either of our models it seems like a valid solution to our problem.

Now that our data issue is solved, we can create our predictions. Figure 29 shows our filled-out bracket for our LASSO regression model with green representing a correct prediction and red representing an incorrect prediction.

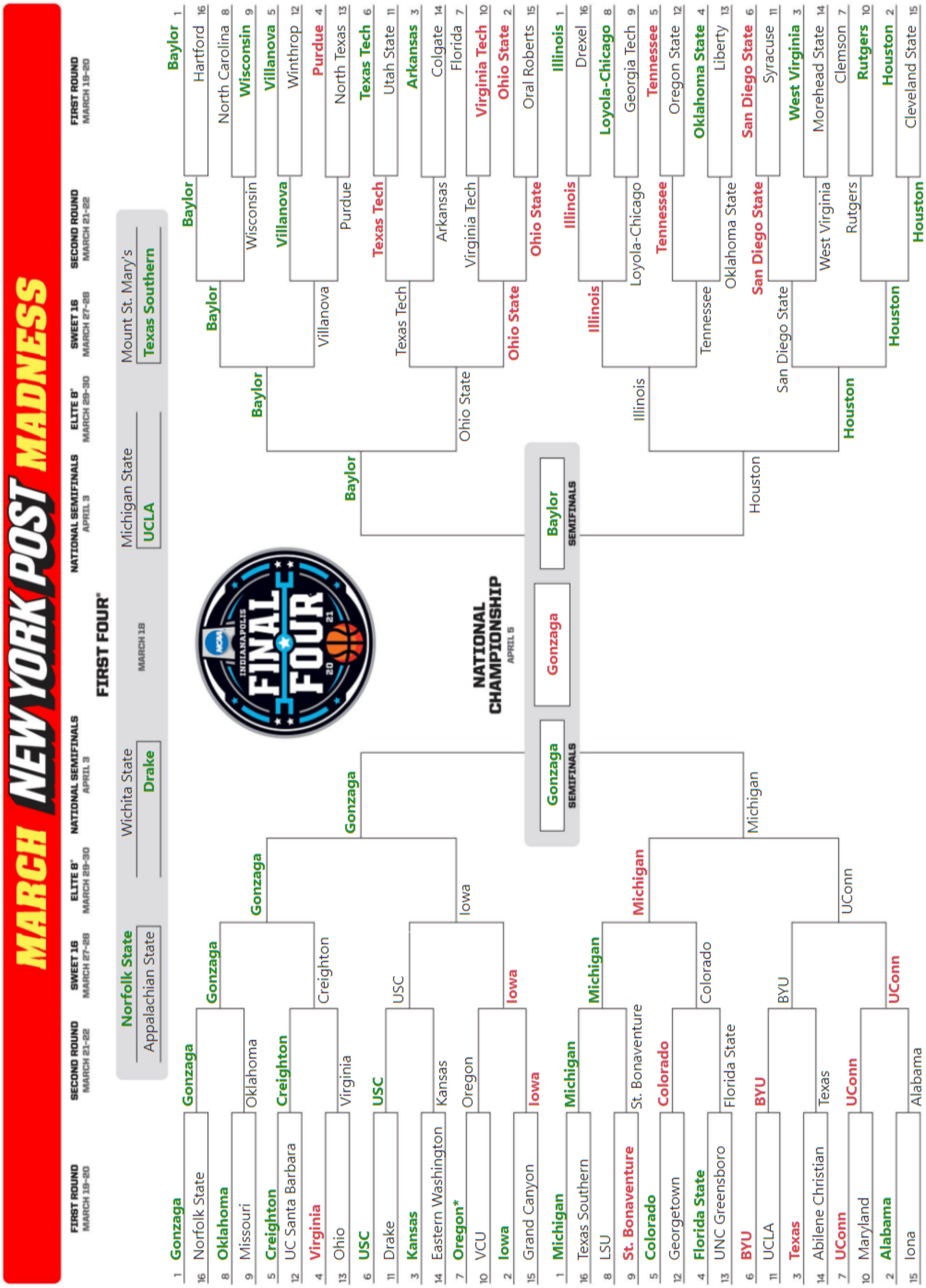


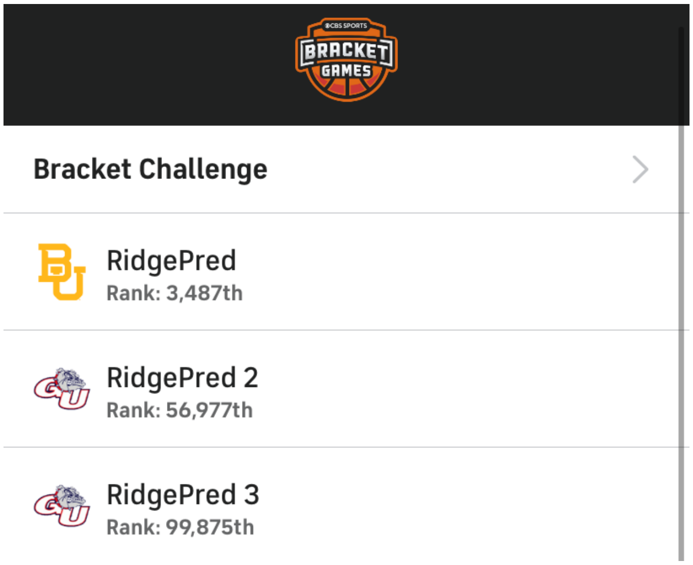
Figure 29: Filled out bracket of 2021 March Madness predictions from our LASSO regression model.
**Oregon advanced to the Second Round due to a VCU COVID-19 case.*

After a long three weeks with hours spent agonizing in front of the television as each game meant a possible increase to our model's accuracies, we see, at a quick glance, that there are a lot more teams highlighted in green than there are in red. Table 10 shows how our ridge and LASSO regression models performed against my own selections.

Round	LASSO	Ridge	My Predictions	Games
First Four	4	2	2	4
First Round	21*	20	23	31
Second Round	7	8	8	16
Sweet Sixteen	4	4	4	8
Elite Eight	3	3	2	4
Final Four	2	2	2	2
Championship	0	1	0	1
Total	**41**	**40**	**41**	**66**

Table 10: Correct predictions for each model by round with potential correct selections on right hand side.
**Correct prediction of Oregon victory removed due to VCU COVID-19 case.*

All three of our brackets performed admirably and Figure 30 shows the rankings of our three brackets in the CBS Bracket Challenge, out of at least 770,000 entries (the highest ranking I could find an individual having on social media).



Bracket Challenge

RidgePred
Rank: 3,487th

RidgePred 2
Rank: 56,977th

RidgePred 3
Rank: 99,875th

Figure 30: The final rankings of our three brackets with RidgePred representing our ridge regression predictions, RidgePred 2 representing our LASSO regression predictions and RidgePred 3 representing my predictions.

Notably points awarded in the CBS Bracket Challenge increase by round, so a correct prediction in the Second Round is worth two points, whereas a correct prediction in the Sweet Sixteen is worth four points. This explains why our ridge predictions finished much higher than our other models since it correctly predicted the championship and received an additional 32 points. Had Gonzaga won the championship, our LASSO predictions would have likely placed in the top 2000 as it was in 2858th place entering the final game.

The most impressive predictions, I believe, came on the first night of the tournament in the First Four. I found it quite surprising that our LASSO regression model correctly predicted all four of the First

Four games. With the teams in those games being very evenly matched all four games seemed as though either team could come out victorious. Two of these games were decided by a single point and another one was decided in overtime. Our LASSO model also correctly predicted three of the four 8 versus 9 seed games selecting #8 Oklahoma, #9 Wisconsin and #8 Loyola-Chicago.

Our models also predicted some upsets including: #10 Rutgers against #7 Clemson (LASSO & Ridge), #6 USC against #3 Kansas (LASSO), #11 Syracuse against #6 San Diego State (Ridge), #8 Loyola-Chicago against #1 Illinois (Ridge) and #11 Syracuse against #3 West Virginia (Ridge).

An issue that we have not accounted for when comparing the number of games that we correctly predicted is that the only time our models are, almost, guaranteed to be predicting the outcome of the same matchup is the First Round. This means that making comparisons of how many games were predicted correctly in Table 10 is not necessarily an apples-to-apples comparison.

To give us a better outlook on which model was truly the most accurate, we will create predictions for the exact 66 matchups that occurred in the tournament. Table 11 shows how our two models faired when predicting the same games.

Round	LASSO	Ridge	Games
First Four	4	2	4
First Round	21	20	31
Second Round	9	9	16
Sweet Sixteen	6	6	8
Elite Eight	3	3	4
Final Four	2	2	2
Championship	0	1	1
Total	**45**	**43**	**66**

Table 11: Correct predictions for each model by round, on exact tournament matchups, with potential correct selections on right hand side.

Although our LASSO regression model, again, correctly predicted more games than our ridge regression model we do not seem to have conclusive evidence that the LASSO model is definitively the better model. The only major difference between the two model's prediction accuracy is that the ridge regression model only predicted two of the first four games correctly. When filling out a bracket the first four games are not usually included in the competition and the LASSO models' advantage there is not necessarily a difference maker in any bracket competition. The ridge regression model also predicted the winner of the tournament, but this can't be looked at as the deciding factor since both models predicted three of the four teams in the Final Four and both teams in the Championship.

Now that we have a comparable set of predictions, we can also compare the mean squared error of our final prediction models. In addition to predicting a higher percentage of games correctly, our LASSO regression model also boasted a lower mean squared error of approximately 223.5009 to our ridge regression models 248.8991. With our LASSO regression model outperforming our ridge regression model in both accuracy and mean squared error it is fair to say that it is the better of our two models.

Conclusion

March Madness is a yearly spectacle bringing sports fans from around the world together to try to be the first individual to successfully predict the entire tournament. From the outset, this year's tournament was filled with upsets and timeless moments that will go down in history. From Oral Roberts becoming the ninth 15 seed to upset a 2 seed and progressing all the way to the Sweet Sixteen, to Jalen Suggs' half-court overtime buzz beater, for Gonzaga, to knock off the pesky 11 seed UCLA Bruins after they became only the second First Four team, in history, to reach the Final Four. With each game being so closely contested and the likelihood of a perfect bracket being so small our models valiantly fought an uphill battle, performing admirably.

Initially we fit a simple linear regression model to our data to develop a benchmark to compare our forthcoming models to. We found that the introduction of bias into our model, through the concept of the bias-variance trade-off, allowed our ridge and LASSO regression models with an initial value for λ of one to outperform our simple linear model.

Next, we explored the process of finding the λ value that minimizes our test data mean squared error and fit models with our new optimal λ value. This was another step in creating more accurate predictions, however there was still potential for our models to suffer from a poor training-testing split.

To combat the possibility of a poorly selected random split, we introduced the concept of *k*-fold cross-validation for our ridge and LASSO regression. This allowed us to split our data into *k* folds/samples (in our case $k = 5$ folds) and limit the opportunity of a poor split. We then averaged the resulting $k = 5$ models to find our optimal λ for both ridge and LASSO.

Using our new optimal λ values we fit ridge and LASSO regression models to our entire data set and proceeded to make predictions for the 2021 March Madness tournament.

Our ridge and LASSO predictions performed very well in the CBS Sports Bracket Challenge placing 3487th and 56977th respectively, predicting 40 and 41 games.

Since each model was not necessarily predicting the same matchups, we decided to have our two models create a final set of predictions on the actual tournament matchups that occurred. Our LASSO regression model, again, predicted more games than our ridge regression model, due to a perfect performance in the First Four.

With neither model clearly demonstrating itself as the definitive best choice we finally compared their mean squared errors for this year's tournament. Our LASSO regression model was able to solidify itself as the better model producing the lesser mean squared error.

This project demonstrated how difficult it is to accurately predict real-world events and an accurate model is such a commodity in any industry. There are numerous variables that our models are unable to account for that could play a factor in a team winning any given night.

Despite not successfully predicting this year's tournament we were able to learn some valuable statistical techniques that may be useful in future studies. Overall, the success our models achieved combined with the new statistical knowledge accumulated has made this project a fruitful endeavor.

References

Hastie, Trevor, et al. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2017.

NCAA Bracket for March Madness. National Collegiate Athletic Association, www.ncaa.com/march-madness-live/bracket.

Men's College BASKETBALL POLLS. Fox Sports, www.foxsports.com/college-basketball/polls?season=2019&poll=1.

2021 College Basketball STANDINGS: College Basketball. Associated Press, collegebasketball.ap.org/standings/2021/d1.

ESPN, ESPN Internet Ventures, www.espn.com/mens-college-basketball/statistics/team/_/stat/scoring/sort/points.

2021 Pomeroy College Basketball Ratings, Ken Pomeroy, kenpom.com/.

Sklearn.linear_model.Lasso, Scikit, scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html.