Exploring the Dynamics of College Basketball

Halloran, Slivinsky, Waterhouse

June 2023

Contents

Abstract

We explore the most important statistics in making a winning basketball team and how these change over time. Our data set contains college basketball game data which we clean and modify into a data frame of team season box score statistics data. PCA and clustering methods reveal what features seperates teams of similar or differing performance. PCA shows that team performance is centrally distributed, so we perform outlier detection to identify points of high variation. To see how the importance of box score statistics change over time, we perform season by season regression on various subsets of features to track feature coefficient values per season. The visualizations of these feature coefficient values per season reveal that the statistics most important to winning games have remained relatively constant over the last 21 years.

1 Introduction

It is hard to quantify what makes a winning basketball team. Over the years there have been multiple shifts in the dominant strategies throughout professional and collegiate leagues. For example, up until the early 2010s, many of the best teams focused on scoring close to the basket and almost always had a dedicated center, but more recently teams have started to focus more on three point shots and sometimes play without a center. We decide to explore if readily available box score data captures changes like these and ask: what are the most important statistics in terms of winning and losing basketball games and how do these change over time? It will be interesting to see if there are certain constants that basketball teams have always needed to win, or if these evolve with the game.
We start in section 2 by introducing our chosen dataset and discuss how we modify it to make it easier to work with. In section 3 we state the different dimensionality reduction, clustering, outlier detection, feature selection, and regression techniques we use on our modified data. Section 4 presents and discusses the results. Finally, section 5 wraps up our paper and states our conclusions.

2 Data

2.1 Overview of Dataset

Our data comes from the Kaggle March Machine Learning Mania 2023 competition dataset [Jef23]. We mainly use the `MRegularSeasonDetailedResults.csv` portion of the dataset which contains box score data from the 2003 - 2023 men's regular seasons. There are 107,634 game entries (rows), and 34 features (columns). Within these features, we focus on the 14 game related statistics:

```
["Score", "FGM", "FGA", "FGM3", "FGA3", "FTM", "FTA", "OR", "DR", "Ast", "TO", "Stl", "Blk",
"PF"].
```

For each game, each of these features are included for both the winning and losing team, so each game has 28 features.

2.2 Modifying the Dataset

To quantify team performance in a single season, we went through the game data and collected the season average box score stats for each team and their opponents. To compress the information provided by a team's and a team's opponents' average statistics in a season into an interaction between the two, we take these values and create the Ratio of Average Team Statistic (RATStat):

$$\text{RAT(Stat)}_{ij} = \frac{\text{Team Average (Stat)}_{ij}}{\text{Opponent Average (Stat)}_{ij}} \tag{1}$$

$$i = 2003, \ldots, 2023(SeasonID) \tag{2}$$

$$j = 1, \ldots,\ 350(TeamID) \tag{3}$$

A concrete example of a ratio statistic is:

$$\text{RATFGA}_{ij} = \frac{\text{Team Average Field Goal Attempts}_{ij}}{\text{Opponent Average Field Goal Attempts}_{ij}} \tag{4}$$

It follows that if a team's season has a ratio statistic that is greater than 1, that means they have more of that given stat on average than their opponent. This modified dataset that we use consists of 7,255 team-seasons with 14 RATStat features as well as a few identifying features such as `TeamID`.

3 Model and Methods

3.1 PCA and Clustering

In order to reduce the dimensionality of our data we perform PCA on the 14 RATStat features and analyze the first 2 principal components. This seeks to identify separation between team performance across teams and seasons. We apply various clustering algorithms to the dimensionality-reduced data to identify groups of teams clustered by their end of season performance. Along with determining which clustering algorithm is best for our data, our goal with clustering the data is to group together teams of similar performance, and to find interesting dynamics in determining team success. Our success with PCA in finding separation with regard to the end of season outcome, games won, leads us to explore the dynamics of the importance of different statistics over the seasons.

3.2 Outlier Detection

To find outliers in our centrally distributed data, we use the outlier detection algorithm `IsolationForest` from scikit-Learn [Ped+11]. We apply this algorithm to both season average data, as well as the original game data with the intent to find games and seasons that stand out the most. Before performing the Isolation Forest on our season data, we perform PCA and then run the algorithm on the first two principal components. For the season data, we choose not to perform PCA before outlier detection and, instead, run the algorithm on the box score features. We do run PCA on the game data later but this is just to reduce the dimensionality in order to visualize it on a 2-D plot. One important parameter of `IsolationForest` that we utilize is contamination. This effects what proportion of the data the algorithm assumes will be outliers; the higher the contamination parameter, the higher the number of outliers will be found.

3.3 Regression

We explore the dynamics of the importance of box score statistics by training three distinct regression models over each season individually and then analyzing the covariate coefficients to see how they change per season. Every model has games won in a season as the output, but they differ in input:

- The first model performs regression on all 14 RATStat features,
- The second model performs feature selection on each individual season to feed in the three best features not including RATScore into the regression
- The third model performs feature selection on each individual season to feed in the two best features not including RATScore or any of the RATMakes into the regression

We find the"best" features through scikit-learn's `SequentialFeatureSelector` [Ped+11]. We run the feature selector for each number of features $1, ..., 14$ and then choose the minimum number of features at the elbow point before the MSE starts to drastically increase. This gives us an easily interpretable number of features and a low error for that season.
For each of these three models, we use 4 different linear regression methods: `LinearRegressionCV`, `LassoCV`, `RidgeCV`, and `ElasticNetCV` which use scikit-learn's built in 5-fold cross validation to find the optimal alpha parameter [Ped+11].
After running all three models for all 21 seasons, for each model we have all RATStat covariate coefficient values for each regression method for each season. In order to visualize the changes in these values, we first calculate the proportion that each coefficient contributes to the output. For example, if we let i denote the season, m denote the current model, and p_m denote the number of features in model m:

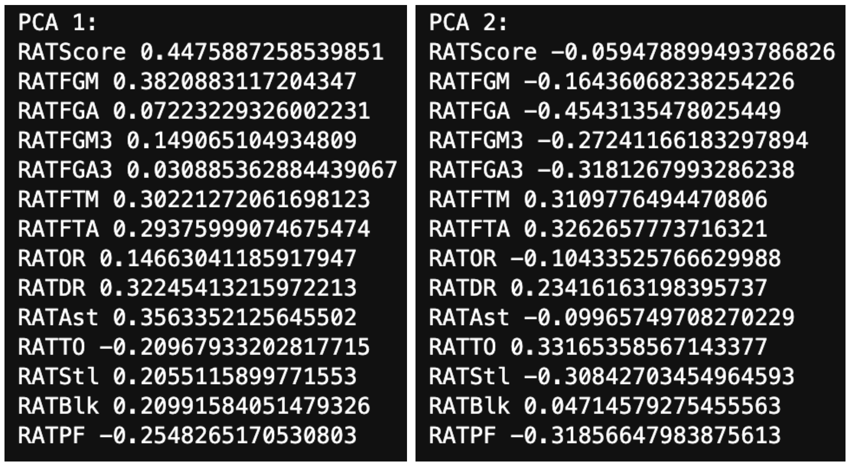
$$\text{Proportion(RATStat)}_{i,m} = \frac{|\text{Coefficient of RAT Stat}|_{i,m}}{\sum_{j=1}^{p_m} |\text{Coefficient}_j|_{i,m}} \tag{5}$$

Where the denominator is the sum of the absolute value of all 14 coefficients of the 14 ratio stats. We then scale these proportions so the sum of the absolute value of all of the coefficients is always equal to one, which makes the proportions into percentages. This allows us to create stacked area plots to show how the percentages of the RATStat covariates change over each of the 21 seasons.

4 Results

4.1 PCA and Clustering

The loading components of the first two principal components are shown in figure 1 and the cumulative variance of all of the principal components is shown in figure 2. We see that 60% of the variance between team's seasons is described by the first two principal components meaning that these are both crucially important in describing team performance. In the first principal component, all of the statistics that relate to a beneficial outcome for a team such as field goal makes and assists have positive components and the only negative statistics are turnovers and personal fouls which are detrimental to a team. We interpret this to mean that principal component 1 is a direct measure of team performance. Principal component 2 is a little bit harder to interpret; the positive components are the two free throw related statistics, as well as defensive rebounds, turnovers, and blocks. Turnovers stick out from this list because it means losing possession of the ball which is a very negative statistic. We therefore interpret principal component 2 as separating teams of



```
PCA 1:
RATScore 0.4475887258539851
RATFGM 0.3820883117204347
RATFGA 0.07223229326002231
RATFGM3 0.149065104934809
RATFGA3 0.030885362884439067
RATFTM 0.3022127206169812З
RATFTA 0.29375999074675474
RATOR 0.14663041185917947
RATDR 0.32245413215972213
RATAst 0.3563352125645502
RATTO -0.20967933202817715
RATStl 0.2055115899771553
RATBlk 0.20991584051479326
RATPF -0.2548265170530803
```

```
PCA 2:
RATScore -0.059478899493786826
RATFGM -0.16436068238254226
RATFGA -0.4543135478025449
RATFGM3 -0.27241166183297894
RATFGA3 -0.3181267993286238
RATFTM 0.3109776494470806
RATFTA 0.3262657773716321
RATOR -0.10433525766629988
RATDR 0.23416163198395737
RATAst -0.09965749708270229
RATTO 0.33165358567143377
RATStl -0.30842703454964593
RATBlk 0.04714579275455563
RATPF -0.31856647983875613
```

Figure 1: The loadings of our first two components.

similar performance.

We plot the first two principal components in figure 3. The separation into horizontal groups (color coded by number of games won in a season), shows that PC 1 does in fact measure team performance and separate good from bad teams. The variation that is seen along the vertical axis supports that PC 2 shows variation in teams of similar performance.

We attempt to extract more information from this principal component plot by trying various clustering algorithms shown in figure 4. Because our data is very centrally distributed, we do not find strongly interpretable separation within it using clustering. However, we can still see some interesting results despite the lack of clarity in the cluster separations. One interesting result is `MiniBatchKMeans` (1st row, 5th column of figure 4). The results obtained from this clustering algorithm stands out from the rest as its clusters are separated by distinct straight lines besides a round cluster in the center of the data. Taking a look at the normal `KMeans` algorithm (1st row, 1st column of figure 4), we can see that it similarly separates its clusters however the results that `MiniBatchKMeans` finds are more easily interpreted. One interpretation of this plot could be that teams are clustered by being good at offense and defense in the top right, bad at offense but good at defense in the top left, average in the middle, bad at offense and defense in the bottom right, and bad at offense but good at defense in the bottom right.

Another interesting result from our cluster plots is `DBSCAN` (2nd row, 4th column of figure 4). This plot separates the main centrally distributed portion of the data from some outliers on the fringes which leads us to the next portion of our clustering: outlier detection. While these results from `DBSCAN` potentially could be used for detecting outliers within our data, as it seems like it performed well, we use a dedicated outlier detection method instead.

4.2 Outlier Detection

We use `Isolation Forest`, an outlier detection algorithm, on our data to see if it can provide further results than `DBSCAN`. We show a contamination rate of 1% in figure 5. To focus on only one individual game and season from our data, we decreased the contamination parameter down to only receive 1 outlier from each dataset. Figure 6 shows the single outliers that we obtain from this low value of contamination. We find interesting outliers using this method. For the season data, `Isolation Forest` finds the 2008-2009 University of Connecticut team that went 27-3 in the regular season. It is important to note, that they finished the postseason with 31 wins and 5 losses and made it to the final 4 in the NCAA tournament. Looking up their season box score statistics, they ranked high among all of the statistics that were important

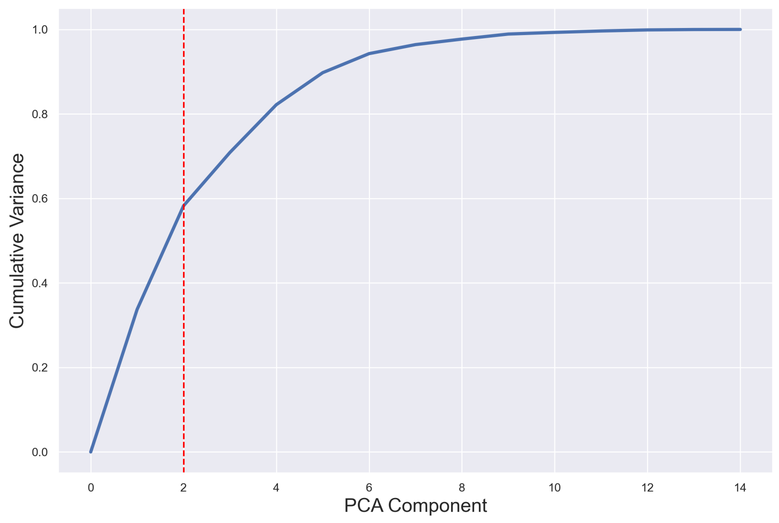


Figure 2: The cumulative variance of PCA components (blue) with a line showing how much variance is explained by the first two components (red). The first two components explain almost 60% of the variance.

to PC1 and PC2, which is why we can look at the left plot on figure 6 and see that they are the data-point furthest in the top right. For the game data, which is the left plot on figure 6, the outlier we find is a game that also included the same 2008-2009 University of Connecticut team. In this game, the final score was 117–127 where Connecticut was defeated by Syracuse University in 6 overtime periods. This is one of 4 games in Men's college basketball history that a game has gone into at least 6 overtimes.

4.3 Regression

Although reducing the dimensionality of data results in one centrally-distributed clump, we can use the RATStats to predict end-of-season outcomes quite well. We first develop a model that predicts the number of games won at the end of the season as a regression on all of the RATStat features, and we show the predictions for one team, Duke, in figure 7. This is a proof of concept to show that regression methods work well at predicting games won in a season with the RATStat features as input. We now investigate how important each of the RATStat features are towards the number of games won for a team in a season.

The stacked area plot of our first model's RATStat percentages by season (14 features) is shown in figure 8. This is a very information dense plot. One could select any two seasons and compare their percentages to see how the formula of winning changes in those seasons. One could also select any two of the linear regression methods, like Lasso and Ridge, and compare their percentages over a season For example, the Lasso and LinearRegression plots in figure 8 show a massive decrease in the importance of RATScore in 2016 compared to the previous year. We could also look at the different regression methods and compare and contrast what they view as important. Many of the RATStats are correlated with each other, and in fact are all pretty correlated with RATScore, and Ridge and ElasticNet perform better on data that is correlated with itself. For example, an assist is passing the ball to someone who then makes a shot so RATAst and RATFGM are very related. This means that we should trust our Ridge and ElasticNet methods more than LinearRegression and Lasso. Because we are trusting the former group of methods more, this leads us to believe that the importance of different statistics has remained relatively constant over the seasons. This plot also shows that Lasso heavily favors RATScore meaning it views it as the most important feature. This is an obvious result because teams that score more than their opponents will win more. We want to find less obvious results than this which brings us to explore our next model.

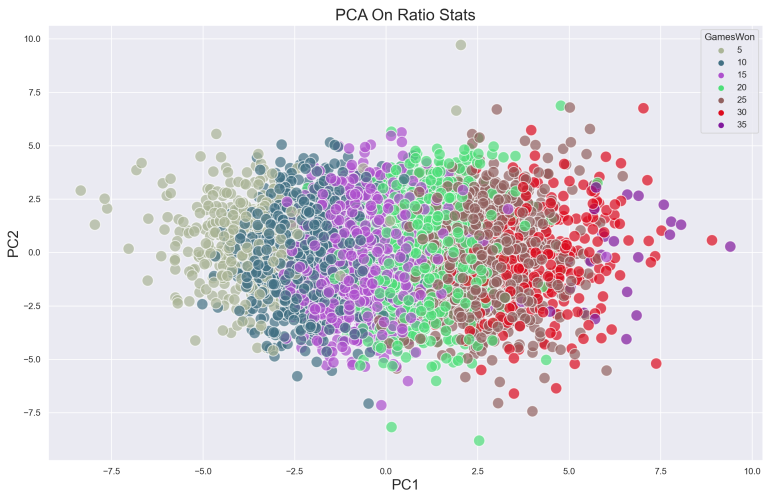


Figure 3: The first and second principal component for each team's season plotted against each other. The coloring corresponds to games won in a season. The PC 1 separates teams by games won and PC 2 separates teams of similar performance.

For our second model, we remove RATScore from the input data. We then perform sequential feature selection to find the three best features for each individual season, which turn out to consistently be: RATFTM, RATFGM3, and RATFGM. We train our models season by season with our selected features and produce figure 9. These results are much more consistent with all regression methods finding similar dynamics and the importance of the statistics themselves remaining relatively constant. This plot does offer an interesting insight on three point shooting. In 2015 the Golden State Warriors won the NBA championship led by Stephen Curry, arguably the greatest three point shooter in history, who won the MVP award that season. This caused an influx of basketball teams, both professional and collegiate, increasing their three point attempts. However, we see in figure 9 that the plots do not show an increase in the importance of RATFGM3 which is unexpected. Of course, the three selected features for this model all directly contribute to scoring so, once again, we have an idea for another model.

For our third model we keep RATScore removed, but also remove all three of the RATMakes stats which were selected previously. We then perform feature selection on each individual season and find that the two best stats are consistently: RATTO and RATDR. We are surprised to find that just using these two non-scoring features, we produce a model with relatively low error. For example, the average MSE from running Lasso with the two feature input on all seasons individually is only about 6.60 which isn't much higher than the average MSE of the all feature model which is about 4.10. To investigate the dynamics found by this two feature model, we train this model on each season and produce the plot seen in figure 10. This figure is similar to figure 9 with the result being relatively constant between models and seasons. We do see a slow rise in the importance of defensive rebounds and, consequently, a slight decrease in the importance of turnovers. Both of these stats are a measure of stopping an opponents scoring opportunity, and generating one for your team. In general, this plot enforces the classic idea that a good defense makes a good offense.

In order to compare the models as a whole, we average the MSEs over each individual season and then plot them in figure 11. The all feature model performed the best, the three feature model without score was next, and the two feature model without score performed the worst. However, the three and two feature model did not perform much worse than the all feature model and their results are much easier to interpret.

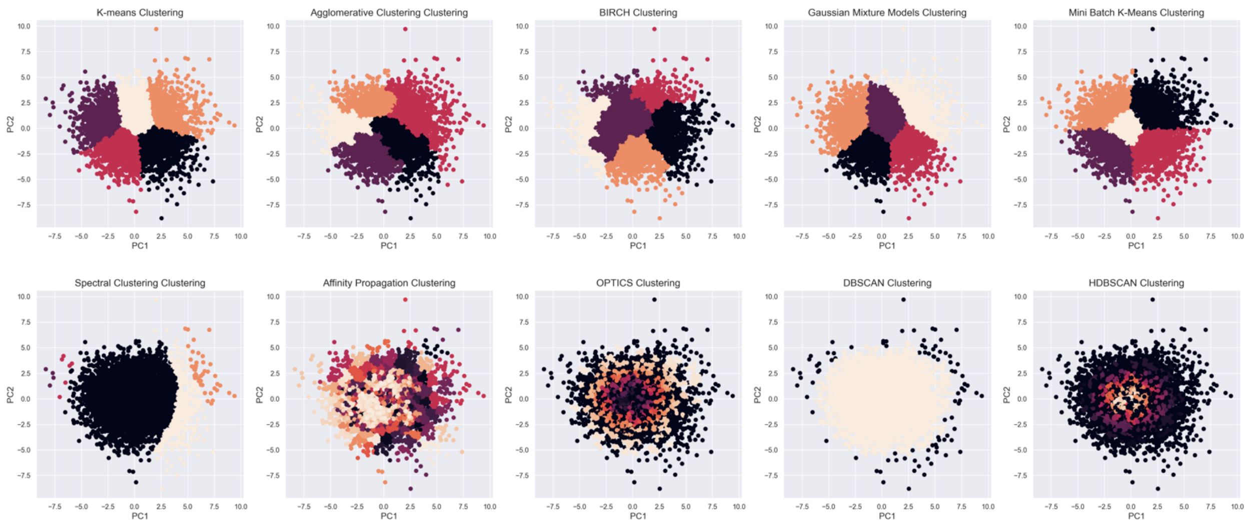


Figure 4: The outcomes of numerous clustering algorithms on our plot of PC 1 and PC 2.

5 Discussion and Conclusion

Basketball is a very complex and fast-paced game, and we found that deriving insight using machine learning algorithms from box score statistics, summary statistics of the dynamics within the game, proves difficult. However, even with only summary data, we were able to find multiple insights. Using PCA, we were able to find what separates winning from losing teams with principal component 1, and how teams of a similar skill level vary between each other with principal component 2. The data produced by PCA was very centralized so clustering algorithms were not able to find much insight besides finding extreme outliers which we pursued with `IsolationForest`.

To build on the success of PCA, we sought to find out how the statistics that most influenced winning teams changed over time. We derived three interesting models with different tradeoffs and trained them season by season to get covariate coefficient data which was our metric for the importance of different statistics in a given season. The third of these two models which only took in turnovers and defensive rebounds, two non scoring stats, as input surprised us in it's efficacy at predicting games won in a season. Because of how different modern day basketball appears to be from basketball even 10 years ago, we expected there to be dramatic changes in the importance of different stats over time. Our linear models told a different story: we found that the importance of certain fundamental stats has remained relatively constant over the years.

References

[Ped+11] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[Jef23] Will Cukierski Jeff Sonas Maggie. *March Machine Learning Mania 2023*. 2023. URL: `https://kaggle.com/competitions/march-machine-learning-mania-2023`.

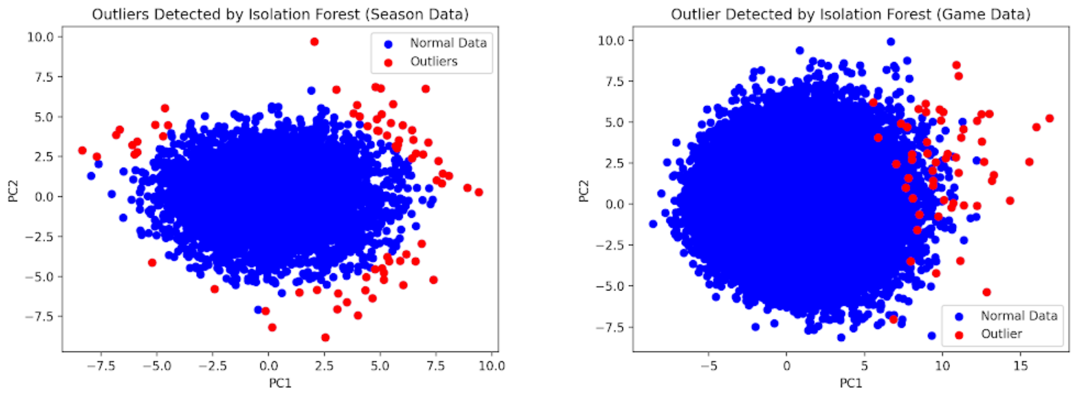


Figure 5: The first two principal components of both season and game data plotted against each other with normal data in blue and outlier data in red. A high contamination value was used in isolation forest resulting in multiple outliers being found. Note that we ran outlier detection on the original games data and not the PCA components; the PCA plot of game data is just for visualization.

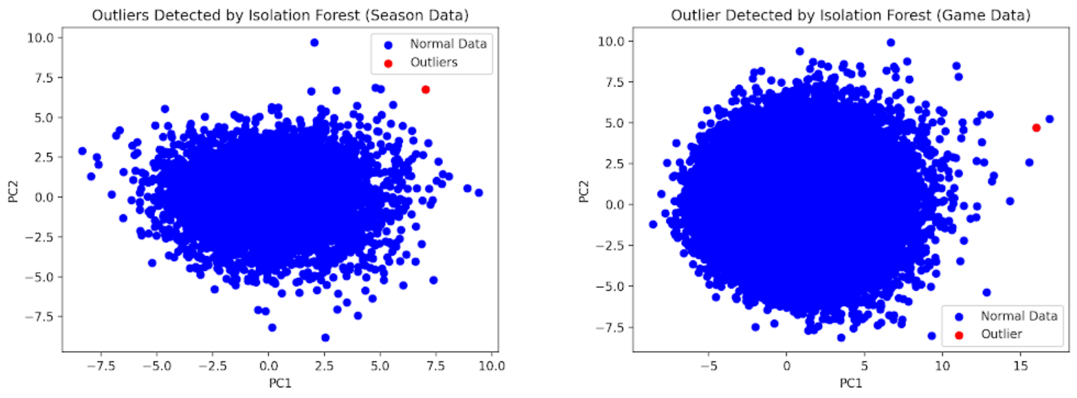


Figure 6: The first two principal components of both season and game data plotted against each other with normal data in blue and outlier data in red. A low contamination value was used in isolation forest resulting in only one outlier for each dataset being found. Note that we ran outlier detection on the original games data and not the PCA components; the PCA plot of game data is just for visualization.

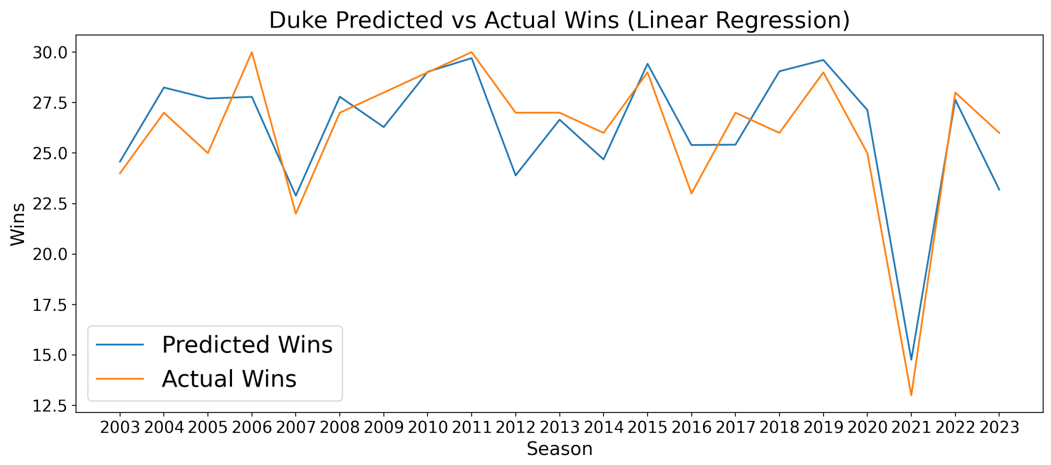


Figure 7: The predicted and actual wins of Duke for the 2003-2023 seasons taking all ratio statisticss as input to standard LinearRegression. Overall model MSE of about 5.16 .

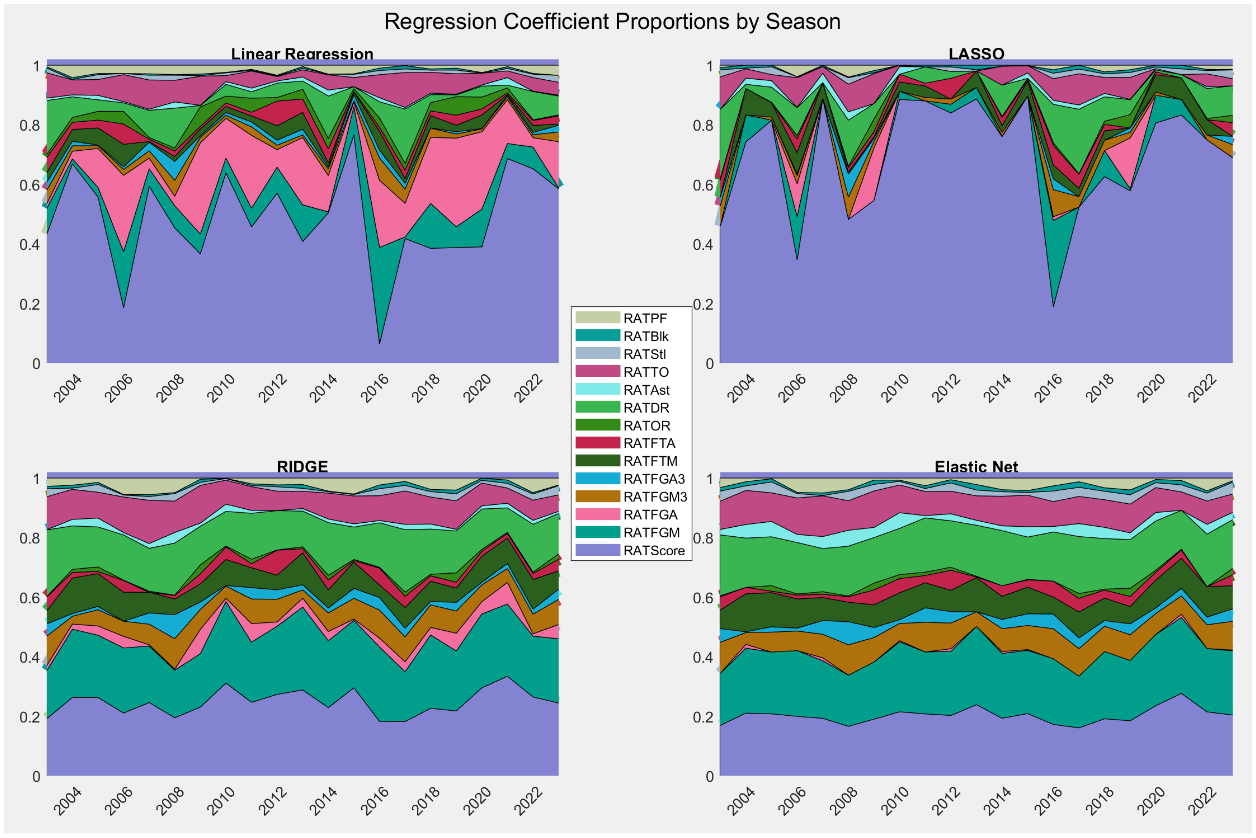


Figure 8: The stacked area plot of model 1 showing the dynamics of feature importance over the seasons. The legend lists the order of the colors from top to bottom showing that RATPF is the top-most color in the stack and RATScore is the bottom-most.

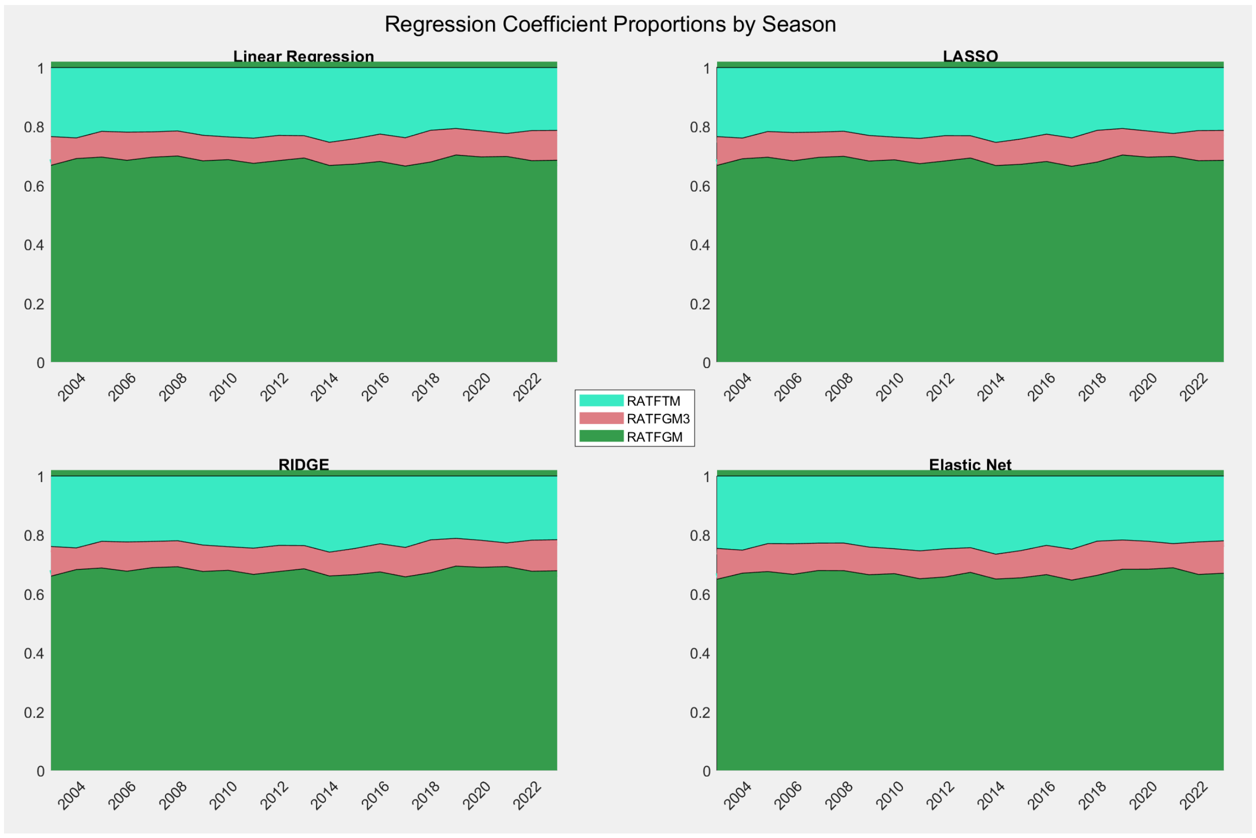


Figure 9: The stacked area plot of model 2 showing the dynamics of feature importance over the seasons.

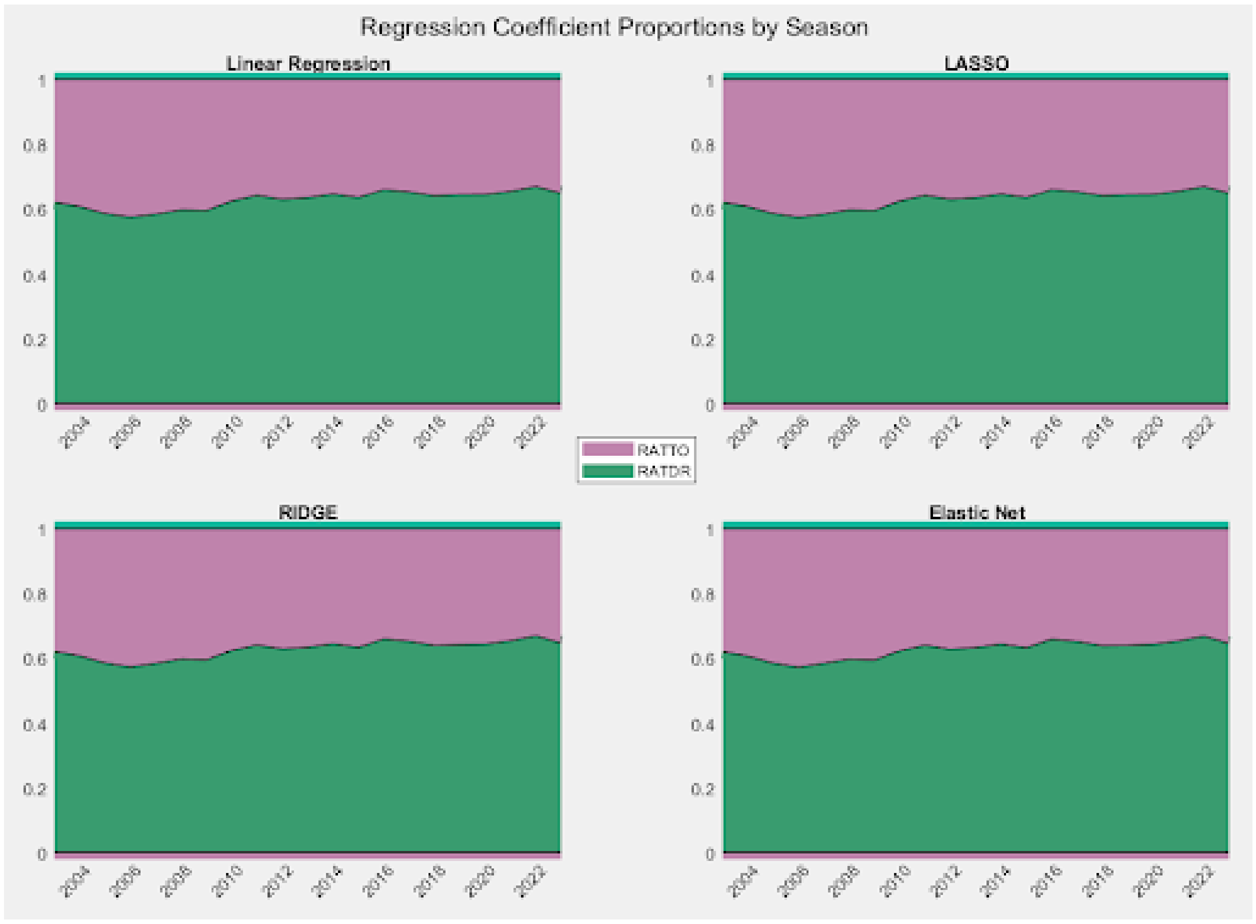


Figure 10: The stacked area plot of model 3 showing the dynamics of feature importance over the seasons.

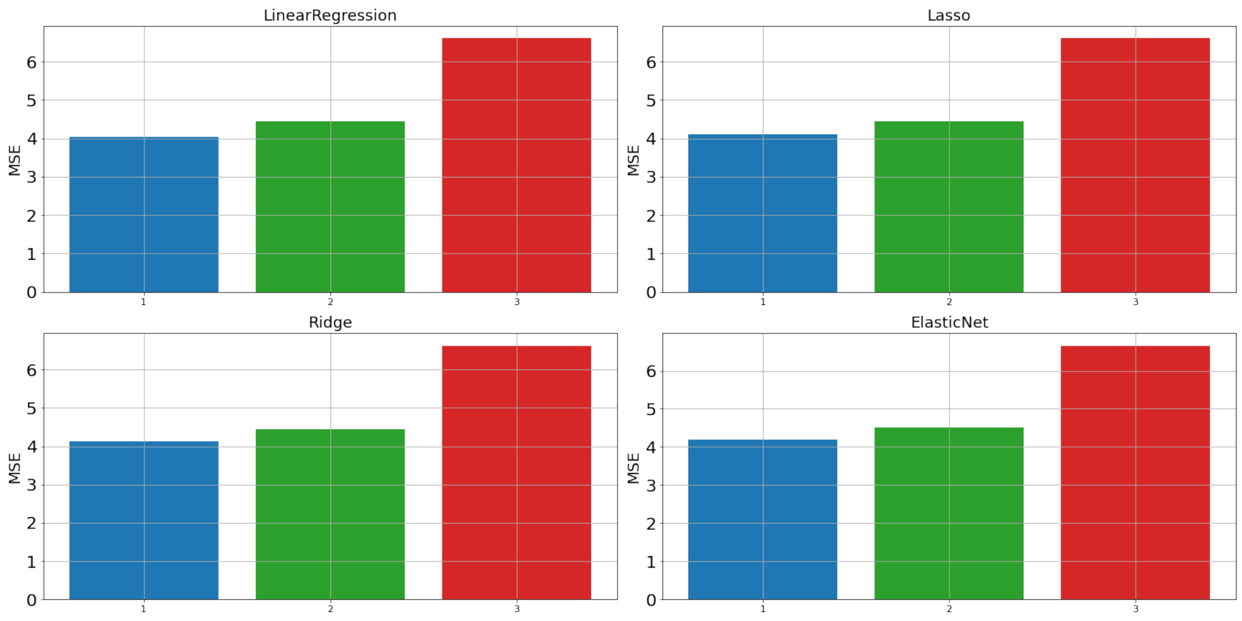


Figure 11: The average MSE of our three different models. The 14 feature model is Blue, the three feature model without score is Green, and the two feature model without score or makes is Red.