Final Report

Team 8

Our team consists of Casey Pridgen (ccp2uz), Colin Dowd (ccd4xc), Xin He (ubn4am), and Sami Adam (zeh4tv)

Section 1: Regression Question Executive Summary

In the world of professional basketball, both teams and players are constantly looking for an edge. Teams want to win championships, and players want to maximize their earning potential. One crucial factor in achieving these goals is understanding which player skills translate to higher salaries. This study dives into this question by using statistical methods to identify the key statistics that have the biggest impact on a player's salary.

This research holds significant value for both parties. Team managers are often tasked with setting fair salaries during contract negotiations. Traditionally, this process might involve a mix of experience and intuition. Our study offers a more objective approach. By analyzing player statistics, teams can create a data-driven framework for evaluating player performance and determining appropriate compensation. This not only helps ensure fairness in negotiations but also allows teams to allocate their resources strategically.

Players can also benefit greatly from understanding the link between statistics and salaries. By knowing which skills have the strongest correlation with higher earnings, players can optimize their training and development efforts. Focusing on these areas will not only improve their overall game but also position them more competitively as an athlete. This empowers players to negotiate from a position of strength and ultimately maximize their earning potential throughout their careers.

Our analysis shows that the most crucial factors are Age, Minutes Played per Game, Value Over Replacement, Steals per Game, and Field Goals per Game. For team managers, the recommendation is to develop a comprehensive player valuation framework that heavily weighs the key statistics identified by our analysis. This could involve creating a formula or algorithm that calculates a player's projected value and appropriate salary level based primarily on their Age, Minutes Played, Value Over Replacement, Steals, and Field Goals statistics. Managers should train their scouting and analytics departments to focus on these areas when evaluating prospects and current players. During contract negotiations, managers can use data-driven arguments anchored on these key metrics to justify their salary offer as a fair market value for the player's production.

As for players, they should work closely with their individual skills coaches, strength/conditioning staff, and shooting specialists to design rigorous training regimens focused on increasing their Minutes Played per Game through improved conditioning, boosting their Value Over Replacement by refining overall offensive and defensive execution, racking up more Steals per Game through intense defensive technique work, and upping their Field Goals per Game by dedicating practice time on shooting form and accuracy. Players should regularly track their progress on these key metrics and set ambitious goals aligned with the level of salary they desire. With a concentrated development plan centered around the high-impact areas pinpointed by our analysis, players can maximize their chances of earning a lucrative contract.

By closely following the interpretable insights, both team managerial staff and individual players can adopt actionable, stats-driven strategies to more accurately determine fair market value and drive better negotiation results. Ultimately, implementing these recommendations allows all parties to make more informed, data-backed decisions in the critical area of player compensation.

Section 2: Regression Data and Variable Description

We obtained the data set from Kaggle. The data set is the result of web scraping player salary information from Hoopshype, and downloading traditional per-game and advanced statistics from Basketball Reference. It looks at one year's worth of data from the NBA. Below, we have a data description of all the variables used in the exploratory data analysis and model sections.

Data Preprocessing

Before analyzing the data, we found that we needed to implement modifications to our data set in order to make our graphs more meaningful. It is important to mention that we applied these transformations to the entire data set, but we will be referencing the value counts we got in the training data set. First, we filled N/A values to equal 0. The certain variables, such as True Shooting Percentage (TS%) contain N/A values. Since the percentage variables measure success over total attempts, some observations are N/A due to the fact the player did not attempt the metric. For example, certain players did not take any shots. This leads to the player to have an undefined TS%. Setting the N/A value to 0 for TS% makes sense in this context because if a player did not take any shots, their true shooting efficiency for that season was effectively 0%. They did not score any points per shooting possession.

Second, we created the categorical variables Starter and Position Group. While exploring the data, we determined that it would be important to see whether being a starter impacted salary. In order to create the variable Starter, we looked at the Games Started (GS) and Games Played (GP) variables. If the player started more than half the games they played, they received a 1. If they didn’t start more than half, they received a 0. As seen in Figure 1, the levels are slightly imbalanced. This is caused by the fact that there is a small proportion of starters in the NBA.

0	1
162	71

Figure 1: Starter Frequency Table (0: Non-Starter, 1: Starter)

The original categorical variable Position has high levels of class imbalance, which impacted both our EDA and model building process. More specifically, the levels PG-SG, SF-PF, SF-SG, and SG-PG have less than three observations each, as seen in Figure 2. In order to mitigate that, we created a new variable Position_Group that consolidated all the levels of the positions into three new levels: “Center”, “Guard”, and “Forward”. So now, the level of granularity is greatly reduced, meaning that the levels are more balanced. The results of the consolidation are shown in Figure 3. While this is a big improvement, it is clear that there are less Centers in the training data set. Moreover, Centers make up around 20% of the sampled players. This is due to the fact that the Center level was not consolidated.

C	PF	PG	PG-SG	SF	SF-PF	SF-SG	SG	SG-PG
50	49	35	1	39	1	2	55	1

Figure 2: Position Frequency Table

Center	Forward	Guard
50	89	94

Figure 3: Position Group Frequency Table

Third, we transformed the response variable Salary. Since salaries are normally skewed right, we decided to take the logarithm of the variable. In doing so, we lessen the magnitude of difference between the higher values, effectively stabilizing the variance. As seen below, the density plot of the logarithmic transformation is more normal. Although it is not perfect, it is a lot better than the original response variable.

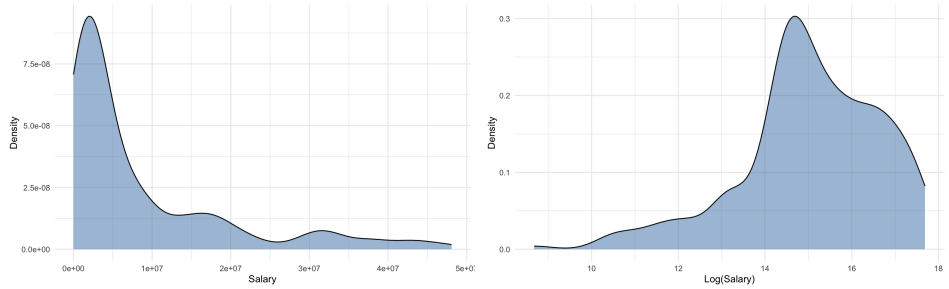

Figure 4: Pre-transformed Response Variable (Left) vs Transformed Response Variable (Right)

Lastly, to address the issue of multicollinearity, several variables were removed from the dataset as they were highly correlated with or redundant to other variables. Variables related to specific shooting categories like Three-Pointers Made/Attempted, Two-Pointers Made/Attempted, Free Throws Made/Attempted, and their corresponding percentages were removed. Instead, the more comprehensive True Shooting Percentage (TS%) was retained as it accounts for all shooting categories. Additionally, Field Goals Made (FG) was kept as a concise measure of scoring efficiency.

Redundant variables like Free Throw Rate (FTr), Offensive and Defensive Rebounds (ORB, DRB), Assist Percentage (AST%), Total Rebound Percentage (TRB%), Steal Percentage (STL%), Block Percentage (BLK%), Turnover Percentage (TOV%), Usage Rate (USG%), Offensive Win Shares (OWS), Defensive Win Shares (DWS), Win Shares (WS), Offensive Box Plus-Minus (OBPM), and Defensive Box Plus-Minus (DBPM) were also removed. These variables were correlated with or could be derived from other more useful explanatory variables like True Shooting Percentage, Total Rebounds, Player Averages, Win Shares per 48 Minutes, and Box Plus-Minus.

The final set of variables retained are described below.

Data Table

Variable Name (Full Name)	Description	Variable Type
Salary: Quantitative Response Variable	The salary a player is paid	Quantitative
Starter	Determines whether a player started more than half the games they played. Derived from GP and GS	Categorical - Levels: 0 (Non-Starter), 1 (Starter)
Position	Lists the position a player plays, as well as potential secondary positions (if there is a secondary position, it is after “-”)	Categorical - Levels: C, PF, PG, PG-SG, SF, SF-PF, SF-SG, SG, SG-PG"
Position Group	Lists overall positions a player plays. It is derived from Position	Categorical - Levels: Center, Forward, Guard
GP(Games played)	Number of games in the 82 game NBA season a player has participated in	Quantitative
GS(Games started)	Number of games in the 82 game NBA season a player has started	Quantitative

MP(Minutes Played)	Average number of minutes a player plays per game	Quantitative
Age	Age of NBA player	Quantitative
FG(Field goals made)	The average number of field goals(two and three pointers) a player makes per game	Quantitative
TRB(Total rebounds per game)	The average number of offensive and defensive rebounds a player has per game	Quantitative
AST(Assists per game)	The average number of assists a player has per game	Quantitative
STL(Steals per game)	The average number of steals a player has per game	Quantitative
BLK(Blocks per game)	The average number of blocks a player has per game	Quantitative
TOV (Turnovers per Game)	The average number of turnovers per game	Quantitative
PF(Personal Fouls)	Average number of fouls a player commits per game	Quantitative
PER(Player Efficiency Rating)	This metric measures a player's per minute performance while adjusting for pace. A league average PER would be 15. It takes into account positive and negative counting stats into output.	Quantitative
TS%(True Shooting Percentage)	This weighs two pointers, three pointers and free throws differently to output a more accurate shooting percentage as degree of difficulty is accounted for.	Quantitative
3PAr(Three point attempt rate)	What percentage of a player's shot selection are three pointers.	Quantitative
WS/48(Win Shares per 48 minutes)	This is a player statistic which divides credit for team success to	Quantitative

	individuals on a team on a per game basis. It attempts to calculate a player's impact of them being on the floor in contributing to a win.	
BPM(Box Plus Minus)	This estimates a player's contribution in points above league average per 100 possessions played. This also combines two variables we previously looked at, defensive box plus minus and offensive box plus minus, into one all encompassing variable.	Quantitative
VORP (Value Over Replacement Player)	box-score estimate of the points per 100 team possessions that a player scores over a replacement/bench player translated to the average team over a full NBA season	Quantitative

Section 3: Regression Question

Exploratory Data Analysis

We first started the EDA process by looking at specific features. First, we checked the correlation among the quantitative features. As seen in Figure 5, there is multicollinearity among the predictors; however, we chose variables that did not overlap in context. The original data had multiple columns measuring the same statistics or performance measurements.

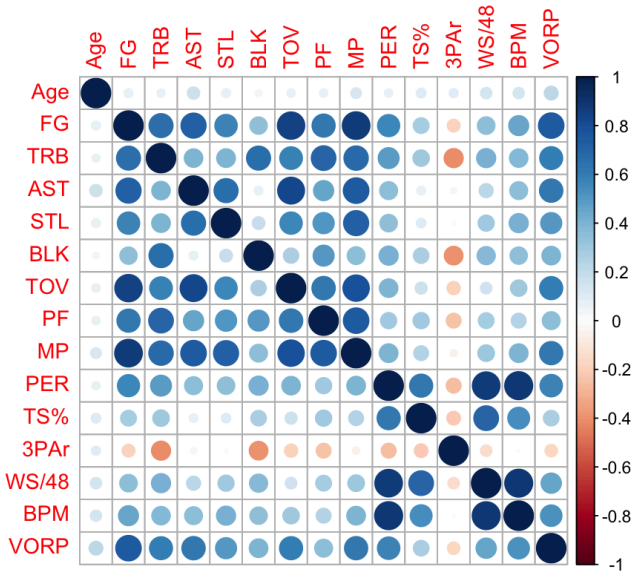

Figure 5: Predictor Correlation Matrix

We then checked the levels of the categorical variables Starter and Position Group. As seen in Figure 1 and 3, the newly created categorical variables are relatively balanced compared to their predecessors. However, the classes are still unbalanced. The Starter variable has 162 non starters vs 71 starters while the Position Group variable has 50 Centers, 89 Forwards, and 94 Guards. This can cause issues in the interpretability of the model, as the estimators could be biased.

Next, we looked at the variables in context of the response variable, Salary. First, we looked at the correlation of the quantitative variables vs the response variable. As seen in Figure 6, almost all the quantitative variables are positively correlated with Salary. This is not surprising as players who perform better through counting stats and advanced metrics are likely to be put in the rotation and earn a higher salary. It is surprising to see the variable Three Point Attempt Rate (3PAr) not being positively correlated to Salary. The lack of correlation between 3PAr and Salary is also surprising since the three-pointer has revolutionized the NBA, and three-point specialists are staple in team lineups. Players taking 80-90% of their shots from behind the arc would logically be better at them and paid more as three-point specialists. However, as seen in Figure 6, there is little to no correlation. These high 3PAr players, unless elite shooters, are also seen as replaceable, with varying skill levels, so their salary reflects that replaceability and variance.

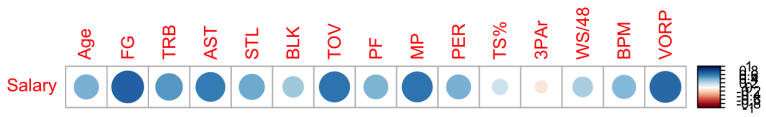

Figure 6: Correlation of Quantitative Predictors vs Response Variable

We then looked at boxplots of the categorical variables. Position Group had relatively similar distributions and medians across all levels. However when looking at the interquartile ranges and outliers, there seems to be a salary skew favoring guards and forwards over centers. When looking at the landscape of the NBA today this is not surprising, as when looking at a list of the top 20 players in the league, almost half of them would be classified as guards under our interpretation. NBA teams have begun to disregard players' natural positions to an extent and have the overall best player on the team bring up the ball to develop their skills more. This is why we see players like Luka Doncic, Lebron James, Zion Williamson, and even 7 foot players like Giannis play point guard either at all times or on certain possessions. A player can have a greater impact on the game if they have the ball in their hands more, so this makes sense that guards and forwards are paid more.

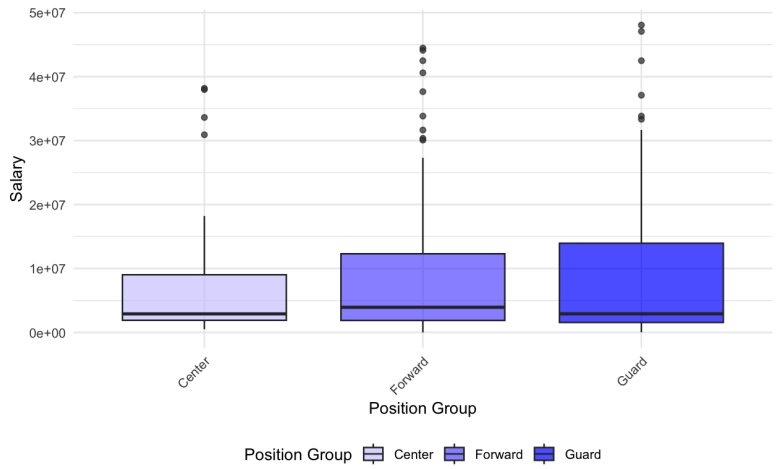

Figure 7: Boxplot Comparison of Position Group Levels vs Salary

The Starter variable, on the other hand, had a bigger difference between the levels. As seen in Figure 8, Starters salaries are generally higher than non-starters. This was expected, as starters command higher salaries than non-starters. This disparity is rooted in the fundamental value they bring to their

teams. Starting players are considered the backbone of a team's lineup, offering a combination of skill, experience, and consistent on-court performance that is highly prized by coaches and team executives.

Starters typically log more minutes on the court, providing a sustained impact throughout the game. Their roles are crucial in executing game plans, making key plays, and shouldering the responsibility of leading their teams to victory. As a result, coaches place immense trust in their starting five, relying on their ability to set the tone, establish offensive and defensive systems, and maintain a competitive edge against the opposition's best players.

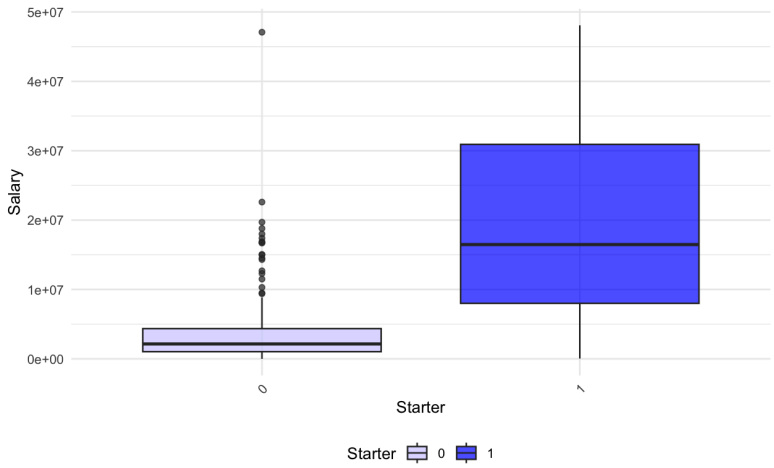

Figure 8: Boxplot Comparison of Starter vs Salary

Shrinkage Methods

Before model training, we implemented several modifications to our dataset to make our predictors more meaningful. We filled N/A values with 0 for all the variables where N/A represented players who did not attempt the variable. We created new categorical variables, Starter and Position GroupFinally, we log-transformed the skewed Salary variable to stabilize the variance and reduce the influence of extreme values. These transformations aimed to handle missing data, create relevant features, mitigate class imbalance, and improve the distributional properties of the response variable for more reliable modeling and analysis.

The final set of variables retained includes Salary, Age, Field Goals Made, Rebounds, Assists, Steals, Blocks, Turnovers, Fouls, Minutes per Game, Player Efficiency Rating, True Shooting Percentage, Three-Point Attempt Rate, Win Shares per 48 Minutes, Box Plus-Minus, Value Over Replacement Player, Position Group, and Starter. This selection aims to provide a comprehensive yet non-redundant set of variables for modeling and analysis.

Threshold

Before applying shrinkage methods to the data set, we first determined a viable threshold. We determined that value by comparing the coefficients of a ridge regression without a penalty with the

model above. We created a proxy error term by summing the squared difference between the coefficients, which is inspired by the MSE term. Our final threshold was 1e-13, with a proxy error term of 1.5 e-10. Below, we have the comparison of the two coefficients.

(Intercept)	14.24188392	14.24189293
Age	0.09811028	0.09811026
FG	0.26399822	0.26399698
TRB	0.05176424	0.05176353
AST	-0.20828856	-0.20829097
STL	0.77541546	0.77541464
BLK	-0.38017773	-0.38017808
TOV	-0.03342605	-0.03341828
PF	-0.47706553	-0.47706662
MP	0.08723610	0.08723637
PER	-0.12935814	-0.12935852
`TS%`	-0.80974171	-0.80975556
`3PAr`	-1.01892522	-1.01892735
`WS/48`	-1.34197423	-1.34192431
BPM	0.13686080	0.13686066
VORP	0.07779078	0.07779166
Position_GroupForward	-0.89013956	-0.89013921
Position_GroupGuard	-0.89316869	-0.89316801
Starter1	-0.08774288	-0.08774304

Figure 9: Lasso Coefficients vs OLS Coefficients

Lasso Regression

We fit a lasso regression onto the training data using the determined threshold. Based on our 10-fold cross-validation, we achieved an optimal lambda of 0.13. As seen in Figure 10, the estimated test MSE against the log of lambda is lowest below 0, meaning a lambda less than 1.

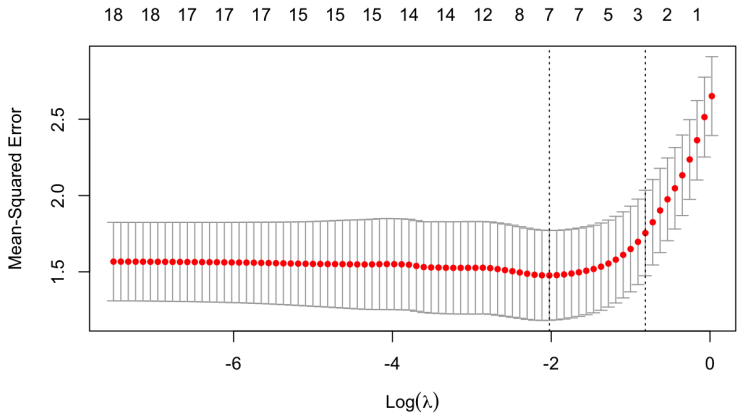

Figure 10: Log Lambda vs MSE

After choosing a suitable lambda, we fit a ridge regression to the data. Unlike Ridge Regression, Lasso can set coefficients to 0. This is due to the fact that the algorithm uses a l1 penalty. After fitting the model, we were left with 7 variables: Age, FG, TRB, STL, MP, VORP, and Starter. As seen in Figure 11, the model substantially shrunk the coefficients, setting some to equal zero.

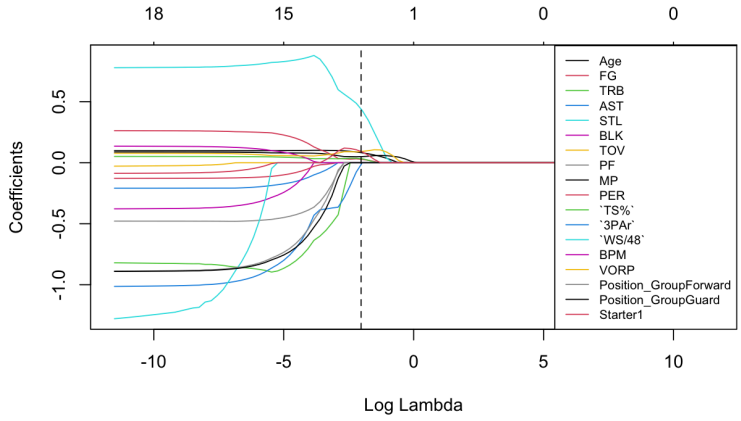

Figure 11: Log Lambda vs Coefficient Estimates (dash line represents optimal lambda)

The true test MSE value achieved by the ridge regression was 1.21.

Regression Trees

Recursive Binary Splitting

We used the same selected predictors as the Shrinkage Methods section to fit the regression tree. The final set of variables retained includes Salary, Age, Field Goals Made, Rebounds, Assists, Steals, Blocks, Turnovers, Fouls, Minutes per Game, Player Efficiency Rating, True Shooting Percentage, Three-Point Attempt Rate, Win Shares per 48 Minutes, Box Plus-Minus, Value Over Replacement Player, Position Group, and Starter. This selection aims to provide a comprehensive yet non-redundant set of variables for modeling and analysis.

We decided to fit a tree using Recursive Binary Splitting, as pruning resulted in a tree with the same number of terminal nodes, as seen in Figure 12. We used the variables listed above. As seen in Figure 13, the tree has 12 terminal nodes and uses the predictors VORP, STL, BLK, Age, TOV, PER, TS, MP, and FG. This regression tree helps us determine which predictors are the most important, and helps to visualize what standards the NBA player should reach to get a higher salary.

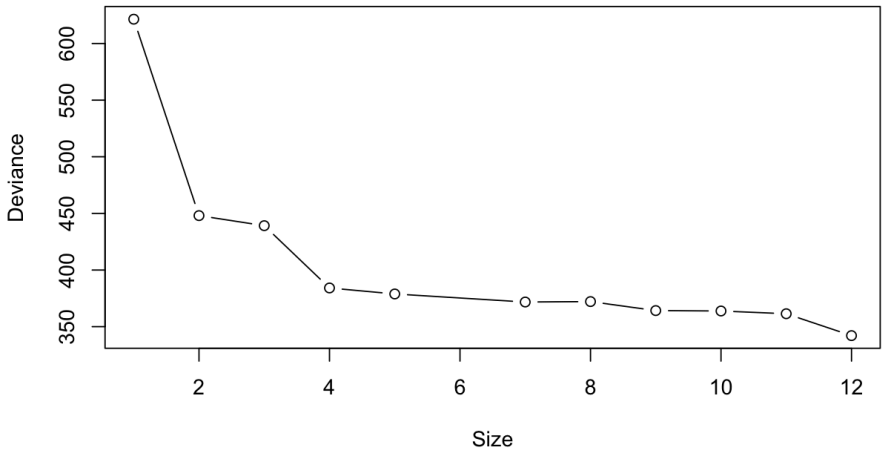

Figure 12: Tree Size vs Deviance

```
Regression tree:
tree(formula = Salary ~ ., data = train_rtree)
Variables actually used in tree construction:
[1] "VORP" "STL"  "BLK"  "Age"  "TOV"  "PER"  "TS"   "MP"   "FG"
Number of terminal nodes:  12
Residual mean deviance:  0.6644 = 146.8 / 221
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-2.46600 -0.50430 -0.02903  0.00000  0.44660  2.61400
```

Figure 13: Recursive Binary Splitting Output

The most important predictors are VORP, STL, and MP, as they are the first splits in the decision tree. In fact, MP is used multiple times, as seen in Figure 14. The decision tree answers our question of interest by graphically representing the most important factors that lead to being paid more. Through the graph, we can see that players who have a higher VORP, play more than 26 minutes a game, and are over 25 years old are expected to have the highest salaries. The test MSE using this tree is 1.33, which is slightly higher than our shrinkage methods.

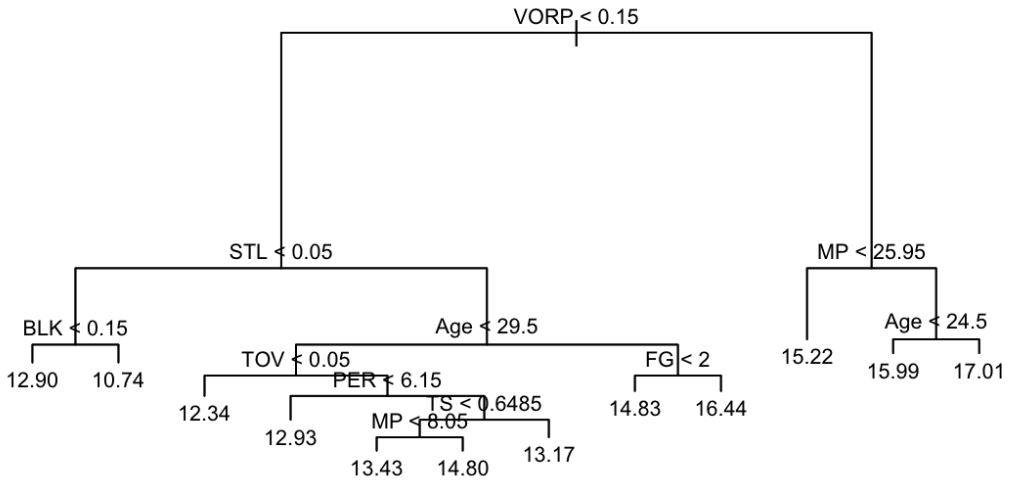

Figure 14: Recursive Binary Tree

Random Forests

We then attempted to improve the Recursive Binary Tree by using Random Forests. We decided to set mtry to 5, as it is 17 divided by 3 (the number of predictors) rounded down. That meant that the algorithm would try 5 predictors at each split. As seen in Figure 15, the most important predictors are Age, Minutes Played per Game, Value Over Replacement, Steals per Game, and Field Goals per Game. Although the two metrics measuring importance, percent increase in MSE and increase in node purity, reveal different predictors, they share common well performing predictors. The test MSE using Random Forests is 0.94, which is the lowest out of every regression problem.

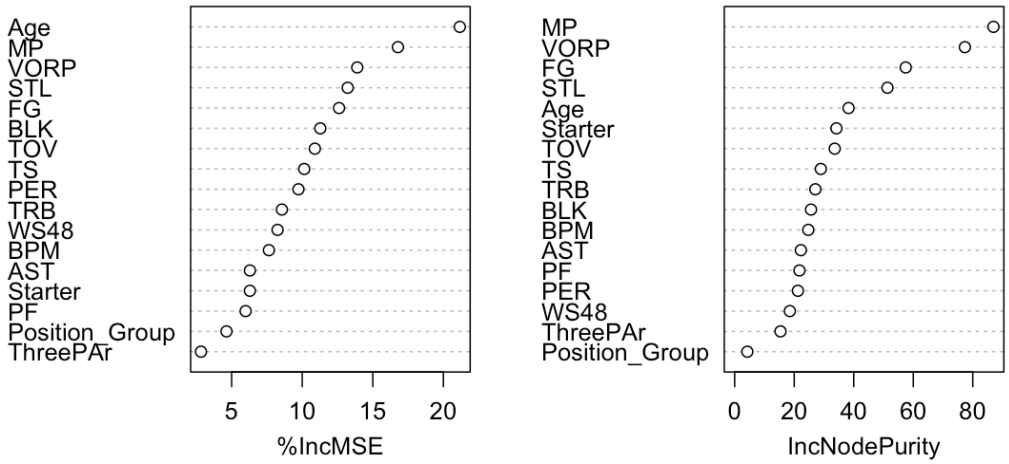

Figure 15: Random Forests Important Variables

Summary of Findings

Looking at the test MSEs of all the regression models, we see that RBS has the highest error rate, then Lasso and Random Forest follow. To our surprise, the Lasso model did not perform the best. We expected that to be the case because of the multicollinearity we saw during the EDA process. As mentioned before, Lasso is able to perform variable selection. We were not expecting the Random Forest model to drastically improve the RBS model.

Model <chr>	Test_MSE <dbl>
Lasso	1.21
Recursive Binary Splitting	1.33
Random Forest	0.94

Figure 16: Lasso, Recursive Binary Splitting, and Random Forest Test MSE Comparison

A key benefit of Random Forests is their ability to identify the most important features. This feature importance ranking makes Random Forests well-suited for interpretability. By analyzing which variables significantly impact the model's prediction, we can directly translate those findings into insights about player value. Random Forests don't just provide a prediction, but also reveal which statistics truly matter for a player's salary. This interpretability is crucial in this scenario. It allows us to move beyond a simple prediction to understand the "why" behind higher salaries. This knowledge is valuable for both team managers, who can make informed decisions about player compensation, and players themselves who can focus on developing the skills that have the greatest impact on their earning potential. The models reveal that the most important predictors are Age, Minutes Played per Game, Value Over Replacement, Steals per Game, and Field Goals per Game. One downside is that we do not know the directional weights of the variables. Unlike decision trees, we cannot output a graphical representation of the decisions. However, since the variables are relatively straightforward, it is safe to assume that having higher values for all the variables leads to a bigger salary.

The error rates in Figure 16 are in terms of the logarithmic transformation of Salary, as we transformed the response variable to achieve a more normal distribution. In order to get the error in terms of the original salary, we would have to exponentiate the predicted responses and test set response variables to recalculate the TMSE. Doing so reveals the TMSE; however, we believe it is more important to look at the RMSE. We calculated the RMSE of the Random Forest model and got a value of $6,811,367. That means that the model has a typical prediction error of $6,811,367. While this number looks big, it is important to compare it to the range of the response variable. Salary has a range of $5,849 - $48,070,014.

In order to better quantify the error, we took the error and divided it by the response variable range to determine the proportion of total range the error represents. Doing so gets us a ratio of 0.1417, or about 14%. This tells us that the typical error is equivalent to roughly 14% of the total range of salaries the model is trying to predict. In other words, while the error is not insignificant, it's a relatively small proportion of the overall salary range the model is attempting to capture. This suggests that the model is still useful for making predictions, despite the error.

Addressing Previous Comments

We addressed the previous comment about interpreting the TMSE in terms of the response variable and question context. The analysis can be seen in the Summary of Findings section.

Section 4: Classification Question Executive Summary

Unleashing a championship-caliber lineup hinges on picking the right starters. But defining what separates a game-changer from a benchwarmer can be an exhausting task. This study aims to crack the code on which player stats translate to starter material.

The question of interest that is being answered using classification methods is: What stats lead to players being overall starters? The goal of this question is inference - identifying which player statistics and skills make it more likely for them to be a starter and get more playing time. This is a valuable question to explore because the insights can benefit multiple stakeholders across the sport.

Teams and coaching staff would be interested in understanding the statistical drivers of being a starter. This information allows them to optimize their roster construction, player acquisition strategies, and development priorities based on the profiles that lead to more playing time. Teams would also want to identify and target the key stats that signal a prospect's readiness for a starting role.

In addition, players themselves are also highly motivated stakeholders, as they often seek clarity on which aspects of their game to focus on improving in order to increase their chances of being a starter and earning more minutes on the court. The ability to get consistent playing time as a starter is pivotal for a player's career development and value.

Through our analysis, we identified Turnovers per Game, Free Throw Percentage, Steals per Game, and Blocks per Game as the most significant factors influencing a player's starting status. Interestingly, the factors also directly contribute to team success on both ends of the court. Minimizing turnovers protects possessions and prevents transition opportunities for the opposing team. A high free throw percentage adds efficient and demoralizing points. Racking up steals disrupts the opponent's offense and creates extra transition chances. Blocking shots energizes teammates and weakens scoring threats. While these metrics evaluate individual contributions, they also impact team-wide performance in the crucial areas of scoring, ball control, defense, and momentum.

Moreover, these four variables are highly actionable from a coaching and developmental standpoint. They rely on technique, fundamentals, effort, and decision-making - all skills that can be drilled, reinforced, and improved through coaching. Teams can bring in specialist instructors to work on ball-handling for turnover reduction, shooting forms for free throw percentage, studying film to identify passing lanes for steals, and defensive footwork for protecting the rim with blocks. With a keen focus on honing these high-impact skills, teams can invest in increasing their overall talent level across an entire roster.

The key recommendation for teams and coaching staff is to concentrate their scouting, player evaluation, and developmental programs around these four critical statistics. For players themselves, the recommendation is to dedicate training towards improving in these high-leverage areas. By concentrating their development on these skills shown to be the strongest predictors of a starting role, players give themselves the best chance to earn a starter's minutes.

In summary, our analysis provides an analytical framework that allows teams to optimize their talent evaluation and acquisition strategies, coaches to make data-driven decisions on playing rotations, and players to strategically focus their training to increase playing time as starters. The actionable insights highlight Turnovers, Free Throw Percentage, Steals, and Blocks as the areas of maximum impact for securing a starting role. Implementing these recommendations will lead to smarter personnel decisions and more productive player development across the sport.

Section 5: Classification Data and Variable Description

Data Preprocessing

We created the categorical variable Starter. While exploring the data, we determined that it would be important to see whether being a starter impacted salary. In order to create the variable Starter, we looked at the Games Started (GS) and Games Played (GP) variables. If the player started more than half the games they played, they received a 1. If they didn't start more than half, they received a 0. As seen in Figure 17, the levels are slightly imbalanced. This is caused by the fact that there is a small proportion of starters in the NBA.

```
----------
  0    1
----- ----
 162   71
----------
```

Figure 17: Starter Frequency Table (0: Non-Starter, 1: Starter)

Data Table

Variable Name (Full Name)	Description	Variable Type
Starter: Categorical Response Variable	Determines whether a player started more than half the games they played. Derived from GP and GS	Categorical - Levels: 0 (Non-Starter), 1 (Starter)
Position Group	Lists overall positions a player plays. It is derived from Position	Categorical - Levels: Center, Forward, Guard

GP(Games played)	Number of games in the 82 game NBA season a player has participated in	Quantitative
GS(Games started)	Number of games in the 82 game NBA season a player has started	Quantitative
FG(Field goals made)	The average number of field goals(two and three pointers) a player makes per game	Quantitative
TRB(Total rebounds per game)	The average number of offensive and defensive rebounds a player has per game	Quantitative
AST(Assists per game)	The average number of assists a player has per game	Quantitative
STL(Steals per game)	The average number of steals a player has per game	Quantitative
BLK(Blocks per game)	The average number of blocks a player has per game	Quantitative
TOV (Turnovers per Game)	The average number of turnovers per game	Quantitative
WS/48(Win Shares per 48 minutes)	This is a player statistic which divides credit for team success to individuals on a team on a per game basis. It attempts to calculate a player's impact of them being on the floor in contributing to a win.	Quantitative
OBPM(Offensive Box Plus Minus)	This estimates a player's contribution in points above league average per 100 possessions played.	Quantitative
DBPM(Defensive Box Plus Minus)	This estimates a player's contribution defensively above league average per 100 possessions played.	Quantitative
VORP (Value Over Replacement Player)	box-score estimate of the points per 100 team possessions that a player	Quantitative

	scores over a replacement/bench player translated to the average team over a full NBA season	

Section 6: Classification Question

Exploratory Data Analysis

In order to understand the relationship between our binary response variables and the predictors, we created Figure 18 in order to compare the two levels across all of our quantitative variables. We showed the most significant differences in the figure below. The graphical representation of our data indicated that starters generally have higher VORP scores, turnovers per game, field goals per game, rebounds per game, and assists per game. The VORP score finding aligned with our expectations, as VORP measures a player's value to their team compared to a hypothetical replacement-level player. So we would expect that players who start would have more value to their team compared to non-starters. The turnovers difference did not align with our expectations, as we hypothesized that starters would turn the ball over less frequently. However, we acknowledge that the turnovers are probably inflated by the higher number of minutes played by starters.

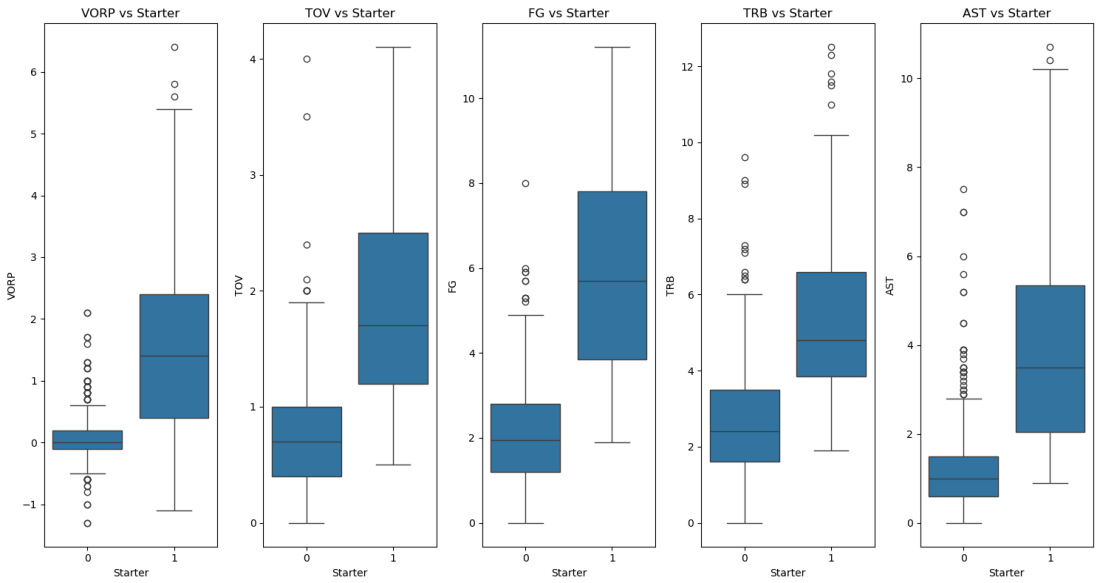

Figure 18: Boxplot of the relationship between response and predictor variables

Next, we looked at a correlation plot of all our variables in order to determine whether multicollinearity existed. As seen in Figure 19, many of the variables have positive correlations to each other. This might lead to possible concerns, as multicollinearity can distort classification accuracy. In Logistic Regression, multicollinearity inflates standard errors, hindering the ability to test predictor significance.

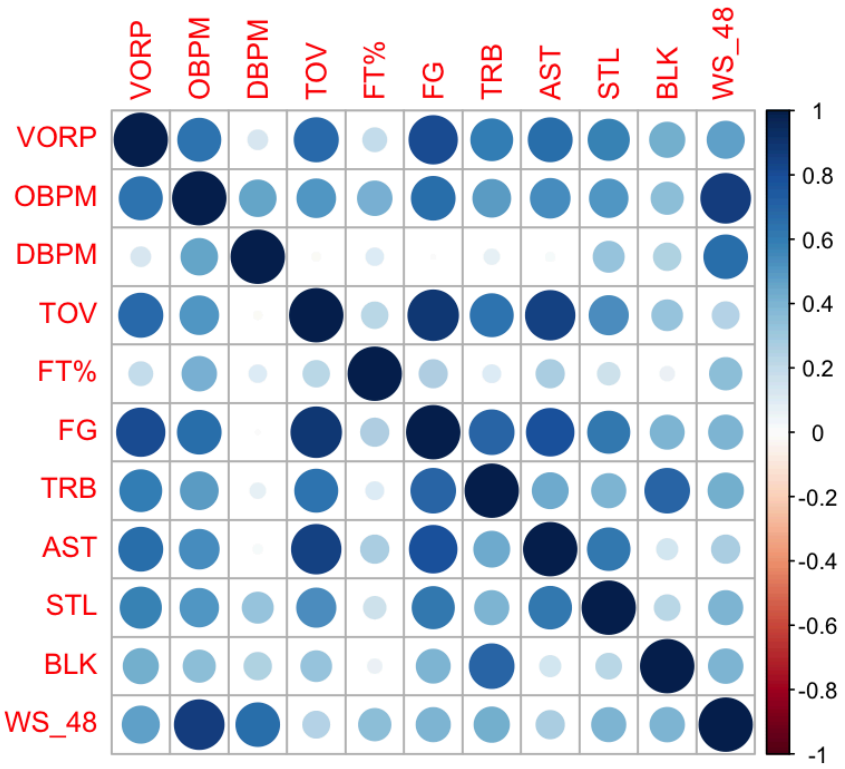

Figure 19: Correlation plot of quantitative variables

Lastly, we wondered whether the position group impacted any of the values of the predictors. We hypothesized that the position of a player would greatly influence the relationship between variables. In order to test our hypothesis visually, we created a scatter plot matrix that is sectioned by color for each position group. We selected the variables that were most “team orientated”. These team-oriented variables likely capture aspects of performance that are highly dependent on a player's role and responsibilities within the overall team strategy and tactics. By coloring the data points by position group, we could easily inspect whether the patterns of correlation or the directions of relationships between these team-oriented variables differed substantially across the different playing positions. This visual inspection enabled us to evaluate if a player's position acted as a moderating factor that altered the associations between key performance indicators. As seen in Figure 20, there does not appear to be a clear delineation between the position groups. Therefore, we decided to exclude the predictor from further analysis.

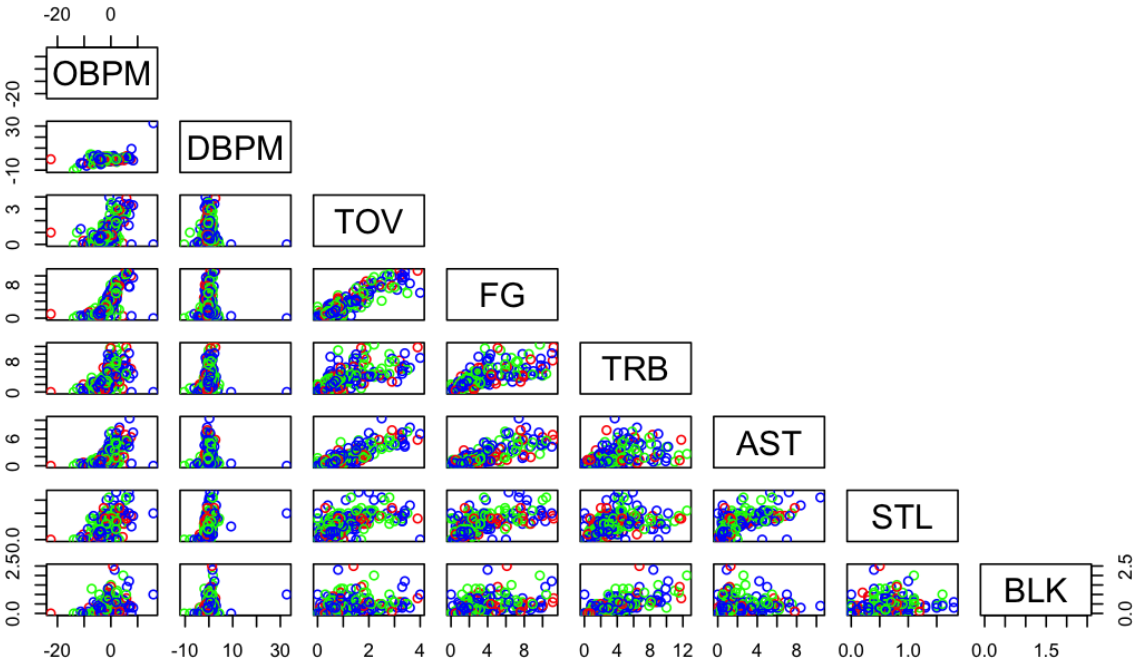

Figure 20: Scatter plot matrix of quantitative variables sectioned by position group. Red observations are Centers, blue observations are Forwards, and green observations are Guards

Logistic Regression

The final set of variables retained includes Starter, Value Over Replacement Player, Offensive Box Plus/Minus, Defensive Box Plus/Minus, Turnovers per Game, Free Throw Percentage, Field Goals per Game, Total Rebounds per Game, Assists per Game, Steals per Game, Blocks per Game, and Win Shares per 48. This selection aims to provide a comprehensive yet non-redundant set of variables for modeling and analysis.

After conducting Kurtosis and Skew tests, we found that the multivariate distribution of the variables was not normal. As seen in Figure 21, the p values of those tests were 2.2 E -16, indicating that the distribution was far from normal. Due to the non-normality of the multivariate distribution, we used a Logistic Regression model for this part of the analysis.

```
        Multivariate Normality Test Based on Kurtosis

data:  train[, -1]
W = 16833, w1 = 0.35503, df1 = 65.00000, w2 = 0.61538, df2 = 1.00000, p-value < 2.2e-16


        Multivariate Normality Test Based on Skewness

data:  test[, -1]
U = 762.52, df = 11, p-value < 2.2e-16
```

Figure 21: Kurtosis and Skew test of multivariate distribution

As seen in the Logistic Regression output below, many of the predictors are insignificant. More specifically, Value Over Replacement Player, Offensive Box Plus/Minus, Defensive Box Plus/Minus, Assists per Game, and Win Shares per 48 were not significant. Having numerous insignificant predictors in a Logistic Regression model indicates that these variables do not significantly contribute to explaining the outcome variable, which can lead to issues such as overfitting, multicollinearity, and including irrelevant predictors.

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -16.069079   3.682755  -4.363 1.28e-05 ***
VORP          0.003409   0.535144   0.006  0.99492
OBPM         -0.211328   0.371250  -0.569  0.56920
DBPM         -0.208477   0.443672  -0.470  0.63843
TOV          -2.816107   1.081646  -2.604  0.00923 **
FTP           6.768816   2.691856   2.515  0.01192 *
FG            1.507007   0.538246   2.800  0.00511 **
TRB           0.685015   0.222909   3.073  0.00212 **
AST           0.402563   0.369080   1.091  0.27540
STL           6.108313   1.943637   3.143  0.00167 **
BLK           2.980793   1.304099   2.286  0.02227 *
WS_48       -17.696032  13.330987  -1.327  0.18436
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 286.502  on 232  degrees of freedom
Residual deviance:  81.884  on 221  degrees of freedom
AIC: 105.88

Number of Fisher Scoring iterations: 8
```

Figure 22: Linear Regression ROC Curve

Despite the high number of insignificant predictors, the Logistic Regression model achieved an impressive ROC curve, as seen in Figure 23. The Area Under the Curve (AUC) of 0.93, is indicative of the model’s strong discriminatory ability in distinguishing between classes. Additionally, cross-validation analysis further underscores the model's robustness, with 5-fold and 10-fold cross-validation error rates of 0.09 and 0.08, respectively. These low error rates suggest reliable performance and good generalization across different validation sets. Furthermore, the test error rate of 0.12 confirms the model's effectiveness in accurately classifying instances in unseen data.

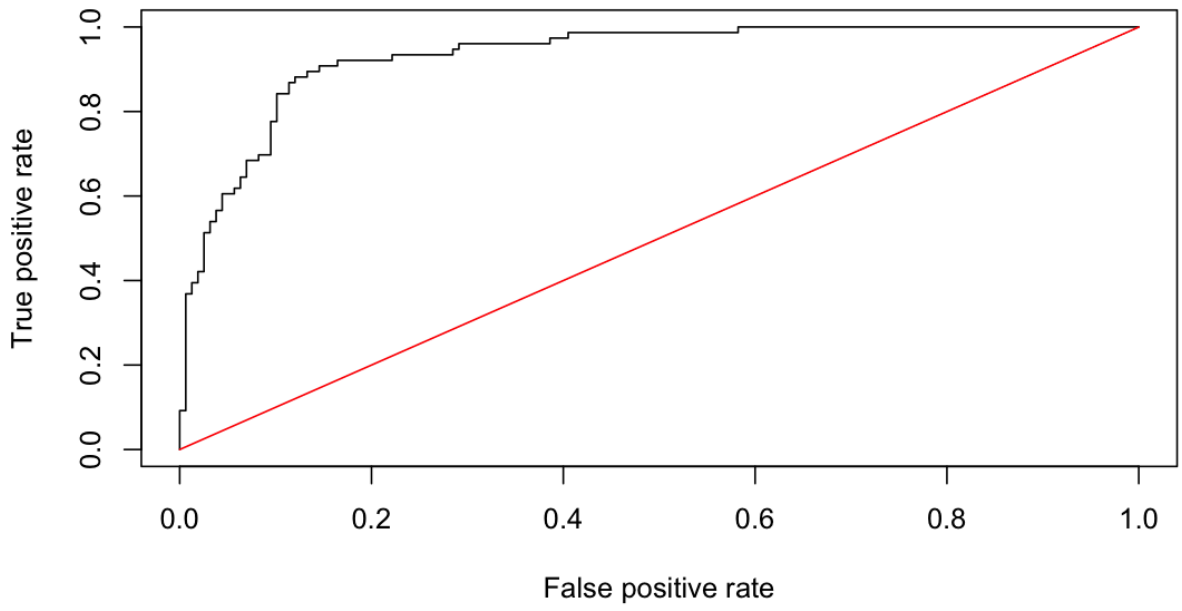

Figure 23: Linear Regression ROC Curve

As seen in the confusion matrix below in Figure 24, the Logistic Regression at a threshold of 0.5 obtained a false positive rate of 0.1 and false negative rate of 0.17. The imbalance in false classifications is likely due to the class imbalance of starters. This means that the ROC curve may overstate the true performance of the classifier by not accounting for the skewed balance.

	FALSE	TRUE
0	142	16
1	13	63

Figure 24: Linear Regression ROC Curve

Classification Trees

Pruned Tree

We used the same selected predictors as the Logistic Regression section to fit the classification tree tree. The final set of variables retained includes Starter, Value Over Replacement Player, Offensive Box Plus/Minus, Defensive Box Plus/Minus, Turnovers per Game, Free Throw Percentage, Field Goals

per Game, Total Rebounds per Game, Assists per Game, Steals per Game, Blocks per Game, and Win Shares per 48. This selection aims to provide a comprehensive yet non-redundant set of variables for modeling and analysis.

We decided to fit a pruned tree, as Recursive Binary Splitting resulted in a tree with many more terminal nodes. The RBS tree contained 14 nodes, while the pruned tree used 9. As we can see in Figure 25, the deviance is minimized at 9 nodes, however the deviance is relatively similar to the full tree. As seen in Figure 26, the pruned tree has 9 terminal nodes and uses the predictors VORP, STL, AST, TRB, FTP, and WS_48. This classification tree helps us determine which predictors are the most important, and helps to visualize what standards the NBA player should reach to become a starter.

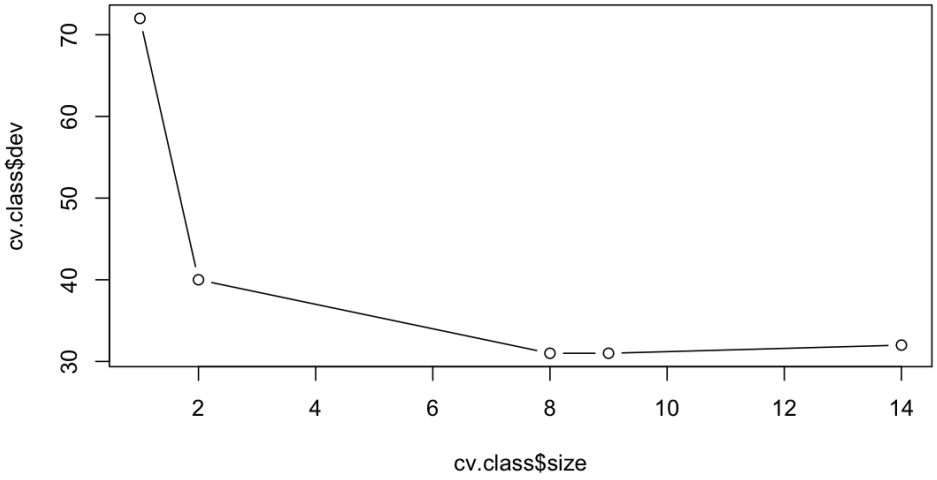

Figure 25: Tree Size vs Deviance

```
Classification tree:
snip.tree(tree = tree.class.train, nodes = c(3L, 4L, 47L, 93L
))
Variables actually used in tree construction:
[1] "VORP"  "STL"   "AST"   "TRB"   "FTP"   "WS_48"
Number of terminal nodes:  9
Residual mean deviance:  0.346 = 77.51 / 224
Misclassification error rate: 0.0515 = 12 / 233
```

Figure 26: Recursive Binary Splitting Output

The most important predictors are VORP, STL, and AST, as they are the first splits in the decision tree as seen in Figure 27. The decision tree answers our question of interest by graphically representing the most important factors that lead to being a starter. The test error rate using this tree is 0.175, which is the exact same as our RBS tree. The pruned tree's advantage is that it is a lot smaller, leading to an easier interpretation.

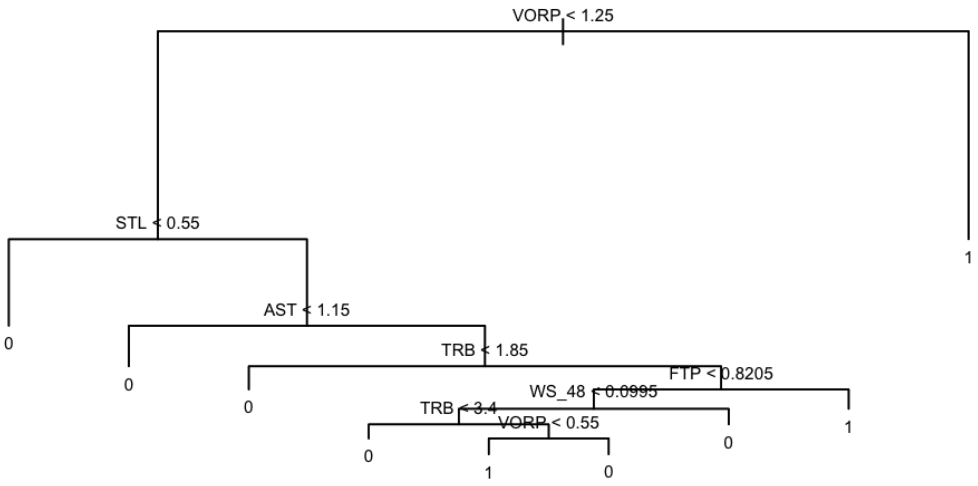

Figure 27: Recursive Binary Tree

As seen in the confusion matrix below in Figure 28, the pruned tree at a threshold of 0.5 obtained a false positive rate of 0.1 and false negative rate of 0.17. Similar to the Logistic Regression model, the false classification imbalance is likely due to the class imbalance of starters.

	0	1
0	135	23
1	18	58

Figure 28: Linear Regression ROC Curve

Random Forests

Lastly, we attempted to improve the pruned tree by using Random Forests. We decided to set mtry to 3, as it is the square root of 10 (the number of predictors) rounded down. That meant that the algorithm would try 3 predictors at each split. As seen in Figure 29, the most important predictors are Assists per Game, Value Over Replacement, Total Rebounds per Game, Steals per Game, and Turnovers per Game. Although the two metrics measuring importance, mean decrease in accuracy and mean decrease in Gini, reveal different predictors, they share common well performing predictors. The test error rate using Random Forests is 0.14, which is lower than the pruned tree, but higher than the Logistic Regression.

Looking deeper into the results reveals that the Random Forest model has an extremely low FPR at 0.07 and high FNR at 0.25. This means that the model is classifying a majority of observations as

non-starters, which is why the FPR rate is so low. This leads to some starters to be misclassified, which is why the FNR is high.

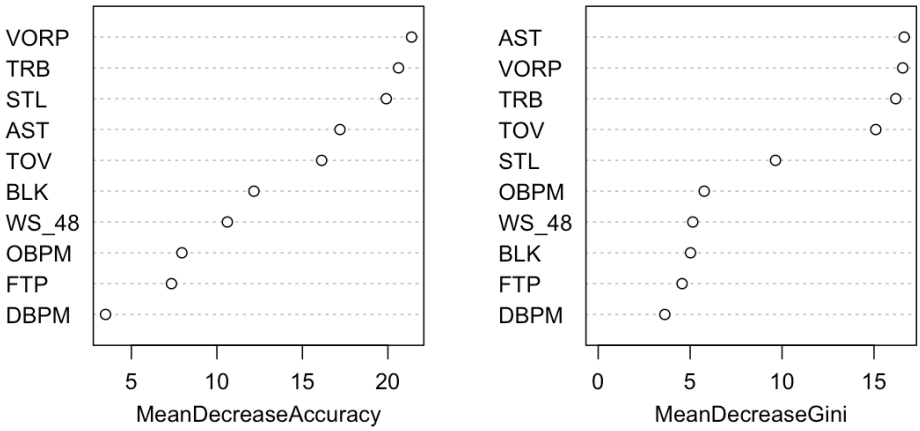

Figure 29: Random Forests Important Variables

Summary of Findings

Overall, based on these metrics at a threshold of 0.5, the Logistic Regression model appears to perform the best, with the lowest test error and a more balanced trade-off between false positives and false negatives. The Random Forest model also performs reasonably well, with a low false positive rate but a slightly higher false negative rate. The Pruned Tree model seems to be the least accurate among the three, with higher false positive and false negative rates, as well as the highest test error.

In Figure 30, we compare all the classification methods. Focusing on the False Positive Rates (FPRs), the Random Forest model has the lowest rate at 0.07, indicating it correctly identifies the most negative instances. The Logistic Regression model has a higher FPR of 0.10, while the Pruned Tree model has the highest FPR at 0.15, misclassifying more negative instances as positive. Regarding False Negative Rates (FNRs), the Logistic Regression model has the lowest rate at 0.17, suggesting it correctly identifies more positive instances. The pruned tree model has a slightly higher FNR of 0.24, and the Random Forests model has the highest FNR at 0.25. In terms of Test Errors, the Logistic Regression model has the lowest value at 0.12, indicating better overall classification performance on the test data. The Random Forest model has a slightly higher Test Error of 0.14, while the Pruned Tree model has the highest Test Error at 0.18 among the three models.

It's important to note that these observations are specific to the chosen threshold of 0.5, and the relative performance of the models may vary at different threshold values.

Model <chr>	test_errors <dbl>	FPRs <dbl>	FNRs <dbl>
Logistic Regression	0.12	0.10	0.17
Pruned Tree	0.18	0.15	0.24
Random Forest	0.14	0.07	0.25

Figure 30: Logistic Regression, Pruned Tree, and Random Forest Comparison

We believe that the threshold should not be adjusted. By maintaining the threshold at 0.5, we ensure that the individuals classified as starters have a higher probability of truly being suitable for the starter role. This approach trades off a higher false negative rate for a lower false positive rate, which is more desirable in this context.

While some promising players may be missed due to false negatives, those identified as starters can have a higher degree of confidence in the model's prediction. This allows teams to reliably evaluate which statistical characteristics are associated with increased playing time and a starting role. Players themselves can then use this information to understand which specific stats or skills to focus on improving, in order to increase their chances of being classified as a starter by the model. Coaches can also use the objective criteria from key stats when making decisions on playing rotations and starter roles.

	FALSE	TRUE
0	142	16
1	13	63

	0	1
0	135	23
1	18	58

	0	1
0	150	12
1	18	53

Figure 31: Logistic Regression, Pruned Tree, and Random Forest Confusion Matrices at threshold of 0.5

By maintaining the 0.5 threshold, the analysis provides a solid framework for teams to optimize talent evaluation, acquisition, and developmental focus, while also giving players clarity on which skills may guarantee more minutes.

The Logistic Regression model appears to be the best choice for identifying the statistical drivers of players being overall starters. Firstly, it provides interpretable coefficients that directly quantify the impact of each predictor variable on the outcome, allowing teams, coaches, and players to easily understand which specific stats are most influential in determining a starting role. The Logistic Regression model has the lowest test error rate among the compared models, indicating better overall classification performance and generalization ability in accurately classifying players as starters or non-starters based on their statistical profiles.

Second, the Logistic Regression model strikes a more balanced trade-off between false positives and false negatives at the 0.5 threshold, which is desirable in this context. Using the Logistic Regression model reveals that Turnovers per Game, Free Throw Percentage, Steals per Game, and Blocks per Game are the most impactful and controllable variables in order to maximize a player's chances of being a starter. A player should aim to reduce their turnovers, and increase their steals, blocks, and free throw percentage.

Addressing Previous Comments

We addressed previous comments about class imbalance and checking the FPR and FNR as the test error rate can be misleading. The analysis can be found in each model section and the Summary of Findings.

Section 7: Further Work

One thing that our group would consider exploring if we had more time is to test data from different years and compare how player statistics and their salaries have changed over time. The NBA has changed drastically over the years, with a large shift in the way games are played. Before the year 2000, players rarely shot 3-pointers, defense was a very important part of a players game, and the final scores of games usually ended in the 70-90 point range. As the 21st century has progressed, players are becoming better and better offensively, most notably in their 3-point shooting but also just all around offensively.

This has seemingly shifted some emphasis away from a players defense abilities, as teams would rather have a very good offensive player who lacks defensively rather than a great defensive player who struggles to score. This offensive shift can be seen in the final scores of games, as this past season games rarely ended with a team scoring less than 100 points, and teams would often finish around the 130 point range. With this shift in the landscape of the NBA, we would be interested in testing data from different decades and comparing the findings.

Another thing that we would consider doing in the future is adding physical and athletic variables to our model, like height, speed, vertical jump, etc. With this shift in the dynamic of the NBA, the desired physical abilities of players has shifted as well. Back in the day the very tall players were big, not very agile, and had the job of just sitting under the basket to take easy shots and get rebounds. The smaller players were in charge of maneuvering the ball around as they were faster and more agile. However, with the recent shift towards an emphasis on offense, the roles of certain players have shifted as well. Being big and tall is no longer enough to make it in the NBA, teams now look for these sometimes 7 foot plus tall players who are athletic, agile, and can shoot from anywhere. Consequently, smaller players have to be quicker, jump higher, and be more agile than ever in order to have a role on a team. We think adding these physical attributes to our model would create a very interesting, well-rounded look at what creates the most desirable NBA player.

Section 8: Learning Reflection

This project has greatly enhanced our understanding of key concepts in class. We applied the theoretical and practical knowledge from class and applied it to our dataset, which led us to understand the concepts on a much deeper level. When you have to explain these concepts using your own words and data, it just further reinforces our learning.

When looking at models like logistic regression, decision trees and random forests we gained a greater grasp of the practical reasoning behind model selection and evaluation metrics. We were forced to look at the trade offs between different models looking at test error and other metrics. It was also interesting to deal with raw data and the cleaning process that you usually don’t have to do in class.

When your data is throwing errors at you it forces you to figure out the root of the problem and just further deepens your understanding of the concepts and models. Encoding categorical variables, in our case the Starter variable and Position Groups was an example of this, as well as deciding to look at the log of salary due to the large skew which would throw off our models. We also had to look at and decipher visualizations such as confusion matrices and scatter plot matrices which just led to a deeper understanding of the impact of predictors on the response variable.

In the end the project provided a holistic, hands on experience of in class concepts that reinforced our learning through implementation. It bridged the gap between our theoretical knowledge and its practical application, equipping us with important skills that will benefit us in our academic and professional futures.