UNIVERSITY OF WASHINGTON
SCHOOL OF SCIENCE, TECHNOLOGY, ENGINEERING AND MATHEMATICS
DIVISION OF COMPUTING AND SOFTWARE SYSTEMS

**MACHINE LEARNING**

---

**Final Assignment**

# National Basketball Association Game Predictor

---

**Professor:** Afra Mashhadi
**Students:** Alan Lai
Elias Martin
Logan Choi
Phat Tran

WASHINGTON, DECEMBER 2023

# Contents

**Abstract**

Our project focuses on improving fan engagement and tackling the unpredictability in sports analytics, particularly in NBA game outcomes. Moving beyond the profit-driven models used in betting and NBA games, we aim to develop a model that both enhances fan experience and provides a deeper understanding of the game's complexities. Our approach involved experimenting with various algorithms like logistic regression and ensemble classifiers, addressing the dynamic nature of sports. Notably, our top-performing model reached a 70% accuracy rate, matching the industry's best, thus offering fans a more educated way to engage with NBA games and contributing significantly to sports analytics. This achievement highlights the effectiveness of machine learning in predicting outcomes in unpredictable sports environments.

# 1 Introduction

The National Basketball Association (NBA) is a professional basketball league in the United States of America. The organization comprises thirty teams, divided into an Eastern Conference and a Western Conference. Each team plays an eighty-two-game regular season for placements, after which qualified teams enter the playoffs to compete for the coveted Larry O'Brien trophy. Our group aimed to investigate whether machine learning models could predict the outcome of each game. This project is relevant due to multiple modern use cases for such models.

Firstly, many teams utilize data to understand potential matchups and implement different training regimens during specific parts of the schedule. For example, every NBA team engages in opponent analysis to comprehend playing styles, strategies, and key players. This involves studying game tapes, tracking player movements, and employing statistical models to predict opponents' behavior on the court. Furthermore, by measuring the team's offensive and defensive ratings, a team can grasp what adjustments are needed to secure a victory.

Another potential use case is sports gambling. With the legalization of sports betting in multiple states, it has become a booming and growing industry. These companies employ machine learning models for matchups to promote over-unders and rig them to stay profitable. However, these predictions can go both ways, whether it is for the sportsbook companies or the consumers. Consumers can improve their winning percentage by using trained machine learning models to inform their bets. For instance, if a model is accurate enough, perhaps one can use it to make sports betting a revenue stream with an above-average winning percentage.

For our model, we wanted to utilize datasets that included game logs and player statistics that we would engineer features for to predict the results of the NBA games. The overarching explainability concept centers on predicting the outcome of NBA games based on the amalgamation of NBA 2k player statistics and game logs. By supplementing our dataset through using NBA 2k ratings, we are able to leverage yet another useful dataset that NBA fans have meticulously crafted and kept updated, allowing our models to perform even better.

However, there are multiple limitations to take into consideration for this model. Due to the high level of volatility within the NBA such as trades, injuries, and players having hot/cold streaks, it is difficult to predict NBA games. By studying the following related works, the average models had an accuracy rate of 65% while the more sophisticated models had an average of 70%-73%. Nevertheless, we built our model to take into account the team's recent success as well as average availability with their members which will be further discussed in the data section.

# 2 Related Works

In the field of sports analytics, several researchers and data scientists have developed models to predict the outcomes of NBA games:

- Ben Jensen created an NBA Predictor that used game logs to feed a Gradient Boosting Regressor, resulting in a 58% accuracy.
- Vikas Paruchuri also developed a model, the NBA Future Game Predictor, which analyzed game logs from the 2017-18 season and predicted future games with a 63% success rate using a Ridge Regression Model.
- Another notable work is by Josh Weiner, who created the NBA Game Outcome Predictor using season records from over a decade and achieved 67% accuracy with a RandomizedSearch model.

These studies and models have laid the groundwork for current predictive efforts. The goal now is to build upon this foundation with a new model that competes with the current standard in accuracy and goes beyond by improving how fans interact with the game. This model would have applications beyond betting, helping teams with strategies and enhancing the overall fan experience. The model will also attempt to consider the unpredictability of the sport, such as trades, injuries and player streaks, aiming to reach and possibly exceed the accuracy levels of between 65% to 73% found in existing models. The focus is on making a practical tool for different NBA stakeholders, combining accuracy with utility.

# 3 Exploring, Analyzing, and Constructing Data

## 3.1 Preliminary Strategy, Data Crawling Endeavor, Final Dataset Model

Our NBA predictor model relies on accurate, real-time data to simulate real seasons during our game-by-game progression. When we looked for datasets for our models, our target was to find a way to represent player talent, team performance, and injury. We based our data needs off of what is considered common sports knowledge for what contributes to a team winning a match: team performance, player talent, team form, and health levels. At first, we were going to try to predict modern day games with real-time data for the 2023-24 season, but after trying to web crawl data with Helena on official sites like ESPN, the data was difficult and hidden for every team, where buttons had to be manually clicked repetitively in order to obtain data for all teams, making it exceedingly difficult to data gather in this way. Injuries were also challenging to gather for recent seasons, as there is no conventional way to measure a player's durability and resistance to injury, even with the basketball reference website.

We then changed our approach to finding real datasets for previous seasons, as they become more available and cheap (newer NBA Data is typically expensive cost-wise, such as injury reports and advanced analytics). There were a handful of NBA season game logs, NBA player game logs, NBA team averages, and NBA 2K rating/player skill rating datasheets available for the 2015-2020 NBA seasons. Since our data was consistent from the datasheets in this 5 year spawn of NBA data, we switched our model to only learn data within 2015 up to the year it is predicting- simulating real-time knowledge availability of a certain point in time as our model predicts games.

## 3.2 Our Game Impact Categories Explained

For player talent, we used a player skill rating log, which rates players overall out of a maximum score of 99 based on different player skill levels. The player ratings data set is from NBA 2K, a basketball video game that rates players based on all of a player's real life statistics, accolades, and talent level. NBA 2K is notorious for their high-accuracy ratings, with real data scientists, research, and analytics done to decide their ratings, we found it to be a great stabilizer for our player talent measurement. Moreover, the NBA 2K dataset was already cleaned and processed, which when coupled with its meticulously crafted player statistics, made it an obvious choice to incorporate into our models. The NBA 2K data included many statistics including the most impactful rating for winning/losing:

plus and minus performance. This number represents how much a team wins or loses when that player is on the court. An NBA team typically plays 8-10 players in a rotation, so we averaged top 10 player plus-minus ratings, skill rating averages, to get a general power level and plus-minus performance of a team. This was one of the comparisons our model considers when picking a winner between two teams. Furthermore, we explored the data to understand the dataset clearly and thoroughly. We generated pairplots of the most important features for the player talent pool. The columns we utilized were points (PTS), assists (AST), rebounds (REB), minutes (MIN), +/-, and rankings.

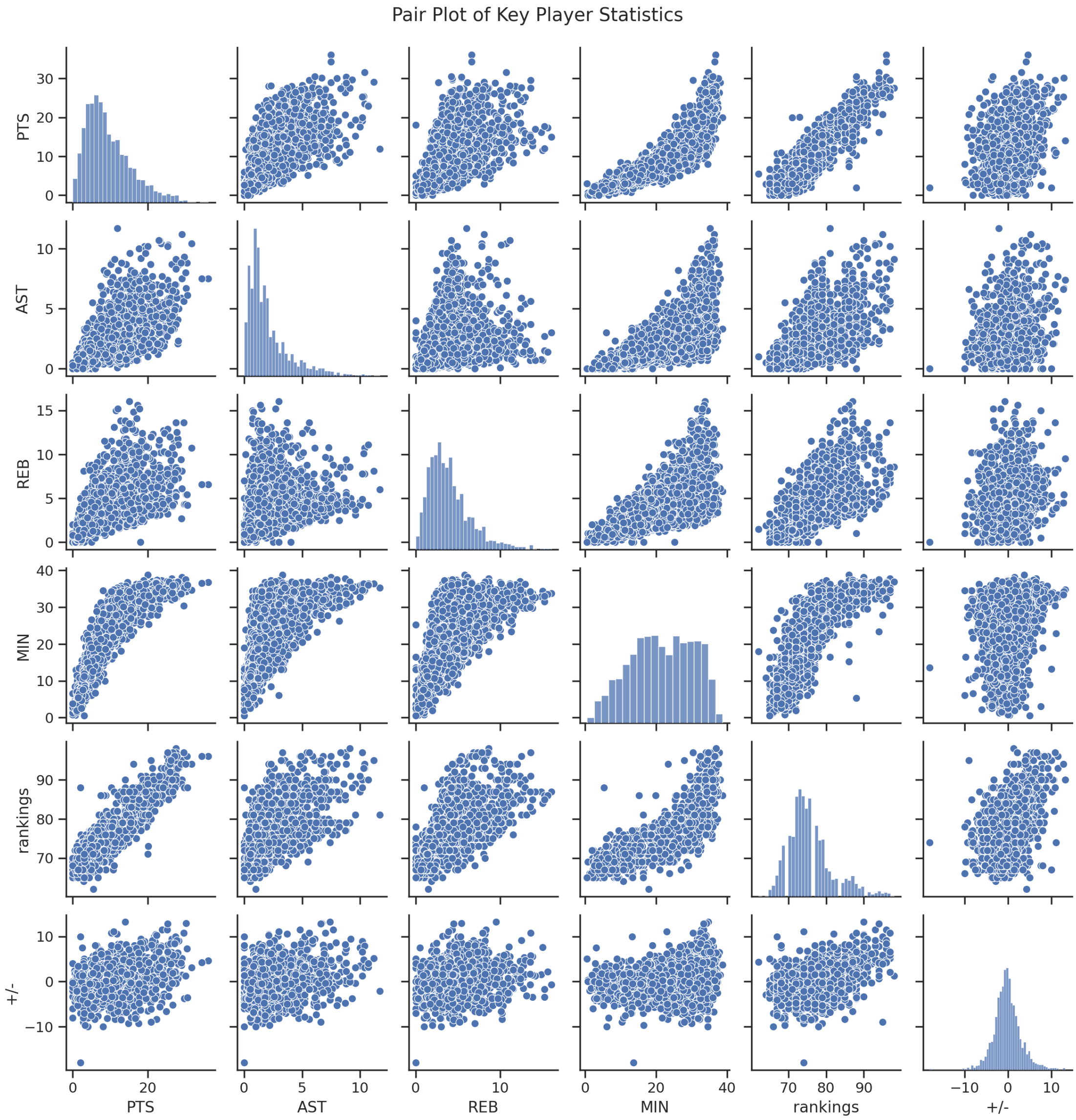


Figure 1: Pair Plot of Key Player Statistics.

Additionally, we generated boxplots that measured every player's rankings for each team. The figure is shown below.

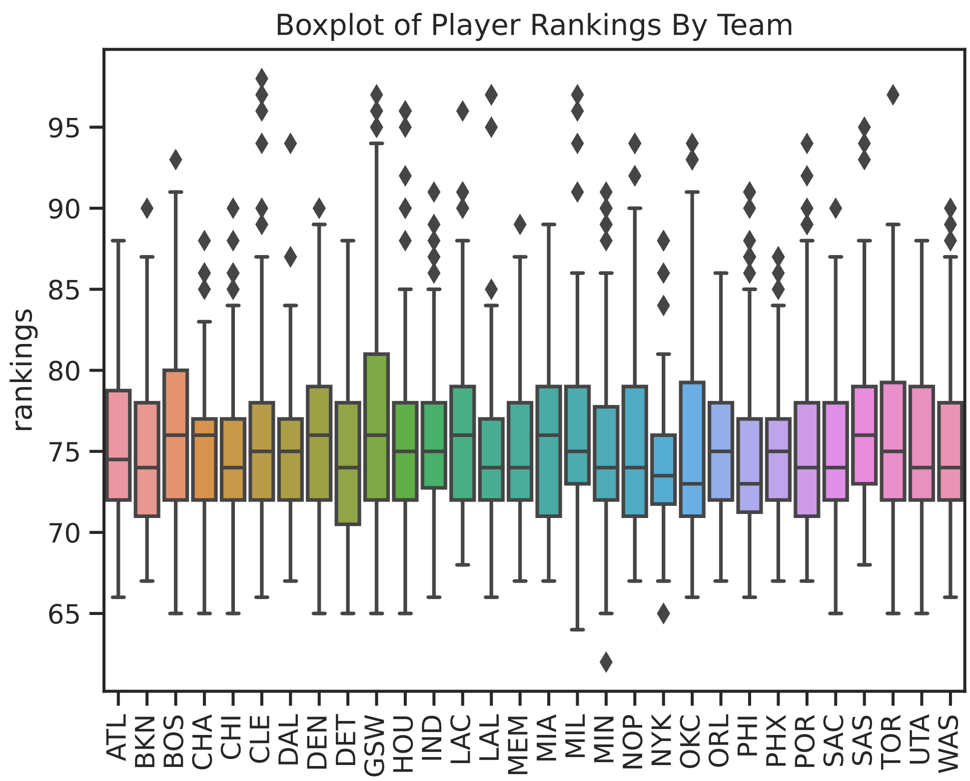


Figure 2: Boxplot of Player Rankings By Team.

As the chart above shows, there is a consistent average across all teams. However, the outliers that are above the boxplots are the most important. As the NBA is a very player-driven league where superstars are considered the best and carries a heavy-load in leading the team to success, those outliers need to be taken into consideration for our final dataset.

As for team performance, we take advantage of average skill ratings and plus-minus ratings (which we feature build later) as these are available in a real season before a game starts (even first game of the season: uses previous season data!). In NBA seasons, teams typically play better based on recent play. As the season progresses, we use offensive and defensive ratings to measure a team's form. For instance, a high performance team will typically win over a struggling team. Thus, our model will compare team performances by utilizing these ratings.

Finally, we wanted to include injuries in our model because they play a significant role in the result of a game, especially when a team is missing multiple impact players or their star player. This is a pivot from our initial midterm report, as we realized in order to reach new levels of accuracy and build a revolutionary model, we needed to incorporate this feature into our models to leverage the most influential features possible when determining game outcomes. We started by first using a player game log from multiple seasons to contrive this feature. While this data alone does not directly tell us how durable a team or player is, we then created a durability score for each player and averaged them for each team by season.

In conclusion for our datasets, the 2015-2020 game log and NBA2K/player statistic data was enough to numerically measure player talent, team performance, team talent, and consider injuries in our model.

## 3.3 Pre-Processing and Organization

Our game log and player statistic data was clean in terms of null values. We simply altered the date formatting, sorted by season, and reshuffled our data in order of importance. We also trimmed out unnecessary data from the player statistics sheet, focusing on only the most impactful information that represents a win or loss of every match. This included offensive rating, defensive rating, plus-minus, and skill level. For game logs, we removed points scored for matches and used win boolean as our target variable for our model. The data considers team ratings and player performance but does not average out injury / talent levels by team- so we had to construct these features.

## 3.4 Feature Construction and Engineering

Now, let's explore the feature engineering part of our project. This is where we work on creating and refining the data attributes (features) that our model relies on to make predictions. We've put together a set of features that cover various aspects of the NBA, like recent team performance, player skills, injuries, and more. These features are essential for our model to make sense of the data and provide accurate predictions. So, let's dig in and see how we've built these features.

### Average Offensive/Defensive Rating (Last 10 Games)

We wanted to understand how well a team has been performing offensively and defensively in their recent games. So, I calculated the average offensive and defensive ratings based on their last 10 games. The offensive rating tells us how efficiently a team scores points, while the defensive rating shows how well they prevent opponents from scoring. These averages give our model a snapshot of a team's recent form and their ability to both score and defend effectively over a short period.

### Home Field Advantage

Home court advantage is a big deal in sports, so we made sure to account for it. We created a simple binary feature - a 1 when the team plays at home and a 0 when they're away. This acknowledges the fact that teams tend to perform better when playing on their home turf, due to factors like crowd support and familiarity with the environment. It's crucial because it can significantly impact a team's performance and, ultimately, the game's outcome.

### Team Form (Last 10 Games)

To gauge how well a team has been doing recently, I looked at their last 10 games. I considered things like the number of wins and losses in this period to measure their current momentum and success rate. This helps us understand whether a team is on a winning streak, likely to continue performing well, or if they've been struggling lately, possibly indicating a decline in their performance.

### Injury Rating

Injuries can be game-changers, especially when key players are sidelined. So, we created a feature to account for this. We looked at player game logs and calculated a durability score for each player. Then, we averaged these scores for each team by season. This gives our model a way to factor in the impact of injuries on a team's overall performance, making our predictions more accurate.

### Plus Minus Average Performance of Team

We wanted to understand how specific players impact their team's performance when they're on the court. So, we calculated the average plus-minus ratings of the top 10 players in each team's rotation. This tells us how much a team scores (plus) or concedes (minus) when these key players are in the game. It helps us evaluate the overall contribution of these players to the team's success.

### Team Age

Age can play a role in a team's performance due to factors like experience and injury susceptibility. To factor this in, we calculated the average age of all players on each team. This helps us consider the age-related dynamics within teams.

Older players might bring experience but could also be more prone to injuries. This feature helps account for the potential influence of age on team performance.

### Average Team Skill Rating

We wanted to know how skilled each team was overall. So, we calculated the average skill rating of all players on a team. These skill ratings take into account various player statistics and talents. This feature provides valuable insights into a team's talent and skill level, helping us predict their performance accurately.

### Max Player Rating

We wanted to know what each team's best player's ranking was. Thus, we retrieved the team's best player ranking and added it into our dataset. This feature takes into account who is the best player on the court as the NBA is a heavily player driven league.

### Number of Superstars

Superstar players can be game-changers, so we counted how many players on each team had skill ratings of 90 or higher, labeling them as superstars. This feature helps us assess the star power of a team and how the presence of superstar players may impact game results.

## 4 Experiments and Results

When evaluating our models, we utilized a combination of metrics to build the most robust possible interpretation of each model's individual performance.The primary metric we used to evaluate models was accuracy, which is especially important for our particular use case as we want to determine exactly how accurate our model is at predicting whether a team will win or lose a matchup against another team. To substantiate this finding, we also evaluated the precision, recall, AUC score and ROC curve to further visualize how effective our model is. We chose to determine precision and recall as we wanted to ensure that our models were minimizing the amount of false predictions it was making. We incorporated AUC score and the ROC curve as we wanted to visualize the tradeoff our models were making between true and false positive rates, as this is important to see how discriminative our model is. The ROC curve for our models are shown below:

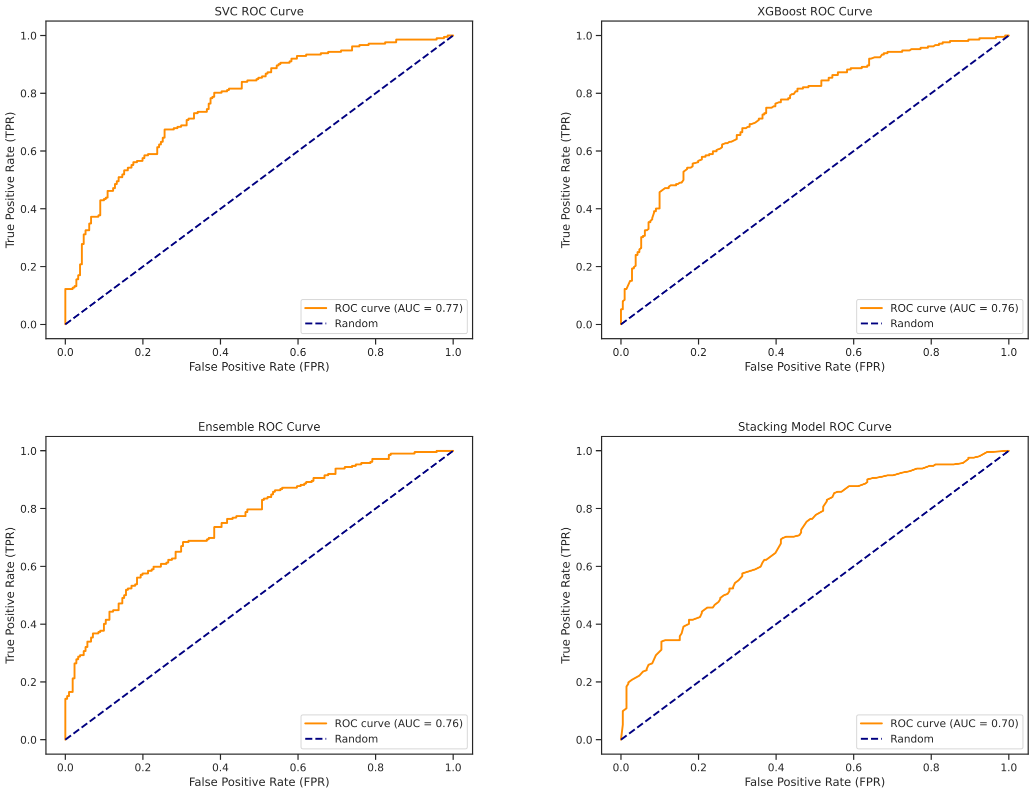


Figure 3: ROC Curves for Different Models.

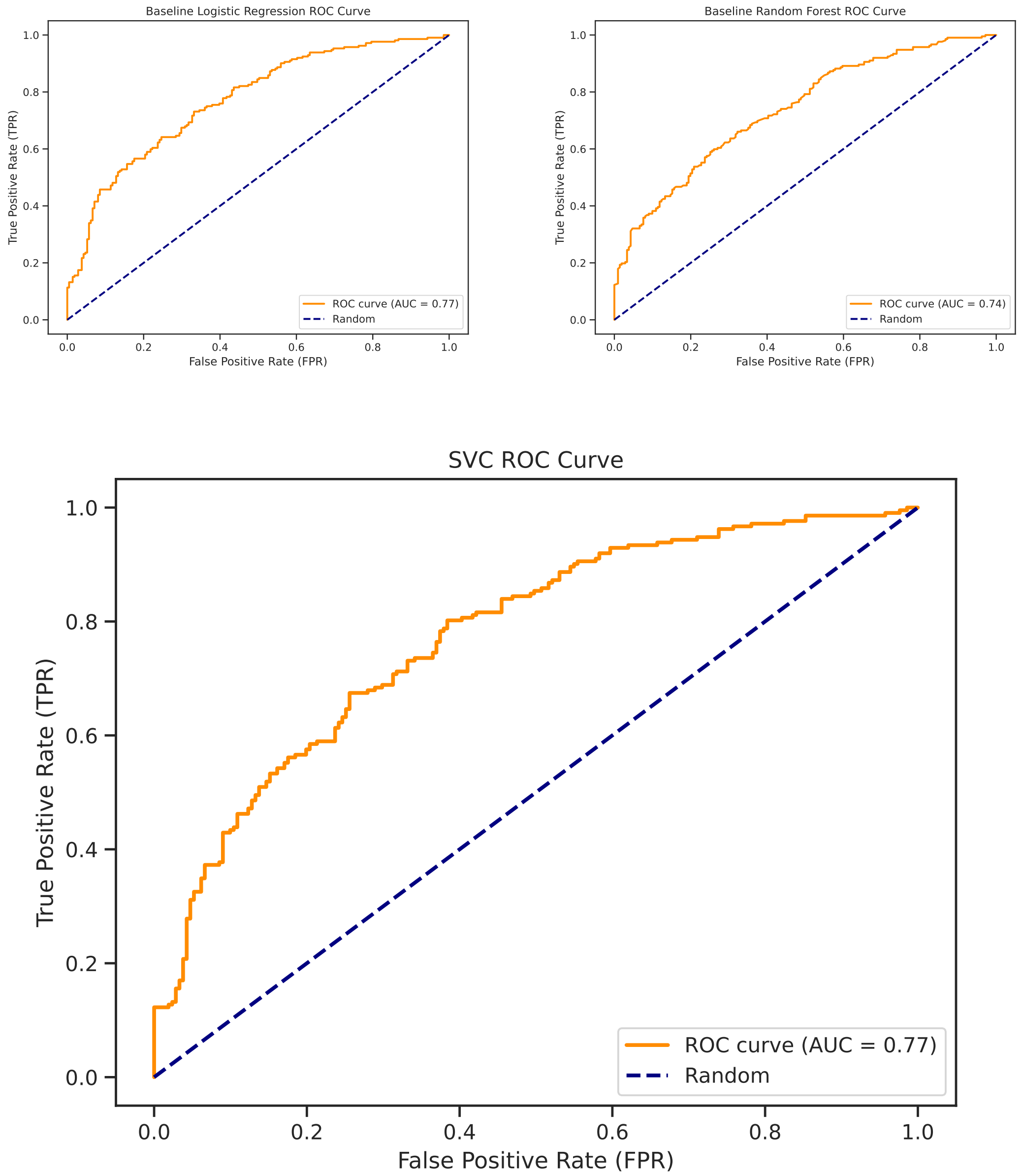


Figure 4: ROC Curves for Different Models.

When tuning and validating the models, we employed the following series of techniques to ensure that we were utilizing our models potential to the fullest:

- Cross-validation (5-fold and 10-fold).
- Customized Hyperparameter tuning specific to each individual model and

explored through grid searches.

- Scaling of feature data for all models (especially important for models like SVC for example).
- Combination of baseline models to get initial performance metrics, and then further refining models through using the above tuning methods.
- Range of models from boosting methods that leverage weak learners, to more substantive models like a stacking classifier and voting classifier to emulate ensemble methods that combine multiple models to yield the highest level of performance.

In using cross validation, we used high levels of cross validation ranging between 5-fold and 10-fold cross validation to use various subsets of the data to ensure that our models provide the best performance estimator, and also that when we are applying our hyperparameter tuning through grid searches, these are compared across a consistent standard. For hyperparameter tuning, we first researched the most influential parameters for each of the models individually, so that we can make sure we are taking advantage of different models strengths / addressing their weaknesses individually. Furthermore, we employed grid search to make sure we test each possible combination of these customized hyperparameters fully, so we can be confident that our models are utilizing the best combination of parameters for its configuration, and then used that model to make our predictions to evaluate the models performance. Moreover, we scaled our input data using a StandardScaler, as some models can be sensitive to feature scale, and thus using a consistently scaled dataset provides an equal platform for evaluating model performance.

Our main takeaways from our performance evaluations was that simplistic models that were well suited to binary classification performed best in conjunction with the previously mentioned tuning methods. Particularly, our SVC model yielded the highest accuracy with 0.70 and AUC score with 0.77. We believe this is due to the SVC being able to take advantage of the complex and non-linear decision boundary of who wins a game matchup since SVCs perform well in multi-dimensional datasets such as ours.

Perhaps the most interesting finding that we discovered was that the Ensemble methods and Stacking classifier did not perform as well as anticipated. These models yielded accuracies of 0.68 and 0.63 respectively and an average AUC score of 0.70, which is not what we initially hypothesized would happen. We theorize that this lower than expected performance can be attributed to the fact that the weak learners in the boosting / ensemble methods utilized in the voting and stacking classifiers might not have been able to take advantage of the complex features utilized in the dataset the way that the SVC was able to. Moreover, we assume that the decision boundary for our problem is highly non-linear, and this could also have confused the stacking / ensemble classifiers to the point that they were not able to make the distinction between wins and losses, resulting in an overall decreased accuracy and lower discrimination rates. Additionally, since the size of our dataset is relatively small, we also theorize that there could have potentially been some overfitting in the stacking classifier and voting classifier that led to weak generalization when interpreting the scaled test data.

# 5 Conclusion and Future Works

In conclusion, the superior performance of the Support Vector Classifier (SVC) in our NBA game outcome predictions can be attributed to its design for binary classification and its proficiency in handling complex, non-linear decision boundaries. This allowed it to capture complex features more effectively than other models. On the other hand, methods like Boosting, Ensemble, and Stacking underperformed, likely due to their inability to discern relationships in complex data as efficiently as SVC, and a tendency for overfitting which resulted in poor generalization. Notably, the highest accuracy was achieved in the analysis of the 2015-2016 NBA season. This season's predictability was primarily due to the NBA being top-heavy, with only a few well-developed teams dominating the league.

In future updates to our NBA game outcome prediction project, we plan to use real-time data for more responsive predictions and improve our over/under predictions to further the betting implications of our models. Our long-term goal is to predict future seasons by considering player and team development. Furthermore, we would like to add a back-to-back games feature where it shows whether or not a team has played the previous night. This will take into account any potential fatigue a team could experience with a fast turnaround in games. However, the increasing competitiveness in the NBA makes accuracy more challenging, so we'll be adapting our methods and including more factors to keep up with these changes. Another application of our model we hope to explore could be a full-stack application that allows users to visually drag the teams they want to see in a matchup and then use our model's predictive power in the background to display our data and metrics to them in a more interactive and visual way to further increase fan engagement.