

NCAA Men's Basketball Game Data Visualization and Analysis

Sta 523 - Final Project

AUTHORS

Yixiao Wang

Beijie Ji

Yiming Cheng

Zhihao Chen

Weitong Liang

Introduction

The goal of our final project was to design a shiny web app to aggregate and show history NCAA men's basketball game data, serving as a preparation of our future prediction task. In this write up section, we will give some background of the NCAA games, explain the functionality of the application and discuss how we implemented the task.

The **National Collegiate Athletic Association (NCAA) Men's Basketball** program is one of the most celebrated and dynamic components of college sports in the United States. It features hundreds of teams from colleges and universities competing across various divisions, culminating in a high-stakes national tournament known as March Madness. This tournament captures the nation's attention with its single-elimination format, thrilling games, and underdog stories. NCAA Men's Basketball is more than just a game—it is a cultural institution that brings communities together, fosters young talent, and showcases the unrelenting spirit of competition.

As students of Duke University, we take immense pride in the efforts and performances of our men's basketball team, a program renowned for its rich history and unwavering spirit of excellence. To celebrate and analyze the team's achievements, we have chosen to create a Shiny app that showcases relevant data in an interactive and visually engaging way. This app will allow users to explore key statistics, game trends, individual player and overall team performances, providing deeper insights into the factors that contribute to our team's success. By leveraging this innovative tool, we aim to highlight the dedication and talent that define Duke basketball.

Our final Shiny app features five main components designed to provide a comprehensive and engaging user experience: Game Data, Team-player Data, Schedule, Prediction, and Write-Up. In the Data section, we developed interactive interfaces that allow users to click and instantly view summary statistics for a single match or a team's overall performance. In the Prediction section, we implemented the XGBoost algorithm and tried to focus specifically on games between Duke and UNC to predict both the winner and the difference in total final points. Schedule section provides real time information of current NCAA games and trend of Duke's team. This write-up provides a concise overview of our work. The detailed implementation of each component is discussed below.

Methods / Implementation

As for implementation procedure, we mainly leveraged functions in `ncaahoopR` package and data from NCAA official website. However, we found that there exist issues in these pre-built functions and recorded data. So we tried to follow the logic and build implementations which fit our goal to show data Shiny app well.

Competition Data: Gather, Display and Analysis

Motivation

Despite the vast amount of attention and data surrounding NCAA men's basketball games, we identified gaps in the accessibility and accuracy of game statistics and player information during our research. To address these issues, we would like this part to be a more reliable and user-friendly platform for querying and analyzing NCAA men's basketball data. By improving data accuracy and enhancing the user experience, our app aims to provide a better tool for fans, analysts, and sports enthusiasts alike.

During the development of this part, we identified several limitations and challenges in the NCAA basketball data presented on its official website. One significant issue was the duplication of player statistics across teams, where a player could erroneously appear in both the home and away rosters for a single game. This not only led to inaccuracies but also complicated the process of analyzing individual and team performance. Additionally, accessing and querying data on the NCAA website proved cumbersome, requiring substantial manual effort to extract meaningful insights. To address these issues, we implemented robust preprocessing steps to clean and aggregate the data, ensuring accuracy and consistency. By automating data fetching and processing, we streamlined the analysis workflow and minimized errors. These efforts were crucial in creating a reliable and user-friendly platform for basketball game analysis, providing users with real-time insights and intuitive visualizations.

For example, on April 8, 2024, during the NCAA men's basketball game between Purdue and UConn (game ID = 401638645), there were notable errors on the official website. These included incorrect display of school names and duplicated player rosters (5 S. Castle). Our webpage successfully addressed these issues by providing accurate school names and ensuring that the player rosters were displayed correctly. This example highlights our motivation for creating this webpage: to deliver precise and reliable information that enhances the user experience and corrects inconsistencies in official sources.

Game statistics from NCAA official website:



Player Stats
UConn

STARTERS		MIN	PTS	FG	3PT	REB	AST	PF	FT	OREB	DREB	STL	BLK
5	S. Castle G	33	15	6-13	1-4	5	3	1	2-4	2	3	1	0
32	D. Clingan C	31	11	5-8	0-1	5	1	4	1-1	1	4	0	1
11	A. Karaban F	36	5	2-7	1-6	6	4	0	0-0	3	3	0	2
2	T. Newton G	39	20	6-13	2-5	5	7	2	6-6	3	2	0	0
12	C. Spencer G	34	11	5-12	1-4	8	2	3	0-0	2	6	2	1

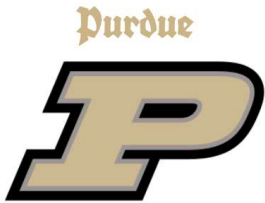


Player Stats
Purdue

STARTERS		MIN	PTS	FG	3PT	REB	AST	PF	FT	OREB	DREB	STL	BLK
5	S. Castle G	33	15	6-13	1-4	5	3	1	2-4	2	3	1	0
15	Z. Edey C	38	37	15-25	0-0	10	0	3	7-10	4	6	0	2
55	L. Jones G	23	5	2-3	0-1	3	0	3	1-1	0	3	0	0
4	T. Kaufman-Renn F	15	4	2-4	0-0	2	0	2	0-0	0	2	0	1
2	T. Newton G	39	20	6-13	2-5	5	7	2	6-6	3	2	0	0
11	A. Karaban F	36	5	2-7	1-6	6	4	0	0-0	3	3	0	2



Our implementation:



Away

Score: 60

Win Prob.: 0%

VS



Home

Score: 75

Win Prob.: 100%

Player Stats



Home Team

Starters

player	FG	3PT	FT	MIN	OREB	DREB	REB	AST	STL	BLK	PF	PTS
A. Karaban	2-7	1-6	0-0	36.00	3.00	3.00	6.00	4.00	0.00	2.00	0.00	5.00
D. Clingan	5-8	0-1	1-1	31.00	1.00	4.00	5.00	1.00	0.00	1.00	4.00	11.00
S. Castle	6-13	1-4	2-4	34.00	2.00	3.00	5.00	3.00	1.00	0.00	1.00	15.00
T. Newton	6-13	2-5	6-6	39.00	3.00	2.00	5.00	7.00	0.00	0.00	2.00	20.00
C. Spencer	5-12	1-4	0-0	34.00	2.00	6.00	8.00	2.00	2.00	1.00	3.00	11.00



Away Team

Starters

player	FG	3PT	FT	MIN	OREB	DREB	REB	AST	STL	BLK	PF	PTS
T. Kaufman-Renn	2-4	0-0	0-0	15.00	0.00	2.00	2.00	0.00	0.00	1.00	2.00	4.00
Z. Edey	15-25	0-0	7-10	39.00	4.00	6.00	10.00	0.00	0.00	2.00	3.00	37.00
B. Smith	4-12	1-2	3-4	38.00	0.00	3.00	3.00	8.00	2.00	0.00	2.00	12.00
F. Loyer	0-5	0-1	0-0	30.00	1.00	1.00	2.00	0.00	1.00	0.00	2.00	0.00
L. Jones	2-3	0-1	1-1	23.00	0.00	3.00	3.00	0.00	0.00	0.00	3.00	5.00

Overview

This part serves as a comprehensive tool for NCAA basketball game analysis, providing users with interactive visualizations and insights into team and player performance. The app integrates multiple R packages, including `dplyr` and `echarts4r`, while also leveraging custom scripts. Initially, we employed the `plotly` package for dynamic visualizations. However, after encountering performance and fluidity issues, we transitioned to `echarts4r`, which offered smoother and more versatile visualizations. Additionally, we navigated challenges with NCAA basketball data, such as duplicated player statistics, missing team logos, and refined the app's aesthetics by drawing inspiration from sports websites. Moreover, to address the issue of missing team logos in the dataset, we manually supplemented and standardized the icons for a polished presentation. This write-up delves into the app's features, the efforts invested, and the core functions that make it effective.

This part also offers a dynamic platform to visualize and analyze basketball game data. It includes the following components:

1. Win Probability Visualization: A real-time chart displaying the chances of each team winning throughout the game.
2. Team Statistics Comparison: Side-by-side visual comparisons of rebounds, assists, shooting percentages, and other metrics.
3. Player Performance Analysis: Detailed stats for individual players, including star performers presented through flower charts and tables.

By combining these features, users can observe score changes and win probabilities dynamically during a game.

Development Challenges and Adjustments

1. Visualization Strategy: Initially, we employed `Plotly` for dynamic charting. While it met basic functionality requirements, its performance struggled with real-time updates. To address this, we transitioned to `echarts4r`, which offered significantly faster rendering and enhanced visual quality. This shift involved a learning curve with the `echarts4r` package and redesigning the visualizations to fully leverage its advanced features.
2. Data Quality Issues: The NCAA basketball data presented notable challenges, such as duplicated player statistics that compromised accuracy. To tackle this, we developed preprocessing steps to clean and aggregate the data, ensuring both accuracy and consistency. Additionally, missing team logos in the dataset posed a visual inconsistency. To overcome this, we manually curated and integrated the missing logos, resulting in a more seamless and professional presentation.
3. Aesthetic Refinements: To enhance user experience, we drew inspiration from professional sports websites. Features like collapsible sections, responsive tables, and dynamic updates were added to align with industry standards.

User Interface (UI)

The UI is structured using `fluidPage` for responsiveness and ease of navigation. Key features include:

1. The Sidebar Panel includes a `textInput` field for entering the game ID and an `actionButton` to update the charts and statistics dynamically, ensuring an intuitive and efficient user experience.
2. The main panel integrates several interactive components to enhance data visualization and user engagement. The Win Probability Chart dynamically displays real-time changes in win probabilities alongside scores, offering a clear view of game momentum. The Team Statistics Section uses bar and pie charts to compare key team metrics effectively. Additionally, the Player Statistics Section presents detailed player data, featuring a flower chart to highlight top performers and comprehensive tables for all players, ensuring a thorough and visually appealing analysis.
3. The interface incorporates collapsible blocks for each section, enabling users to toggle content visibility with ease, thanks to JavaScript-powered functionality. To enhance the visual appeal and user experience, custom styling is applied through CSS, featuring polished designs with animated transitions, responsive layouts, and engaging hover effects.

Helper Functions

In order to fit our expectations, we leveraged several helper functions in this part.

1. **wp_chart:** The `wp_chart` function creates a Win Probability Chart for NCAA basketball games using `ggplot2`. It fetches play-by-play data for the specified `game_id` and calculates metrics like Game Excitement Index (GEI) and Minimum Win Probability. Team colors and logos are determined dynamically or provided as inputs. The function processes game data, including win probabilities and elapsed time, and adjusts for overtime periods. The resulting chart visualizes win probabilities over time, with additional details such as team scores and game date. The function outputs the chart alongside related game and team information, enabling detailed visual and statistical insights.
2. **get_compe_stats:** The `get_compe_stats` function analyzes NCAA basketball play-by-play data. It retrieves game data using the `get_pbp_game` function (from the `ncaahoopR` package) and calculates three types of statistics. First, it summarizes shooting performance by team and shot type (e.g., field goals, three-pointers, free throws) and computes totals and percentages. Second, it tracks scoring patterns by calculating point differences and grouping plays by the scoring team. Third, it extracts behavioral stats, such as rebounds, assists, steals, blocks, and fouls. Finally, the function combines all these metrics into a single dataset, providing a detailed overview of each team's performance.
3. **convert_game_data:** The `convert_game_data` function processes NCAA basketball box score data for a given `game_id`. It first validates the `game_id` and retrieves the data using `get_boxscore`. The nested list of game data is combined into a single data frame, excluding team-level statistics. Key metrics like field goals, three-pointers, and free throws are summarized into readable columns (e.g., "FG" as made-attempted). The data is then split into home and away teams, with unnecessary columns removed. Finally, players are divided into starters and bench players for each team, and the function returns a list containing these categorized data frames.

Visualizations

Visualizations in this part are designed to offer clear and dynamic insights into basketball game data. The win probability chart uses `echarts4r` to track probabilities and scores throughout the game. It features smooth animations, team color coding, and real-time updates. Customizable tooltips provide detailed game phase information, allowing users to observe how probabilities shift as the game progresses. This makes it easy to contextualize score changes and turning points. Team statistics are presented through bar and pie charts that compare metrics like rebounds, assists, and shooting percentages. Bar charts use negative values for one team to create a balanced visual comparison. Pie charts illustrate the distribution of metrics such as free throws, field goals, and three-pointers. These charts are enhanced with color-coded segments and interactive tooltips, making team performance easy to interpret and engaging for users.

Player performance is visualized using a flower chart that highlights leaders in points, rebounds, assists, and blocks. This is complemented by detailed tables summarizing stats for both starters and bench players. These tools provide a comprehensive view of individual contributions and team dynamics, enabling users to dive deep into player performance. Interactive features further improve the user experience. Collapsible sections help users focus on specific insights, while dynamic updates refresh charts and stats instantly when a new game ID is entered. Hover effects allow users to explore detailed information within charts, and real-time updates enable

the monitoring of score changes and win probabilities as they happen. Together, these features create an intuitive and seamless analysis platform for basketball enthusiasts and analysts.

Summary

This part exemplifies the power of interactive data visualization for basketball game analysis. The development process involved transitioning to `echarts4r` for improved visualizations, addressing data quality issues, resolving missing team logos, and refining aesthetics inspired by professional sports websites. The result is a platform that provides real-time insights into team and player performance. But we have to mention that there are still some issues in functions of `ncaahoopR` package, so we cannot get correct `game_id` using more common game information like team or date. So here we keep the function using only `game_id` for query. However, users can use link to our team data page to get these `game_ids`.

Team Data: Gather, Display and Analysis

Structure

This part provides team-based navigation enhanced by visual team logos, making team identification immediate and intuitive. We've implemented sophisticated conference-based filtering that allows users to quickly narrow their focus to specific segments of the college basketball landscape. The application also offers detailed team statistics and schedules, complemented by interactive player performance analysis tools. Perhaps most notably, we've developed advanced visualization capabilities for assist networks, providing unique insights into team dynamics and player interactions. All of this functionality is available across multiple seasons, allowing users to track changes and trends over time.

1. Teams Panel:

- **Key Features:**

- Interactive grid display of team logos and names.
- Dynamic filtering by conference or specific team name.
- Responsive design adapting to different screen sizes.

- **User Experience:** Through a visually appealing grid, users can quickly find teams of interest. Conference-based and team name filters streamline the discovery process, ensuring swift navigation even amid large, complex datasets.

2. Details Panel:

- **Key Features:**

- In-depth team overview with season-by-season analysis.
- Comprehensive game schedule data, including team and opponent scores.
- Player statistics exploration, offering granular performance metrics.
- Advanced assist network visualizations that reveal team passing patterns and player interactions.

- **User Experience:** After selecting a team in the Teams Panel, users gain access to rich historical data and dynamic visualizations. By enabling deep dives into schedules, player stats, and intricate assist networks, the Details Panel transforms raw data into vivid basketball narratives.

Methods

We implemented several user interface design methods to enhance the display and interaction with game data. First, we collected all team logos and created filter bars that allow users to sort by conference or team, enabling quick access to their favorite teams. Second, we designed a dynamic navigation bar that appears when users click on a team logo to view detailed statistics. This bar indicates the current page context and disappears when users return to the team overview page, improving navigation and user efficiency.

In addition to providing detailed game data for individual teams, we summarized player performance statistics to offer insights at both the team and individual levels. This dual focus not only enriches the user experience but also supports the development of predictive models, as player performance similarities often serve as key indicators in analyzing team dynamics.

We further enhanced the overall view by utilizing functions from the `ncaahoopR` package to generate assist network graphs. These graphs combine team-level and player-level assist networks, giving users a clear visualization of the team's dynamic structure. Each player in the network is highlighted using their team's unique color, enhancing clarity and engagement. These design elements make the team and player data easy to interact with and visually intuitive.


Prediction

XGBoost (eXtreme Gradient Boosting) is an efficient and scalable implementation of the gradient boosting algorithm, designed for both regression and classification tasks. Developed by Tianqi Chen, XGBoost has gained popularity due to its high performance, flexibility, and ability to handle large datasets with high-dimensional features. Key features of XGBoost include: Gradient Boosting: It uses gradient descent to optimize an ensemble of decision trees by iteratively minimizing the loss function. Regularization: L1 (Lasso) and L2 (Ridge) regularization are incorporated to prevent overfitting and improve generalization. Parallelization: XGBoost supports parallel computation, making it significantly faster than traditional gradient boosting. Handling Missing Values: The algorithm automatically handles missing data without requiring imputation. Tree Pruning: It uses a `max depth` and `min child weight` approach to control tree complexity, reducing the chance of overfitting. Customizable Objective Functions: XGBoost allows users to define custom loss functions, making it versatile for various tasks. Due to these advantages, XGBoost is widely used in data science competitions and real-world applications for tasks like forecasting, classification, and anomaly detection.

In our program, XGBoost is used to predict the point margin between the University of North Carolina (UNC) and Duke basketball teams based on historical game data. The data, gathered from the `ncaahoopR` package, includes scores, game locations, and team identifiers. We create features such as `game_index` (the chronological order of games), `is_unc_home` (indicating if UNC is the home team), and `is_duke_home` (indicating if Duke is the home team), with the target variable being `margin`, which represents the score difference (home score minus away score). The dataset is split into a training set (80%) and a validation set (20%) to evaluate model performance. An XGBoost model is then trained with parameters set for regression tasks, including a learning rate (`eta`) of 0.1, a maximum tree depth of 3, and early stopping to prevent overfitting. After training, feature importance is examined to understand which factors contribute most to predictions. For the 2023-24 season, we retrieve UNC's upcoming schedule and identify games against Duke. The trained model is used to predict the point margin for these games, rounding the predictions up to the nearest integer using `ceiling()` for clarity. Based

It is worth noting that the prediction section is just a naive trial using the XGBoost algorithm. The main purpose of this project is to create an interactive interface to showcase data within the Shiny app. Thus, we fitted the XGBoost model as a baseline, focusing on only games between Duke and UNC to predict both the winner and the difference in total final points. Also, we encountered some issues with the `ncaahoopR` package, which made parsing data time-consuming. To address this, we prepared the game data in advance. Future efforts may address these challenges and incorporate a more robust and insightful prediction model.

Duke



Choose a season:

2023-24 ▾

Team & Player Data
Assist Network

Select a graph

Assist Network
▾

Duke Weighted Assist Network for 2023-24 Season

Weighted Assist Frequency Leader: Tyrese Proctor (22.5%)


Weighted (Assisted) Shot Frequency Leader: Kyle Filipowski (20.5%)

PageRank MVP: Jared McCain (0.178)

Hub Score MVP: Tyrese Proctor (0.577)

Authority Score MVP: Jared McCain (0.553)

Team Clustering Coefficient: 0.914



The functions described above were initially developed as separate Shiny apps. To enhance user experience, we integrated these individual apps into a single, cohesive application. However, as we are not highly experienced with linking files in the Shiny environment, we combined all the code into a single file. While this approach may generate some warnings, the app performs well in the testing environment using the Posit Workbench provided by our department. We recommend opening the final aggregated file in this online workbench for the best experience. Additionally, we will provide the original draft files for reference.

9/10

The navigation bar was refined with colors inspired by the NCAA logo and now includes buttons for all the app's functionalities, offering an intuitive and visually appealing interface. These efforts aim to provide users with a seamless and interactive experience, similar to navigating a professional website. To make the app more coherent with a real time website, we provide information of time remaining towards Duke's next game and last Duke game result in the schedule part, including NCAA games held today. We believe that this will get users, especially Duke fans more engaged into current agenda of the season.

Discussion & Conclusion

In this project, we tackled the challenge of processing unstructured basketball game data and presenting it in a structured, user-friendly Shiny app. The final product is a cohesive application that enables users to explore game data, team statistics, player performance, and even make basic predictions.

Throughout the development process, we applied concepts and skills acquired during the semester, including foundational R programming, tidy data principles, web scraping, and Shiny app development. This project reflects the integration of these techniques into a practical, comprehensive tool.

We hope this app will not only aid in future basketball prediction research but also stand as a testament to our hard work and learning journey in this course. It serves as both a functional application and a milestone of our progress and achievements.