Oracle Arena
Team Project Report

Nolan Coffey, William Duff, Triton Eden, Kien Nguyen, Ryan Peruski

ncoffey3@vols.utk.edu
wduff@vols.utk.edu
teden@vols.utk.edu
knguye34@vols.utk.edu
yhg461@vols.utk.edu

COSC 402 - Computer Science Senior Design
Department of Electrical Engineering and Computer Science
University of Tennessee
Knoxville, TN 37996

Version 1.04
May 2, 2025

Executive Report

Purpose

The goal of this project is to develop an intuitive and visually appealing web application that delivers player performance metrics and predictive machine learning models based on NBA statistics. This platform will provide a significant enhancement to existing methods of accessing NBA statistics while empowering users to make informed decisions on basic NBA bets.

Overview

Our application will consist of three main sections:

1. Landing Page: Displays upcoming NBA games alongside predictive analysis for each matchup.
2. Data and Statistics Page: Offers comprehensive visualizations of player and team metrics for deeper statistical insights.
3. About Page: Highlights the project's goals and introduces the development team.

Key Objectives

To successfully achieve our vision, the following core requirements will guide our development:

- NBA Data Integration: Accurately retrieve, process, and maintain up-to-date NBA statistics.
- Bet Prediction Models: Design and implement machine learning algorithms to analyze historical data and predict game outcomes.
- User-Friendly Interface: Create a sleek, responsive frontend to seamlessly deliver analytics and predictions to users.

Conclusion

This project is positioned to enhance the presentation of NBA statistics while catering to both casual fans and data enthusiasts. By combining machine learning with sleek design principles, we aim to create a platform that is both powerful and accessible. This report serves as a roadmap to achieving our goals, outlining the methods, tools, and frameworks required for success.

Table of Contents

List of Figures

List of Tables

Problem Statement and Background

The fragmented and unsatisfactory state of current NBA analytics tools highlights a significant challenge for fans, bettors, and data enthusiasts. Existing platforms often excel in one area, such as providing raw player statistics, visualizing historical data, or offering basic betting odds. However, these platforms fail to integrate these features into a cohesive and user friendly solution. As a result, users must rely on multiple tools, a process that is both time consuming and frustrating.

The stakes in the sports betting industry are high, with significant economic growth fueled by less restrictive laws. It is projected that "the Global Sports Betting Market will expand at a compounded annual growth rate of 10.26% and reach revenues of more than $167.6B USD by 2029" [1]. Sports betting companies rely on teams of statisticians and betting experts to set initial lines and then adjust them based on incoming bets, a model that ensures billions of dollars in profit directly from their user base. This dynamic highlights an imbalance, where bettors are at a disadvantage due to a lack of access to similar predictive tools and resources. Oracle Arena seeks to address this imbalance, starting with the NBA, by empowering users with robust data analytics, cutting edge predictive models, and engaging visualizations to level the playing field.

In order to accomplish this goal, a new design must fulfill several key functions. The platform will retrieve, process, and display real time NBA data. It will incorporate machine learning models to predict basic bets such as winner prediction and point spread coverage. These predictions must be accurate, transparent, and easily interpretable to build trust among users. Additionally, the platform will provide a seamless user experience through interactive visualizations, enabling users to explore data intuitively.

Oracle Arena will serve a diverse audience, including casual fans, bettors, and data enthusiasts, each engaging with NBA analytics differently. Casual fans will use the platform to explore team and player performance metrics through visually engaging tools, enhancing their enjoyment of games. Bettors will rely on the platform's predictive models to inform decisions, using win predictions and total score estimates to compare against live betting lines. Data enthusiasts will leverage advanced visualizations and transparent modeling to uncover deeper insights into player and team trends.

The platform must also function seamlessly across devices, from smartphones for casual use during games to larger screens for in depth analysis. This flexibility ensures accessibility and satisfaction, regardless of the user's environment or technical expertise.

Several theoretical and technical foundations support this solution. A thorough understanding of NBA metrics such as player efficiency ratings, team pace, and shooting percentages is essential for generating meaningful insights. To achieve robust predictive capabilities, various machine learning models are being explored. For win prediction, Logistic Regression, Random Forest Classifiers, and Deep Feedforward neural networks were tested. For score total predictions, Linear Regression, XGBoost and Gradient Boosting Regression models are being evaluated. As the project evolves, the team aims to experiment with neural networks in both scenarios to determine their effectiveness in capturing complex patterns in NBA data. This iterative approach ensures that the final solution will be informed by empirical testing and optimized for accuracy.

Many current platforms only partially address the needs of NBA fans and bettors, leaving significant gaps in functionality. For example, websites like ESPN and NBA.com provide access to player stats and game summaries but lack advanced predictive tools or personalized insights. Betting platforms like DraftKings and FanDuel offer predictions and odds but often prioritize their profit margins over transparency or user empowerment. Oracle Arena seeks to combine the strengths of these platforms while addressing their weaknesses by delivering a unified solution that integrates advanced analytics, interactive visualizations, and reliable machine learning predictions. Through ongoing development and refinement, the platform aims to empower users with a seamless and insightful experience.

Requirement Specifications

Model Accuracy

The most critical requirement is ensuring the machine learning models achieve high predictive accuracy for game outcomes and total scores. Accuracy is vital because users rely on these predictions for informed betting decisions. A target accuracy of at least 65% was chosen based on a review of existing prediction models in similar domains and team discussion.

Ease of Use

Ease of use ranks second, as a seamless user experience directly impacts the platform's adoption. The interface must be intuitive, allowing users to quickly navigate, access predictions, and interpret results. User feedback sessions with mockups and prototypes will measure usability using metrics such as task completion time and user satisfaction rating.

Player/Team Data

Comprehensive player and team data will complement the predictions, enabling users to analyze historical performance and upcoming matchups. This requirement reflects user feedback highlighting the importance of context in betting decisions. Data accuracy and granularity will be prioritized, with a target of updating information within one minute of availability.

Model Explainability
Transparency is essential for building confidence in the platform's predictions. The app will provide users with explanations of the factors influencing the predictions, such as player statistics or team trends. This requirement reflects our team's vision of a user centric design.

Technical Approach

Oracle Arena is designed as a full stack web application that integrates real time data processing, machine learning, and intuitive visualization to empower NBA fans, bettors, and data enthusiasts. The platform aims to combine predictive analytics and dynamic user interaction within a scalable and robust architecture. While the project has a well defined structure, the development process remains flexible to accommodate improvements and address challenges as they arise.

The backend is built using Python and the Django framework, hosted on an Azure Virtual Machine. This layer is responsible for the core functionalities, including running machine learning algorithms, handling web scraping tasks, and performing database queries. The backend processes real-time NBA data to generate predictions for key metrics like game outcomes and score totals. Wins are predicted by a custom deep feedforward neural network and score predictions are done by XGBoost for the regular season and ridge regression for the playoffs.

Data storage and management are facilitated by a PostgreSQL database hosted on an Azure PostgreSQL server. This database is structured to accommodate both relational and non-relational data, offering flexibility for future expansions. Relational data, such as team performance metrics, player statistics, and game schedules, is stored in a structured format, while non-relational data, such as raw outputs from web scraping or model logs, can be incorporated seamlessly. This hybrid design ensures efficiency, scalability, and adaptability to evolving project needs. The structure of the database is visualized below in Figure 1.

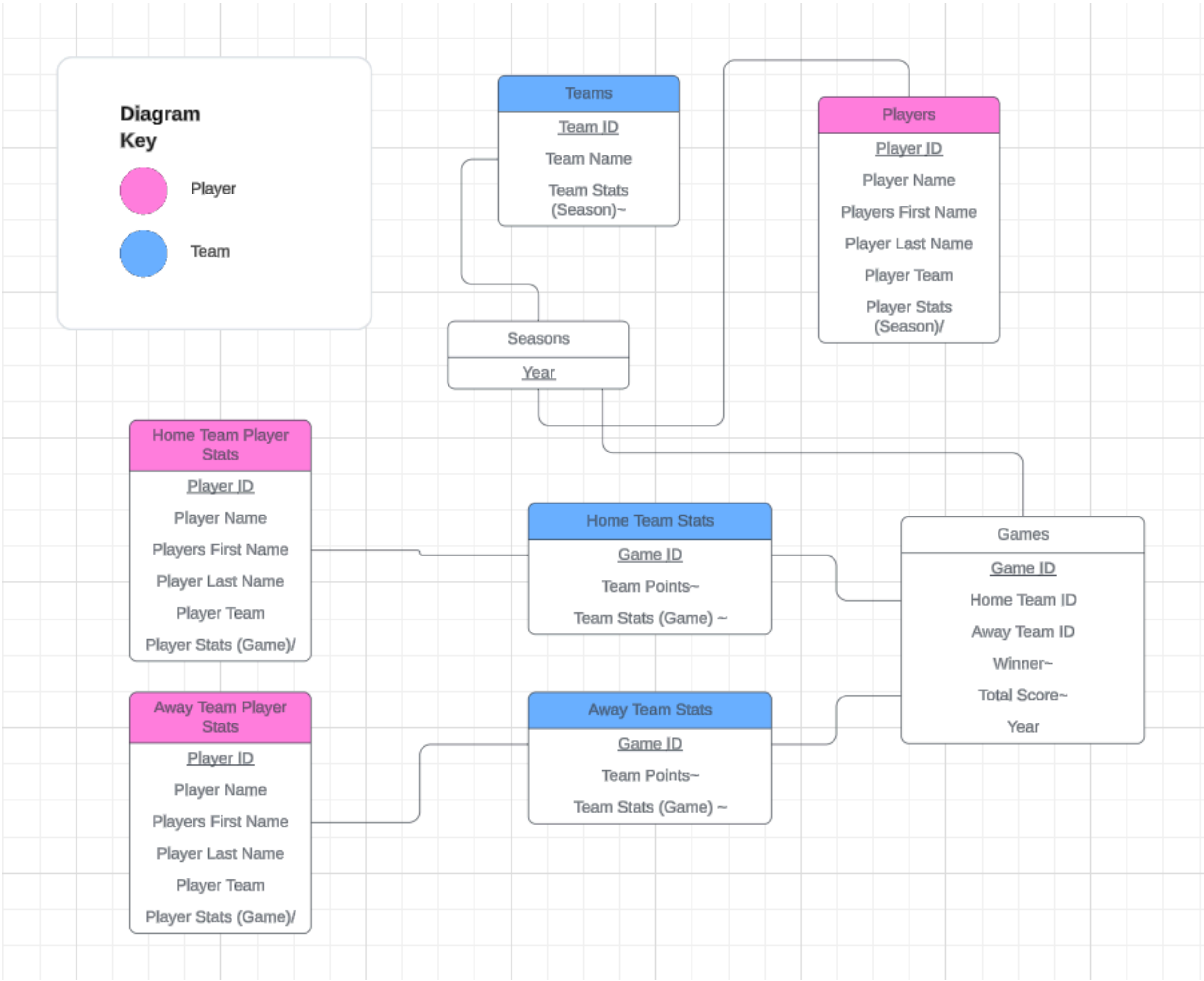
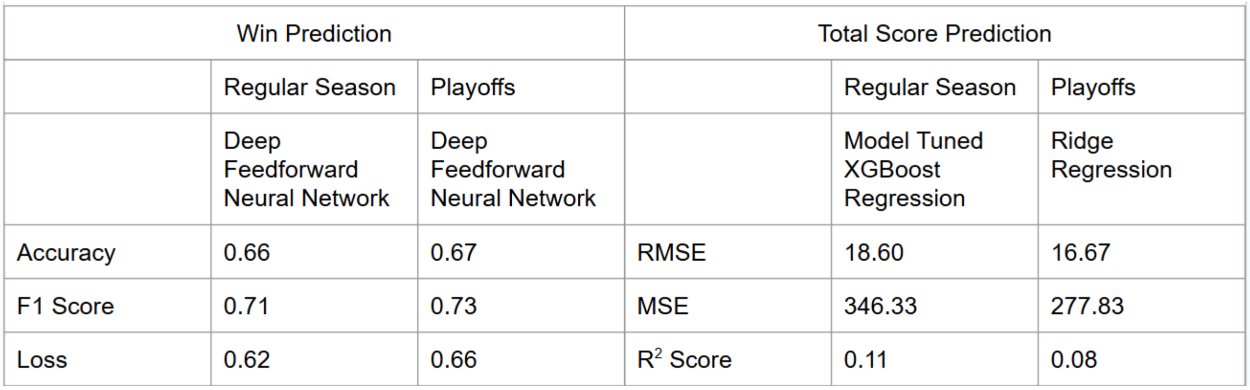


Figure 1: ERD of Database

The frontend, built with the React framework, utilizes TypeScript for its codebase, ensuring type safety and reducing potential runtime errors. This choice enhances developer productivity and maintains a robust and maintainable code structure. For styling, the frontend employs TailwindCSS, a utility-first CSS framework that streamlines the development of responsive and aesthetically pleasing user interfaces. Together, React, TypeScript, and TailwindCSS provide a dynamic and modular user experience, enabling users to interact effortlessly with predictions, visualizations, and game data.

The system's workflow is structured around a seamless integration of its components. A web scraper retrieves real-time NBA statistics from trusted sources, processes and stores the data in the PostgreSQL database. The backend then runs machine learning algorithms on this data to generate predictions, which are sent to the frontend for visualization. This continuous pipeline ensures that users receive accurate, up-to-date insights in real time.

The stats used as input to the models are from the 2018-19 NBA season to the current NBA season. Each of the stats in the stats columns are found for the home and away team per 100 possessions. Those stats are then averaged over the current season and over the past 5 games for the team and their opponent's stats. The data is scaled with a Min-Max scaler to ensure equal contribution of features and then it is split into 80% training data and 20% testing data. This split occurs without shuffling the data to prevent data leakage. Figure 2 is the final testing performance results of each machine learning model.



Win Prediction			Total Score Prediction		
	Regular Season	Playoffs		Regular Season	Playoffs
	Deep Feedforward Neural Network	Deep Feedforward Neural Network		Model Tuned XGBoost Regression	Ridge Regression
Accuracy	0.66	0.67	RMSE	18.60	16.67
F1 Score	0.71	0.73	MSE	346.33	277.83
Loss	0.62	0.66	R^2 Score	0.11	0.08

Figure 2: Machine Learning Model Performance Metrics

Several critical design decisions are being considered to optimize the platform. These include selecting the most suitable machine learning models for different prediction tasks, refining web scraping techniques to ensure data accuracy and reliability, and designing an efficient database schema that supports both structured and unstructured data. On the frontend, achieving a balance between responsiveness and performance is crucial, especially to maintain a smooth user experience across various devices. Additionally, backend hosting strategies, leveraging the Azure VM, are being optimized to handle variable traffic loads without compromising system performance.

Design Concepts, Evaluation, and Selection

Oracle Arena is designed with a strong emphasis on user experience and intuitive functionality, ensuring that the platform delivers robust analytics and predictive capabilities while remaining accessible to fans, bettors, and data enthusiasts alike. The user interface plays a critical role in achieving this balance. With a focus on clarity and ease of navigation, the interface provides users with key components such as a home dashboard for real time NBA game summaries, a data page featuring machine learning generated insights, and an about page listing goals and inspirations for the project.

The dashboard, for example, prominently displays team matchups, win probabilities, and projected scores using dynamic charts and graphs to make complex data approachable.Upon loading, the dashboard defaults to the current date, listing all NBA games scheduled for that day. Each game entry includes the start time, predictions for the winner, and projected total scores. For games that have already concluded, the actual results are displayed alongside the predictions, enabling users to assess the accuracy of the platform's machine learning models directly. The user workflow on the dashboard ensures a seamless experience. Users can select a different date to view historical or upcoming games, with the dashboard dynamically updating to reflect the selection. Interactive elements allow users to drill down into specific games for more detailed analysis, fostering an engaging exploration of Oracle Arena's predictive capabilities. A mockup of this page is shown below in Figure 2.

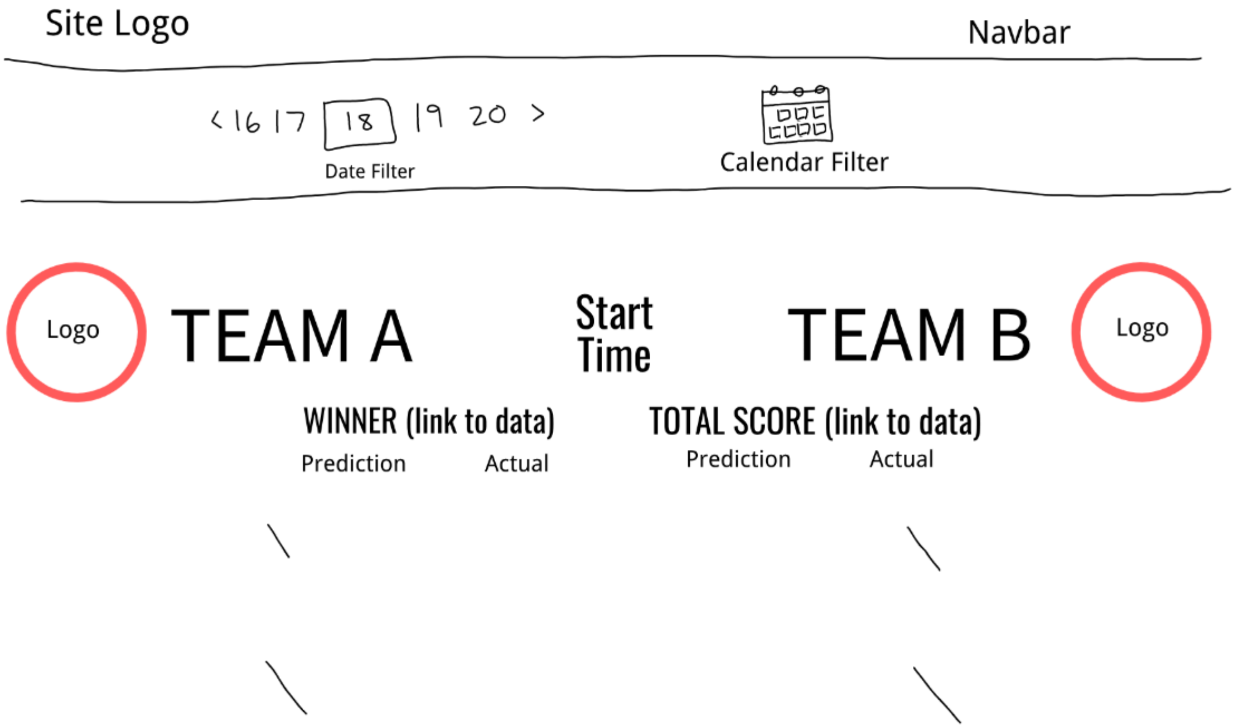


Figure 3: Dashboard Mockup

Design decisions for Oracle Arena will focus on creating a platform that balances functionality, accessibility, and reliability. During development, multiple frontend frameworks will be evaluated to support the platform's dynamic, data driven nature. Angular and React are being considered, with React likely to be selected for its modularity, robust ecosystem, and compatibility with modern web development practices. Tailwind CSS is expected to deliver a clean, responsive design that adapts seamlessly across various devices, enhancing the user experience. TypeScript will be used for frontend scripting to ensure the codebase remains reliable, maintainable, and scalable as the platform evolves.

The overall site layout and workflow will be refined through iterative design processes, incorporating sketches and mockups. Once frontend prototypes are developed, they will be shared with small focus groups of potential users to gather feedback. Additionally, usability surveys will validate design choices by measuring metrics such as navigation ease, feature discoverability, and overall satisfaction. This feedback will ensure the platform aligns with user needs and expectations for navigating predictions, exploring data, and interacting with analytics tools.

Key design decisions will also address the backend architecture and data processing workflows. The backend will need to handle large datasets efficiently while ensuring minimal latency. Django, hosted on an Azure B2s instance, will likely be chosen for its scalability and ability to manage complex operations like web scraping, database queries, and model execution. PostgreSQL, hosted on an Azure flexible server, will provide the flexibility to manage both relational and non-relational data, ensuring the platform can adapt to evolving requirements.

To validate the backend functionality, the primary focus will be on integration and end-to-end testing. Integration testing will ensure that different components of the system, such as the frontend, backend, and database, work together seamlessly. This will involve testing the interactions between the backend and the machine learning models, as well as ensuring that data flows correctly between the server and the frontend. End-to-end testing will be used to simulate real-world user interactions, verifying that the entire system functions as expected from start to finish. This will include testing the real time updates of game data, predictions, and results, ensuring that users receive accurate and timely information.

Product Development and Test/Evaluation Plan

The web application is designed to cater to basketball fans, bettors, and data enthusiasts by providing a simple, responsive interface for exploring player performance metrics, dynamic visualizations, and predictive NBA game outcomes. Users can access three main sections. The landing page, which displays upcoming NBA games alongside machine learning predictive analysis for game winner and total score. The data and statistics page will offer in depth visualizations of player and team metrics for easy insights. The about page will communicate the project's goals and provide team information. The interface prioritizes clarity, seamless navigation, and responsiveness across devices to ensure users can quickly interpret and act on the presented data.

The application is a fully digital platform consisting of a frontend, backend, and database. The frontend is built using Next.js and React to render NBA data in real time, offering dynamic filtering, sorting, and prediction displays. The backend consists of Python based machine learning algorithms that process historical NBA data to generate bet predictions and visualize insights dynamically, as well as APIs to retrieve required data. The database stores NBA statistics, refreshed to ensure predictions that are up to date and that statistics are always available.

The final design of the internal components is structured to ensure seamless data processing and user interaction. Data retrieval is handled through a robust API pipeline that pulls NBA statistics for processing and presentation. The machine learning models will analyze outcomes and predict game results with high accuracy. To present these insights effectively, an interactive visualization tool will be employed to deliver engaging insights on both team and player performance that are easy to interpret.

The prototype is being developed in four key stages:

1. Conceptual Design**:** Initial brainstorming and development of mockups.
2. Initial Development**:** A basic working model focusing on core functionalities and basic UI design. Begin working on and testing machine learning model. Ensure data retrieval is running smoothly and accurately.
3. Enhancement**:** Enhancing the initial product with a more refined UI, bug fixes, and partial feature implementation. Integrate machine learning model predictions and data from backend.
4. Final Prototype**:** The complete product, incorporating all intended features and performance optimizations.

The test plan will focus on several key areas to ensure a seamless user experience and high-quality predictions.

- Data Accuracy Verification: NBA statistics will be verified to ensure they are correctly displayed and aligned with the data source, building user trust in the insights provided by the platform.
- Machine Learning Validation: Predictive algorithms will undergo testing with historical data to confirm their accuracy within acceptable ranges, improving the reliability of predictions for basic NBA bets.
- User Experience Testing: Multiple rounds of testing will be conducted to gather feedback on the UI and navigation, ensuring the interface is intuitive and allows easy access to predictions and visualizations.
- Performance Testing: Data loading times will be evaluated and performance will be optimized for low latency browsing, guaranteeing provided insights are delivered without delays, even with large datasets.
- Cross-Browser Compatibility: Functionality will be verified across popular browsers, ensuring that the platform works seamlessly across different platforms and devices.

- Mobile Responsiveness: Efforts will be made to ensure that visualizations and predictions are fully functional on mobile devices, expanding the platform's usability for a mobile-centric audience.

As work progresses on this project, any modifications to the original design will be listed here:

- *February 12, 2025*: Instead of displaying data visualizations and statistics on the same page, we will create two separate pages for this section of our project. This will provide an easier navigating experience for the user, and help the site layout maintain a clean and polished look.
- *February 28, 2025:* Instead of relying on AWS tools, we ended up switching to the Azure equivalents. This is due to increasing issues with connectivity with AWS as well as UTK being an "Azure-first organization".
- *March 1, 2025*: We modified the ERD to comply with "fourth normal form" (4NF) [3]. This is so that we do not have to worry about contradictory data, as much as possible.
- *March 30, 2025*: While getting all of the data the first time, we had data that was missing, so we added assertions that had to be passed in order to check the integrity of the data. March 30th was the day that we obtained all the previous data with the assertions checked.
- *April 2, 2025*: After discussion, it was decided that a search bar filter would not be necessary on the main page. We removed it and adjusted the design of the Home page after the change.

Social Impact Evaluation

Oracle Arena is a web application designed to be an informational tool that makes predictions and helps basketball sports bettors make more informed decisions. Many sports fans engage in sports betting for enjoyment and as a means to learn more about their favorite sport. However, sports betting is a form of gambling and it is important to recognize the potential ethical concerns raised by this.

Gambling can have significant financial and psychological consequences for individuals, including addiction and financial ruin. On a societal level, sports betting can contribute to issues such as predatory gambling practices and poor financial literacy of users. Recognizing this, our team acknowledges the responsibility of ensuring that our platform is not only accurate but also ethical and transparent in its presentation of data.

To mitigate risks, we will make it explicitly clear that predictions by our Machine Learning models are probabilistic and not guarantees of success. We will implement disclaimers to caution users against reckless gambling and instead advocate for responsible betting. Additionally, we will avoid making any misleading claims about our project (i.e. guaranteed financial success).

By taking these steps, we aim to create a tool that enhances users' understanding of basketball analytics while maintaining ethical integrity and mitigating the negative broader impacts of sports betting on individuals and society.

Deliverables

At the conclusion of the project, Oracle Arena will deliver a fully functional web application designed to empower users with the tools and insights needed to make more informed and competitive NBA bets. The core features of the application will include:

1. Prediction Models: The web app will display real time predictions for NBA games happening on any given day. These predictions will include which team is expected to win and the predicted total score for the game, generated by our machine learning models.

2. Real Time NBA Data: Users will have access to live updates of NBA games, showing not only predictions but also actual game results as they unfold. This feature allows users to compare predictions with actual outcomes, offering valuable feedback to refine future bets.

3. User Dashboard: The dashboard will provide a comprehensive view of all the NBA games scheduled for a given day, along with detailed prediction data for each game. Users will be able to see both the predicted results and the actual game results once the game has concluded. This will allow users to track their predictions and assess the accuracy of the model.

4. NBA Player Statistics: Users will be given the ability to see different scores and evaluations from NBA players in the website's stats page. This will allow for them to have access to common NBA tools on the website to understand how the NBA players are performing.

5. Backend Integration: The backend will seamlessly integrate with the frontend, pulling in live NBA data, running machine learning models, and updating predictions and game results in real time. This integration will ensure smooth data flow, fast updates, and an intuitive user experience.

6. Machine Learning and Predictive Models: Our machine learning models will be designed to make accurate predictions about game outcomes, incorporating a variety of statistical factors and historical data. These models will be continuously refined to improve their accuracy as the project progresses.

7. Mobile Responsiveness: The web application will be designed to be fully responsive, ensuring that users can access the platform on a variety of devices, including smartphones, tablets, and desktops, with an optimized user experience on each.

8. Documentation: The project will include detailed documentation covering the design process, methods of data collection, data cleaning, and processing workflows. Additionally, it will outline the datasets used to train the machine learning models. This documentation will provide transparency, reproducibility, and clarity for both users and developers.

Project Management

As of November 10th, 2024, we are still unsure on what machine learning models will work best. Currently testing using Logistic Regression, K-nearest neighbors, and/or a Random Forest Classifier for the win prediction model and Linear Regression and Gradient Boosting Regression for the score total prediction model. We will hopefully have the chance to try using a neural network for each scenario next semester as well.

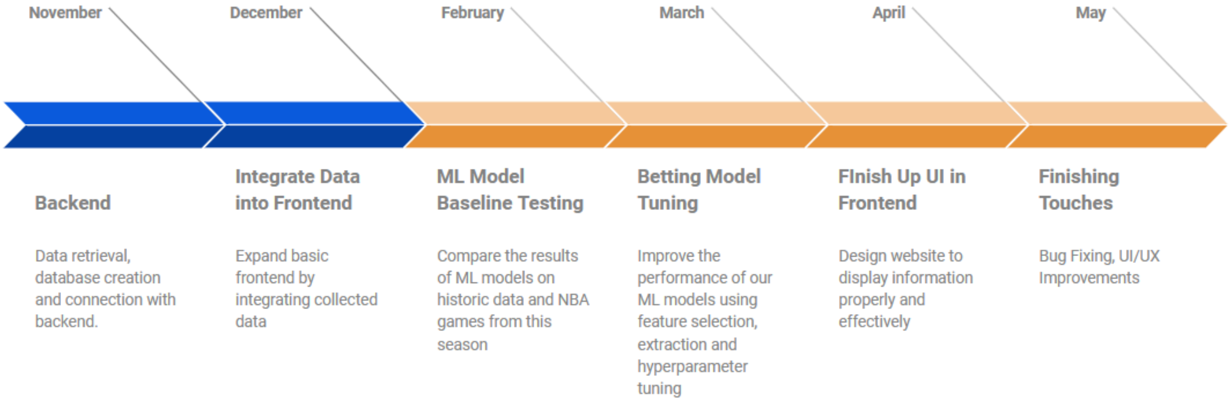

November
December
February
March
April
May
Backend
Data retrieval, database creation and connection with backend.
Integrate Data into Frontend
Expand basic frontend by integrating collected data
ML Model Baseline Testing
Compare the results of ML models on historic data and NBA games from this season
Betting Model Tuning
Improve the performance of our ML models using feature selection, extraction and hyperparameter tuning
FInish Up UI in Frontend
Design website to display information properly and effectively
Finishing Touches
Bug Fixing, UI/UX Improvements

Figure 3: Project Timeline

As of December 3rd, 2024, we have completed the templates for a frontend, backend, and a database. We have all of our initial setup done, so that we can just add to the project during the course of COSC 402.

By the beginning of February 2025, we will have preliminary data for our database to, then, fit into a preliminary ML model. We will also have a template on the frontend ready to display these results of the model. There will also be a statistics page showing data of NBA players in our database. This page will allow for users to look for specific player stats and understand how they can contribute to the probability of winning their games.

During February and March of 2025, we will get a preliminary ML model and observe how this model performs. From there, we will try and tune the model. Furthermore, by now, we will have more data for our dataset, so we can train and test on more data.

During April of 2025, we will put the final touches on the front end. This includes finally importing our model results and database to the front end and displaying those results. Along with displaying our data we also want to hone in on UI/UX design. Most market competitors have unaesthetic and/or clunky web app design so we want to make sure our UI/UX is a strength of our web app rather than a hindrance.

Finally, during the final weeks of the project's life cycle, intensively quality check our web app to make sure that we identify and eliminate any bugs. Any extra time will go to putting any necessary final touches to our UI/UX.

Budget

Our only preliminary budget will include the Azure flexible server for a PostgreSQL database (~$20 a month) and a standard B2s for a virtual machine (around ~$25 a month). We also need to account for storage and networking, which will cost ~$65 a month [2]. Note that this has changed since we were allowed a budget for cloud computing. When accounting for human-hours we estimate by starting on January 21st (start of the Spring semester) and ending on May 7th (last day of classes). This allots to 16 weeks, of which we estimate 7 hours of work per person per week. Given our team of 5 people, this totals to 560 estimated human-hours.

References

[1] Tsygankov, M. (2022). The limits of AI-driven language modeling: Ethical and legal implications. *Gaming Law Review, 26*(5), 287–294. https://doi.org/10.1089/glr2.2022.0046

[2] Microsoft. (n.d.). *Pricing calculator*. Azure. https://azure.microsoft.com/en-us/pricing/calculator/

[3] Fagin, R. (1977). *Multivalued dependencies and a new normal form for relational databases*. *ACM Transactions on Database Systems, 2*(3), 262–278. Archived version