UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Universidad Distrital Francisco José de Caldas
Systems Engineering

Report of a NCAA Basketball Tournament Forecasting System

David Felipe Ariza

Oscar Manuel Contreras

Julian Mendez

Supervisor: Carlos Andres Sierra Virguez

July 12, 2025

Abstract

This report presents the systematic analysis, design, and implementation of a predictive system for forecasting the outcomes of the NCAA Basketball Tournament, with a focus on modularity, robustness, and probabilistic accuracy. The system was developed within the context of the "March Machine Learning Mania 2025" Kaggle competition and leverages supervised machine learning algorithms to estimate game outcomes based on historical performance data and expert-derived variables.

A comprehensive systems analysis was conducted to decompose the tournament environment into its critical components (teams, seeds, matches, and contextual structures) and to model their interdependencies. A modular architecture was then designed, featuring discrete components for data ingestion, feature engineering, model training, inference, and output generation.

Simulation-based evaluation demonstrated that combining statistical features with domain-specific variables such as seed rankings significantly improves predictive performance. The final system achieved strong results on the competition's public leaderboard using the Brier score as the primary metric. While the system accounts for structural complexity and uncertainty, it remains sensitive to data quality and subject to the inherent unpredictability of sports events.

This project provides a scalable and adaptable framework for tournament forecasting and demonstrates how systems engineering principles can be effectively integrated with data science in uncertain and dynamic environments.

Keywords: NCAA Tournament, Machine Learning, Systems Analysis, System Design, System Simulation, Implementation, Forecasting System, Predictive Modeling, Systems Engineering.

Chapter 1

Introduction

The National Collegiate Athletic Association (NCAA) Basketball Tournament represents a complex, high-stakes competition that draws widespread interest from analysts, fans, and data scientists alike. Its single-elimination structure, involving 68 collegiate teams, inherently introduces significant variability, unpredictability, and nonlinear dynamics. These characteristics make outcome forecasting not only a statistical challenge but also a systems-level problem requiring holistic analysis and robust modeling.

This report presents the structured development of a forecasting system designed specifically for the NCAA tournament, with an emphasis on both systemic analysis and predictive performance. The work is framed within the context of the "March Machine Learning Mania 2025" Kaggle competition—a platform that encourages innovative application of machine learning to sports analytics. Participants must design models that can accurately predict the outcome of every potential game, taking into account diverse influencing variables and the stochastic nature of tournament play.

The NCAA tournament system comprises a complex network of interacting components: teams, players, coaches, referees, venues, and fans—all operating under temporal constraints and competition rules. The evolution of these components across the tournament timeline contributes to emergent behaviors that are difficult to anticipate with linear models. Game outcomes may depend on strategic decisions, real-time performance variability, and external perturbations such as injuries or controversial officiating. As such, traditional statistical techniques alone are insufficient.

Existing approaches to sports forecasting have leveraged historical performance metrics, team and player statistics, and probabilistic modeling. However, capturing the emergent dynamics of a tournament requires a comprehensive methodology—one that integrates systems thinking with modern data science techniques. This report responds to that need through a modular system design guided by systems analysis principles, enabling adaptability, maintainability, and iterative improvement.

The remainder of this report is structured to detail the analytical and architectural processes undertaken: from identifying system components and their interrelations, to designing a predictive pipeline capable of navigating uncertainty. The ultimate goal is to develop a flexible, data-driven forecasting platform that not only delivers accurate predictions, but also provides insight into the underlying system behavior driving tournament outcomes.

Chapter 2

Literature

Forecasting outcomes in competitive sports (particularly in high-complexity tournaments like the NCAA Basketball Championship) has emerged as a key area of interest in both the domains of sports analytics and intelligent systems design. The literature reveals a progressive evolution from classical statistical models to more sophisticated machine learning and hybrid approaches, reflecting an increasing recognition of the complex, adaptive nature of sports tournaments.

Early studies predominantly employed statistical models grounded in historical win-loss records, ranking systems, and performance averages. Logistic regression, Elo ratings, and Bayesian networks were frequently used to infer the probability of a team winning a given match. These methods, while interpretable and mathematically grounded, often struggled to accommodate the nonlinearity and chaotic behavior introduced by real-world tournament dynamics such as psychological momentum, coaching tactics, and game-specific anomalies.

As computational power and data availability have expanded, machine learning (ML) approaches have become prominent in this domain. Algorithms such as Support Vector Machines (SVM), Random Forests, XGBoost, and deep learning models have been applied to detect latent patterns and nonlinear relationships in multidimensional sports datasets. These models benefit from flexibility and predictive strength, especially when trained on large volumes of historical and contextual features. However, their "black-box" nature may reduce interpretability and require careful validation through metrics such as the Brier score, log loss, or ROC-AUC.

In parallel, the use of systems analysis to understand sports competitions as complex systems has added a valuable layer of insight. Systems theory emphasizes the identification of interconnected elements and feedback loops within dynamic environments. In the context of NCAA tournaments, this means analyzing how components such as team strategies, referee decisions, seed rankings, fan influence, and venue characteristics co-evolve over time. Recognizing these interactions is crucial for constructing robust forecasting architectures that go beyond static data fitting.

Competitions such as Kaggle's March Machine Learning Mania provide a rich experimental setting where data-driven techniques can be tested under realistic constraints. These platforms encourage reproducible research and foster the development of scalable models that handle missing data, feature drift, and noisy inputs. By simulating the uncertainties of real tournaments, such competitions push for the development of resilient and generalizable prediction systems.

This project builds upon and integrates these bodies of work (combining data science, systems engineering, and sports analytics) to formulate a system capable of producing high-quality forecasts in the face of uncertainty and system complexity. The proposed solution not only employs machine learning but is also grounded in a rigorous analysis of the tournament as a dynamic and stochastic system.

Chapter 3

Background

The NCAA Basketball Tournament, colloquially known as "March Madness," is one of the most complex and unpredictable competitions in collegiate sports. Structured as a single-elimination tournament, it features 68 teams drawn from multiple conferences across the United States. Its multilayered format—ranging from play-in games ("First Four") to the "Final Four" championship—introduces a variety of interdependent factors that influence game outcomes, making predictive modeling a challenging task both analytically and computationally.

The tournament design is characterized by a dynamic progression of matches determined by bracketology principles. Teams are seeded based on their performance during the regular season and conference championships. These seeds aim to balance the competition, but empirical evidence shows frequent upsets—lower-seeded teams defeating higher-seeded ones—adding a strong stochastic component to the system. As such, the tournament exhibits characteristics of a complex adaptive system, with emergent properties that cannot be easily deduced from individual component behaviors.

Several core variables influence match outcomes, including quantitative factors such as scoring efficiency, defensive metrics, and rebound rates, as well as qualitative aspects like coaching strategies, psychological resilience, and team cohesion. Additionally, extrinsic influences—such as neutral venues, travel distance, referee tendencies, and crowd dynamics—can introduce contextual variability that further complicates predictions.

In recent years, the integration of large-scale historical datasets and open-access platforms such as Kaggle has facilitated the use of advanced analytical methods in modeling these competitions. The "March Machine Learning Mania" competition, hosted annually, provides curated datasets containing structured information about teams, seasons, seeds, and outcomes. This enables researchers and practitioners to construct data pipelines and test forecasting models within a controlled yet realistic simulation of the tournament environment.

The background of this project thus lies at the intersection of sports analytics, systems modeling, and machine learning. Understanding the NCAA tournament not merely as a collection of games, but as a complex socio-technical system with sensitivities and emergent phenomena, is key to developing an effective and resilient forecasting solution.

Chapter 4

Objectives

The primary objective of this project is to design and implement a predictive system capable of estimating the outcomes of NCAA Men's and Women's Basketball Tournament games with high accuracy, under conditions of systemic complexity and uncertainty. This system is envisioned as a modular and data-driven solution grounded in both machine learning principles and systems engineering methodologies.

To achieve this overarching goal, the following specific objectives are established:

- Systemic Modeling of the Tournament Domain: To identify and formally describe the key components, actors, and interactions within the NCAA Basketball Tournament system. This includes entities such as teams, matches, seedings, coaches, and venues, as well as their relationships and influence pathways. The objective is to conceptualize the tournament as a dynamic system whose emergent behavior must be captured and modeled.
- Requirements Engineering: To define both functional and non-functional requirements for the prediction system. Functional requirements include data ingestion, feature transformation, model training, probabilistic forecasting, and result output generation. Non-functional requirements encompass performance metrics such as accuracy and computational efficiency, as well as system qualities like modularity, scalability, maintainability, and generalizability across tournament seasons.
- Architectural Design of a Modular Prediction System: To propose and document a layered system architecture based on distinct modules for ingestion, preprocessing, feature engineering, modeling, and output formatting. The design will prioritize separation of concerns, support for pipeline reusability, and flexibility for testing different algorithmic configurations or input datasets.
- Complexity and Uncertainty Mitigation Strategies: To implement analytical and algorithmic strategies aimed at reducing the impact of unpredictable factors inherent to sports events. These strategies include the integration of historical performance smoothing, incorporation of expert-driven variables (e.g., tournament seeds), and use of cross-validation and ensemble techniques to enhance robustness and generalization.

Collectively, these objectives ensure that the resulting system is not only capable of generating accurate predictions but is also resilient, interpretable, and extensible, key characteristics for deployment in evolving competitive environments such as NCAA tournaments.

Chapter 5

Scope

The scope of this project is delineated to ensure a focused and methodologically rigorous development of a predictive system tailored to the NCAA Men's and Women's Basketball Tournaments. The system is designed to operate within the constraints and context of the "March Machine Learning Mania 2025" Kaggle competition, which provides a structured dataset and evaluation framework. The scope includes specific inclusions and exclusions that define the functional boundaries and analytical priorities of the system.

Inclusions:

- Data-Driven Forecasting for NCAA Tournaments: The system is exclusively focused on predicting game outcomes for the NCAA Men's and Women's Basketball Tournaments. Both competitions share similar formats and structural characteristics, making them suitable for parallel modeling under a unified framework.
- Utilization of Kaggle-Provided Datasets: The project uses the curated historical datasets from the Kaggle competition, including regular season results, tournament game outcomes, team seedings, and advanced statistical metrics. These datasets serve as the primary input for model training, validation, and simulation.
- Development of Machine Learning-Based Predictive Models: Predictive modeling is performed using supervised learning techniques. Models are trained on historical match data and evaluated through probabilistic metrics (e.g., Brier score) aligned with the competition's scoring system.
- Generation of Submission Files for Kaggle Evaluation: The system outputs predictions in the required format: a CSV file containing the ID (composed of season, team A ID, and team B ID) and the corresponding predicted Prob (probability of Team A winning). This ensures seamless integration with Kaggle's automated evaluation pipeline.
- Integration of Tournament-Specific Features: The feature set includes domain-specific variables such as tournament seeds, seed differentials, team performance averages, and tournament slot positions, which are known to influence game dynamics and predictive accuracy.

Exclusions:

- Prediction of Other Sports or Leagues: The system is not intended to be generalized to other sports, leagues, or tournaments. Its design, feature engineering, and evaluation logic are domain-specific to NCAA basketball.

- Development of a Graphical User Interface (GUI): The current project is implemented as a functional backend pipeline. It does not include the design or development of front-end components such as dashboards, mobile apps, or visualization platforms.
- Real-Time Data Integration: Real-time ingestion of streaming data (e.g., in-game statistics, live scores, injuries) is not within scope. The system operates exclusively on pre-tournament data.
- Inclusion of Soft Variables and Unstructured Data: Variables such as social media sentiment, player morale, or qualitative expert opinions are excluded due to the complexity of extraction, preprocessing, and modeling within the current timeline and resource constraints.

By establishing these scope boundaries, the project maintains a balance between depth and feasibility, ensuring that the developed system is both technically robust and aligned with the competition objectives and academic expectations.

Chapter 6

Assumptions

The development and implementation of the NCAA Basketball Tournament forecasting system are grounded on a set of foundational assumptions that guide the design decisions, modeling strategies, and evaluation processes. These assumptions are critical for ensuring internal consistency within the system and for delimiting the conditions under which the predictive model is expected to operate reliably. Each assumption reflects both the constraints of the data environment and the operational realities of the Kaggle competition framework.

1. Data Integrity and Availability It is assumed that the datasets provided by the "March Machine Learning Mania 2025" Kaggle competition are accurate, complete, and well-structured. This includes historical records for regular season games, tournament outcomes, team statistics, and seeding data. The quality and consistency of this data are essential for training robust machine learning models and generating meaningful predictions.

2. Relevance of Historical Data The system assumes that historical performance indicators retain predictive value over time. Specifically, patterns observed in previous tournament editions are considered applicable for forecasting outcomes in the 2025 season. While acknowledging year-to-year variability, the modeling process relies on the assumption of temporal stationarity in core predictive relationships.

3. Suitability of Machine Learning Techniques It is assumed that machine learning methods—particularly supervised learning algorithms—are appropriate for capturing the complex, nonlinear relationships between input features and game outcomes. This includes the expectation that patterns embedded in historical statistics and tournament structures can be effectively leveraged to generalize predictions for unseen matchups.

4. Stability of Tournament Format The system assumes that the structure, rules, and operational principles of the NCAA Men's and Women's Tournaments in 2025 are consistent with those of previous years. This includes bracket size, seeding logic, game progression rules, and competition phases. Any structural deviation could impact the model's ability to generalize and interpret tournament context.

5. Feature Sufficiency The selected variables (e.g., team averages, seeds, point differentials) are assumed to contain sufficient explanatory power for predictive purposes. Although the system excludes some contextual variables (e.g., injuries, coaching changes), it is assumed that the included features capture the dominant sources of variation relevant to game outcomes.

These assumptions serve as the epistemic foundation of the forecasting system. While necessary for model tractability, they also represent potential vulnerabilities. Any violation—such as abrupt changes in tournament format or data quality issues—could compromise model performance and would require reassessment of the system's architecture and logic.

Chapter 7

Limitations

While the proposed forecasting system for the NCAA Basketball Tournament demonstrates robust design principles and promising empirical performance, it is subject to a set of inherent limitations that arise from both systemic constraints and the nature of predictive modeling in dynamic environments. These limitations are important to acknowledge, as they define the boundary conditions under which the system's outputs should be interpreted.

1. Data Dependency and Quality Constraints The predictive accuracy of the system is tightly coupled to the quality, granularity, and completeness of the input data. Any inconsistencies, omissions, or biases in the historical datasets—such as incomplete player statistics, unrecorded game anomalies, or mislabeled outcomes—can propagate through the modeling pipeline and degrade performance. The model also relies on the assumption that pre-tournament data captures the most relevant signals for forecasting, which may not hold in all scenarios.

2. Model Abstraction and Representational Limits Machine learning models, by their nature, provide abstract approximations of complex phenomena. Even advanced models such as gradient boosting or ensemble architectures cannot fully capture the multicausal and emergent dynamics of live sports events. Factors like team morale, intra-game momentum shifts, and unquantifiable strategic decisions are difficult or impossible to encode in static feature vectors. As such, the system inherently simplifies the reality it seeks to model.

3. Stochasticity and Unpredictable Events The NCAA Tournament is characterized by high volatility and frequent upsets. Random events—such as last-minute injuries, referee controversies, or anomalous performances—can dramatically alter game outcomes. These events, while impactful, are typically unobservable or unpredictable prior to the match and thus lie outside the scope of any model that relies exclusively on pre-game data.

4. Temporal Drift and Generalization Challenges The model's parameters and feature importances are learned from historical data, which may not fully generalize to future tournaments due to changes in team composition, coaching strategies, or playing styles. This phenomenon—known as concept drift—can lead to declining model performance over time if not properly addressed through periodic retraining or model adaptation.

5. Computational and Resource Constraints The development and training of machine learning models, particularly those incorporating large feature sets and cross-validation techniques, require non-trivial computational resources. Limitations in available memory, processing power, or runtime (e.g., within Kaggle's competition environment) may

restrict the complexity of the models or the breadth of hyperparameter tuning that can be performed.

In summary, the forecasting system is designed to optimize predictive performance within a bounded set of conditions. These limitations do not invalidate the model's utility but rather frame its use as one component of a broader decision-making or analytic process. Future iterations should seek to reduce these limitations through enhanced data integration, adaptive modeling techniques, and deeper system monitoring.

Chapter 8

Methodology

This chapter presents the structured methodology employed in the development of a forecasting system for the NCAA Basketball Tournament. The approach is grounded in a systems engineering framework, combining classical systems analysis with contemporary data science and software architecture practices. The methodology was executed in three phases: System analysis, System design and System Implementation and Simulation.

The overarching goal of the methodology is to develop a modular, extensible, and data-driven prediction platform that captures the nonlinear behavior of tournament dynamics while maintaining computational tractability and engineering rigor. The process integrates problem decomposition, architectural modeling, algorithm selection, simulation planning, and performance evaluation.

Each subsection of this chapter outlines a key component of the methodology:

- Systems Analysis: Element identification, interdependency mapping, and uncertainty modeling.
- System Design: Requirements engineering, architectural planning, and technology selection.
- Implementation and Simulation: Modular construction, evaluation strategy, and execution of simulation scenarios to test system behavior under varying conditions.

This methodology provides a repeatable and adaptable framework for designing intelligent systems in complex, uncertain domains, making it applicable not only to sports analytics but to broader classes of predictive modeling challenges.

8.1 Systems Analysis

The initial phase of our methodology was dedicated to a thorough systems analysis of the NCAA Basketball Tournament. This process, aligned with the objectives of Workshop 1, aimed to deconstruct the tournament environment into its fundamental components and understand their complex interdependencies.

8.1.1 Identification of Key System Elements

The initial step in the systems analysis phase involved decomposing the NCAA Basketball Tournament environment into its fundamental components. This decomposition is critical for constructing a structured model that captures the operational behavior and interdependencies within the tournament system.

Key system elements were identified through iterative review of both the competition's rules and the structured datasets provided by Kaggle. These elements represent entities with distinct roles, attributes, and interactions that collectively influence tournament outcomes. The main components include:

- Season: Defines the temporal boundary for the dataset, serving as a time-based segmentation unit for model training and evaluation.
- Teams: Represent the primary actors within the tournament system. Each team is characterized by performance metrics, seeding information, historical statistics, and conference affiliation.
- Coach: Associated with strategic and tactical decisions, coaching attributes indirectly impact team performance but are not always quantifiable in the available dataset.
- Match: The core unit of prediction. Each match represents a discrete event with known participants (Team A vs. Team B), contextual variables (location, round), and a binary outcome (win/loss).
- Conferences: Serve as organizational groupings for teams. Inter-conference dynamics can influence strength-of-schedule effects and seed allocation.
- Tournament: The competition instance itself, with a predefined structure (e.g., bracket positions, round progression) and constraints (e.g., neutral venues).
- Cities/Venues: Reflect physical locations where matches are played. Although often neutral, some venues may introduce subtle spatial effects.
- Referees: Though not directly modeled in the system, referees are recognized as sources of stochastic influence on match outcomes.

Each element was cataloged and described using a systems-oriented lens, enabling the identification of entity-specific attributes (e.g., seed, win ratio) and their roles within broader system processes. This structured classification forms the basis for interaction modeling, feature engineering, and architectural design in subsequent stages of the project.

8.1.2 Mapping Element Interactions and System Complexity

Following the identification of core system elements, the next phase involved modeling the interactions among those components to understand the NCAA Tournament as a complex adaptive system. This interaction mapping is essential for revealing the data flow, dependency chains, and feedback loops that characterize the system's dynamic behavior.

The relationships between elements were conceptualized through an entity-interaction diagram that can be seen in Appendix A, which captures both structural and behavioral links. For example:

- Teams progress through the tournament by winning Matches, which are scheduled according to bracket Slots and are influenced by contextual attributes such as Venue and Seed.
- Coaches influence team strategies and, indirectly, match outcomes through tactical decisions and player development, even though these factors may be latent in the available data.

- Conferences group teams based on regional and organizational affiliations. Inter-conference play during the regular season introduces variability in team preparedness and seeding outcomes.
- Referees and Fan presence, though not explicitly included as features, were acknowledged as contributors to the stochastic nature of certain match results and systemic variability.

This network of interactions reveals several emergent system characteristics:

- Nonlinearity: The impact of one component (e.g., an underdog team's unexpected win) can propagate through multiple rounds, altering the downstream matchups and probabilities.
- Interdependence: No component operates in isolation. For instance, the statistical profile of a team is meaningful only in the context of its opponents and match circumstances.

8.1.3 Analysis of Sensitivity and Chaos

A critical dimension of the systems analysis was the evaluation of sensitivity and chaos within the NCAA Tournament environment. These phenomena, intrinsic to complex systems, describe how small perturbations in initial conditions or system inputs can lead to disproportionately large or unpredictable outcomes, an essential consideration in the design of a resilient forecasting model.

Sensitivity Analysis
The system exhibits parameter sensitivity, whereby certain input variables have a high leverage effect on prediction results. For example:

- The unexpected injury or absence of a high-impact player can drastically alter a team's statistical profile and subsequent match outcomes.
- A misclassification or missing value in seed ranking can propagate through the bracket, causing significant divergence in predicted versus actual paths.

To account for these sensitivities, the system design emphasizes robust feature selection and the use of aggregated statistical indicators (e.g., season-long averages) to smooth short-term fluctuations. Moreover, sensitivity-aware strategies such as seed normalization and variance reduction in features were implemented during preprocessing.

Chaotic and Stochastic Behavior
In addition to sensitivity, the tournament environment exhibits chaotic characteristics—outcomes that emerge from interactions of many subsystems and are highly sensitive to initial or contextual conditions, but are difficult to model deterministically. Sources of chaos include:

- Referee decisions, which may introduce unquantifiable bias or error.
- Psychological factors, such as pressure, crowd reactions, or momentum shifts during games.
- Bracket dynamics, where early upsets alter the competitive landscape of future rounds.

While these elements cannot be predicted directly, their influence can be statistically attenuated. For instance, using ensemble models, cross-validation over multiple seasons, and probabilistic targets (e.g., win probabilities instead of binary outcomes) reduces overconfidence and promotes robustness under chaotic conditions.

8.2 System Design

Building upon the insights from the systems analysis, the second phase of the methodology focused on the design of the predictive system, as outlined in Workshop 2. This involved translating the analytical findings into a concrete architectural blueprint and a plan for implementation.

8.2.1 Definition of System Requirements

Translating the insights from systems analysis into actionable system specifications required a rigorous definition of both functional and non-functional requirements. These requirements serve as the blueprint for the architectural and implementation decisions that follow and ensure alignment with project objectives, technical feasibility, and the evaluation framework defined by the Kaggle competition.

Functional Requirements (FRs) The functional requirements define the core operational capabilities that the system must support. These were structured to reflect the modular pipeline architecture and include:

- FR1 – Game Outcome Prediction: The system must compute probabilistic predictions of match outcomes between two teams based on historical data, producing a likelihood estimate (e.g., P(Team A wins)).
- FR2 – Data Ingestion and Preprocessing: The system must be capable of reading, validating, and transforming structured datasets (e.g., CSV files) into clean, analysis-ready formats. This includes merging multiple sources, handling missing values, and aligning schemas across seasons.
- FR3 – Model Training and Validation: The system must support the training of machine learning models using labeled historical data. It must also allow for validation using cross-validation or temporal splits to evaluate generalization.
- FR4 – Output Generation for Kaggle Submission: The system must generate output in the format required by the Kaggle competition: a CSV file with two fields—ID (Season, Team A ID, Team B ID) and Prob (predicted probability of Team A winning).

Non-Functional Requirements (NFRs) Non-functional requirements describe system attributes that affect usability, maintainability, and performance. These include:

- NFR1 – Processing Efficiency: The system must complete data processing, training, and prediction within reasonable runtimes suitable for local execution or Kaggle kernels, ensuring responsiveness and reproducibility.
- NFR2 – Modular Architecture and Adaptability: The system must adopt a modular design allowing for independent updates or replacements of components (e.g., switching model types or modifying feature engineering strategies) without requiring full system redesign.
- NFR3 – Maintainability and Readability: The system codebase should follow clean coding practices and be appropriately documented to facilitate debugging, collaboration, and future enhancements.

- NFR4 – Reproducibility: The system must produce consistent results when executed on the same dataset and configuration, supporting academic rigor and competition reproducibility.

Together, these requirements operationalize the conceptual model into a concrete implementation strategy. They guide the architectural design presented in the next section and ensure that the forecasting system meets both the technical and contextual demands of NCAA tournament prediction.

8.2.2 Architectural Design

The architectural design of the NCAA tournament forecasting system follows a modular, pipeline-based architecture engineered to support flexibility, maintainability, and scalability. This design approach enables independent development, testing, and refinement of each processing stage, while maintaining a coherent data flow across the system.

The architecture is structured as a sequence of logically distinct but interoperable modules, each responsible for a specific phase in the prediction pipeline. The design promotes separation of concerns and allows for plug-and-play substitution of models or preprocessing strategies without altering the overall system behavior. The high-level architecture is illustrated in Appendix B.

Module 1: Ingestion Module. This component is responsible for importing and organizing raw data. It ingests structured files such as regular season results, tournament outcomes and team seeds.

Module 2: Feature Engineering Module. This module transforms raw data into a format suitable for model consumption. It computes derived variables that enhance predictive signal strength.

Module 3: Model Training Module. The training module encapsulates all logic related to fitting predictive models. It supports algorithmic strategies like XGBoost and also includes the Croos-validation.

Module 4: Prediction Module. This module applies trained models to test data to produce probabilistic predictions.

Module 5: Output Writer Module. This final module formats and exports predictions to match the requirements of the Kaggle competition:

The modularity of the architecture ensures that each component can be tested independently, facilitating debugging and future upgrades. It also supports reusability across future tournaments or predictive tasks, provided the system interfaces are preserved. This architectural approach thus balances performance with engineering flexibility—a crucial feature in dynamic, competition-driven environments such as NCAA forecasting.

8.2.3 Technological Stack Selection

The selection of the technological stack was guided by the system requirements, the nature of the data science task, and common practices within the Kaggle community. The core tools identified include :

Components	Tools
Language	**Python**: It allows handling large amounts of data using the Pandas and NumPy libraries.
Modeling	**Scikit-learn [2], XGBoost [1], Logistic Regression [4]**: They allow predicting probabilities, are simple, some use decision trees and are very powerful.
Intake	**Pandas**: Essential for handling structured data (CSV, data frames).
Cleaning and Feature Engineering	**NumPy**: Fast vectorized operations on arrays (e.g., subtract metrics between devices). Conversion to NumPy arrays to feed models (X_train, y_train). Direct calculation of Brier Score if necessary.
Testing	**Brier Score (MSE in probabilistic classification)**: A metric that measures the accuracy of probabilistic predictions, evaluating the difference between predicted probabilities and actual outcomes.
Output	`.csv` generated by Python script: Output with probabilities in the format (ID, Prob), required by the Kaggle competition.
Environment	**VS Code, Kaggle, GitHub**: VS Code for coding, Kaggle for model deployment, and GitHub for version control and workflow documentation.

Table 8.1: Summary of tools, components and environment

This stack provides a flexible and powerful environment for building the proposed predictive system.

8.2.4 Strategies for Addressing Complexity and Uncertainty

Given the inherently volatile and high-dimensional nature of the NCAA Basketball Tournament, the system design integrates several strategies specifically aimed at mitigating the effects of systemic complexity, parameter sensitivity, and stochastic events. These strategies are rooted in both systems thinking and best practices from predictive modeling, ensuring that the system maintains performance and robustness under uncertainty.

1. Integration of Domain-Informed Variables Incorporating tournament-specific knowledge was essential for increasing the model's sensitivity to structural and expert-driven factors. Two critical inputs were:
    - Tournament Seeds and Seed Differences: Seedings provide a proxy for team strength and expert ranking. Including both absolute and relative seed positions (e.g., Seed

A – Seed B) improved the model's ability to account for pre-tournament expectations and bracket placement.

2. Statistical Smoothing via Averaged Team Metrics To reduce the volatility caused by isolated outlier performances or small sample sizes, the system computes season-level average statistics per team. These smoothed variables include:

   - Average point differential
   - Scoring efficiency metrics (e.g., 3-point field goals, free throw success)

   This reduces the noise from short-term fluctuations and enhances generalization to new matchups.

3. Cross-Validation for Robust Generalization The system applies k-fold cross-validation, with optional time-aware or grouped splits, during model training. This allows for:

   - Detection of overfitting to specific tournament years or subsets.
   - Evaluation of model consistency across different historical contexts.
   - Estimation of real-world performance on unseen pairings.

   Cross-validation also facilitates ensemble averaging, where predictions from multiple folds are aggregated to improve stability.

4. Controlled Feature Set Experiments The simulation framework included a sequence of controlled experiments that incrementally introduced new features. This helped evaluate marginal utility and interaction effects between variables, avoiding overcomplicated models that incorporate noisy or irrelevant inputs.

5. Probabilistic Outputs and Calibration Rather than outputting binary win/loss predictions, the system generates probabilistic forecasts (e.g., P(Team A wins) $\in [0,1]$). These continuous predictions allow for:

   - Better performance under scoring metrics like the Brier score.
   - Compatibility with ensemble averaging and simulation pipelines.
   - Calibration diagnostics (e.g., reliability curves) to assess confidence accuracy.

8.3 System Implementation, Simulation and Evaluation

The final phase of the system development lifecycle consisted of the implementation, simulation, and evaluation of the forecasting pipeline. This phase aimed to instantiate the architectural blueprint previously defined, validate the design assumptions under operational conditions, and assess the system's predictive performance using both quantitative metrics and controlled experiments.

The implementation strategy adopted an incremental and modular development approach, aligning with agile principles to support iterative refinement, testability, and ease of debugging. Each module was constructed independently and integrated sequentially into the pipeline, allowing for early identification of bottlenecks, logical errors, or data inconsistencies.

In parallel, a simulation framework was designed to emulate the full end-to-end prediction workflow, starting from raw data ingestion and ending with the generation of competition-ready submission files. This framework was essential for:

- Stress-testing the system under varying data configurations and feature sets.
- Observing the behavior of the model under different input distributions.
- Measuring runtime efficiency, memory consumption, and output stability.

To assess the system's effectiveness, a combination of internal validation (cross-validation Brier score) and external benchmarking (Kaggle public leaderboard) was employed. These evaluation stages allowed for both controlled academic analysis and real-world competition relevance.

The following subsections detail the key steps in this phase: code implementation (8.3.1), evaluation methodology (8.3.2), data preparation strategies (8.3.3), simulation planning (8.3.4), and execution of forecasting scenarios (8.3.5).

8.3.1 Implementation

The implementation of the NCAA tournament forecasting system was carried out following a modular and iterative development methodology, guided by the architectural blueprint defined in earlier design stages. The system was developed primarily in Python, leveraging open-source libraries for data manipulation, modeling, and evaluation. Emphasis was placed on writing clean, maintainable, and testable code to support both experimentation and long-term extensibility.

Modular Construction Each functional component of the system was implemented as an independent module, with clear input/output interfaces and encapsulated logic. The pipeline was structured as shown in 8.2.2.

This modularization enabled parallel development, unit testing, and rapid iteration of model configurations and feature sets, without disrupting other parts of the system.

Development Tools and Environment The system was implemented using the core tools mentioned in 8.2.3

Agile-Inspired Workflow Although not formalized into sprints, the development followed agile principles such as iterative refinement, continuous integration of components, and feedback-based adjustments after each simulation cycle. Configuration files were used to manage model parameters and feature sets, improving traceability and reproducibility.

By implementing the pipeline iteratively and incrementally, the development team was able to validate each stage under controlled conditions before executing full-scale simulations.

8.3.2 Evaluation Methodology

To assess the performance and reliability of the NCAA forecasting system, a structured evaluation methodology was employed, combining both internal validation metrics and external benchmarking against the public Kaggle leaderboard. The goal was to ensure that the system not only performed well on historical data but also demonstrated strong generalization capability on unseen tournament scenarios.

Primary Evaluation Metric: Brier Score The central performance indicator used for both internal and competition-based evaluation was the Brier score, a metric specifically designed to evaluate the accuracy of probabilistic forecasts. It is defined as the mean squared difference between predicted probabilities and actual outcomes.

A lower Brier score indicates better model calibration and probabilistic accuracy.

Validation Strategy: To prevent overfitting and evaluate the model's generalization capability, the following internal validation technique was used:

- Cross Validation: A season was selected (starting from the oldest to the most current) and a prediction and training was made for that season taking into account the other seasons.

External Evaluation:
Kaggle Leaderboard: After internal validation, the system's final predictions were submitted to the March Machine Learning Mania 2025 Kaggle competition. The public leaderboard provided real-time feedback using the same Brier score metric, offering an external, standardized benchmark of the system's effectiveness relative to other participants.

Evaluation Scope
Performance evaluation covered not only accuracy but also:

- Stability across simulations (variance of scores across feature sets).
- Computational efficiency (runtime per module).
- Robustness to input perturbations (sensitivity testing with reduced or noisy features).
- Interpretability of feature importance (e.g., seed ranking effect sizes).

This multi-faceted evaluation strategy ensured that the system's results were not artifacts of overfitting or hyperparameter tuning, but rather reflected true learning from historical tournament dynamics.

This comprehensive methodology, combining systems analysis with a structured design process, provides a solid foundation for developing an effective system to forecast NCAA Basketball Tournament outcomes.

8.3.3 Data Preparation

The first phase of the implementation was the data preparation, the datasets were cleaned and reduced to only the most essential information. The data identified as fundamental included tournament results, regular season results for both men and women, and team seed rankings. These elements were selected because they provide a clear overview of each team's historical performance. Tournament results reflect how teams perform under pressure in elimination games, while regular season results offer a broader picture of consistency and overall strength. Seed rankings, assigned before the tournament begins, indicate expert expectations and can serve as a useful reference for comparing actual outcomes. Focusing on these core datasets helps simplify the system without losing valuable insights, ensuring that the predictive model is built on relevant and meaningful information.

Data preparation was a critical foundational step in the implementation of the forecasting system. Given the high dimensionality and variability of sports datasets, a careful curation and transformation process was necessary to ensure the predictive model was trained on relevant, consistent, and interpretable features. The objective was to balance information richness with computational efficiency and model robustness.

Core Dataset Selection

From the broader data repository provided by the Kaggle competition, only a subset of the most predictive and interpretable datasets was retained for modeling. The selected datasets included:

Regular Season Results (Men's and Women's): Provided comprehensive game-level data including team IDs, scores, and locations, used to compute season-long performance metrics.

Tournament Results (Men's and Women's): Captured the outcome of elimination games, serving as ground truth labels for supervised learning during training.

Team Seeds (Men's and Women's): Contained expert-derived rankings for each team, which were later used to generate both raw seed features and seed differentials between matchups.

These datasets were prioritized due to their temporal alignment, direct relevance to match outcome prediction, and availability across multiple seasons, which enabled robust cross-season validation.

8.3.4 Simulation Planning

To evaluate the system's behavior under varying input conditions and modeling configurations, a simulation framework was designed to emulate realistic prediction workflows and test the system's modular performance. This simulation phase was essential for assessing how different combinations of features, model assumptions, and variable groupings affected predictive accuracy, stability, and computational cost.

Simulation Objectives
The simulations were designed to:

- Test the end-to-end functionality of the system pipeline—from data ingestion to submission output.
- Evaluate the incremental impact of variable sets (e.g., seed features, scoring stats) on model performance.
- Validate the robustness of predictions under different feature reduction strategies.
- Emulate realistic tournament forecasting scenarios, simulating unknown matchups under future data.

Controlled Variable Selection
Each simulation was structured to vary one or more dimensions of the input space or modeling pipeline, including:

- Feature Sets: Different subsets of variables were selected to analyze their relative predictive contribution (e.g., full stats vs. reduced stats vs. seed-only features)..
- Training Horizon: Different historical season ranges were used for training to simulate drift-resilience and temporal generalization.

Exclusion of Contextually Weak Features
Although certain features—such as game location (city) or event attendance—may offer marginal signal in other sports contexts, they were deliberately excluded from the simulation design for the following reasons:

- Neutral Court Effect: Most NCAA tournament games are played at neutral venues, minimizing home-court advantage.

- Simulation Focus: The goal was to isolate performance-based variables to evaluate their pure statistical signal in win probability estimation.

The simulation framework functioned as both a diagnostic and exploratory tool, revealing insights about feature importance, model sensitivity, and the dynamics of prediction under uncertainty. These findings directly informed the final implementation choices documented in the next section.

8.3.5 Executing the Simulation

To validate the effectiveness of the forecasting system and quantify the impact of different feature configurations, four structured simulations were executed. Each simulation tested a distinct set of input variables and modeling assumptions, allowing for a comparative analysis of performance using the Brier score as the evaluation metric.

Simulation 1 – Full Statistical Feature Set
In the baseline simulation, the model was trained using an exhaustive set of aggregated team statistics. These included:

- Average points scored
- Field goal metrics (FGM, FGA, FG3M)
- Rebounds, assists, turnovers, and other season-long averages

Objective: Establish a baseline performance using the richest available dataset.
Result: Kaggle Brier Score: 0.16903
Observation: While the model benefited from a large number of features, the inclusion of all variables did not necessarily improve predictive accuracy, possibly due to feature redundancy and increased noise.

Simulation 2 – Reduced Statistical Feature Set
This simulation limited the input variables to a smaller, curated subset of team statistics.
Objective: Evaluate the predictive power of a minimal, interpretable feature set.
Result: Kaggle Brier Score: 0.16256
Observation: A slight improve in performance indicated that the full feature set was noisy, and maybe contained redundancy or similar.

Simulation 3 – Seed-Based Features and set of simulation 2
In this iteration, seed information and seed differences between the two teams were used as input features, and the variables set of simulation 2 were used too.
Objective: Include the contribution of expert-derived rankings to predictive accuracy.
Result: Kaggle Brier Score: 0.11929
Observation: Seed-based features significantly improved performance, highlighting their strong correlation with match outcomes and tournament structure. This confirmed that domain-informed variables can outperform purely statistical aggregates.

Simulation 4 – Combined All Statistical + Seed Features This final simulation integrated the full set of team performance statistics with seed-based variables, aiming to capture both empirical performance and expert rankings.

Objective: Evaluate the synergy between historical performance metrics and expert judgment.

Result: Kaggle Brier Score: 0.11790

Observation: This was the best-performing configuration, demonstrating that combining quantitative season data with seed-based context yields a more holistic and accurate forecast.

Conclusion of Simulation Execution

These experiments demonstrate the importance of feature selection and domain knowledge integration in predictive modeling. The simulations also validated the pipeline's flexibility: by simply modifying input configurations and model parameters, the system was able to adapt to different levels of complexity and data abstraction.

These findings guided the final configuration of the production model and informed the discussions and visual analyses presented in subsequent chapters.

Chapter 9

Results

The analysis of the NCAA Men's and Women's Basketball Tournament systems resulted in the identification of key elements and their relationships, as depicted in A.1. This diagram illustrates the complex interactions between various components of the system, including teams, coaches, matches, and external factors such as fans and referees.
The design phase produced a high-level architecture for the prediction system, as shown in B.1. As mentioned above, this architecture comprises five main modules:

- Data Ingestion
- Data Cleaning and Feature Engineering
- Model Training
- Prediction Generator
- Output Writer

The system design incorporates several strategies to address the challenges of complexity and uncertainty in tournament prediction:

- **Inclusion of Seeds and Slots:** Incorporating official rankings and tournament assignments as predictor variables.
- **Averaging of Team Statistics:** Using averaged historical statistics to mitigate the impact of random fluctuations in team performance.
- **Cross-validation:** Employing cross-validation techniques to reduce overfitting and improve the model's generalization ability.

It became evident, as previously mentioned, that seeds played a significant role in the system. as shown in Figure 9.1 and Figure 9.2.

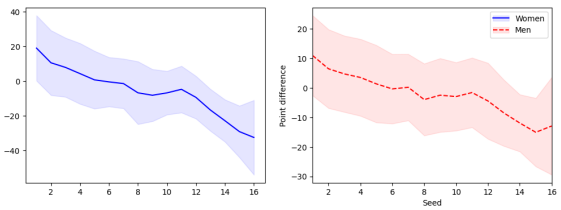

Figure 9.1: Point Difference vs Team 2 Seed

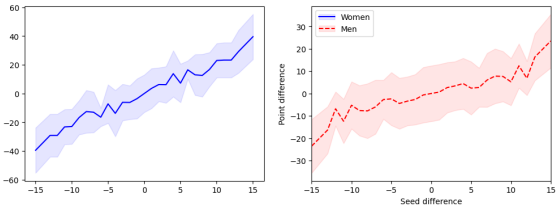

Figure 9.2: Point Difference vs Seeds Difference

The subsequent action taken was to create a graphic displaying the point difference and its win probability for men and women:

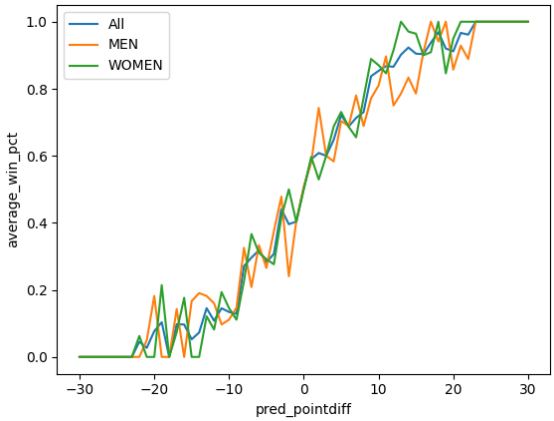

Figure 9.3: Average Win Prob vs Point Difference

And, as expected, the win probability increases as the point difference does, too.

Finally, another graphic was made, but with the predicted point difference vs the win probability. The probability of each game was determined using the aforementioned graphics, with the calculation of this probability being contingent upon the predicted point difference of the respective games.

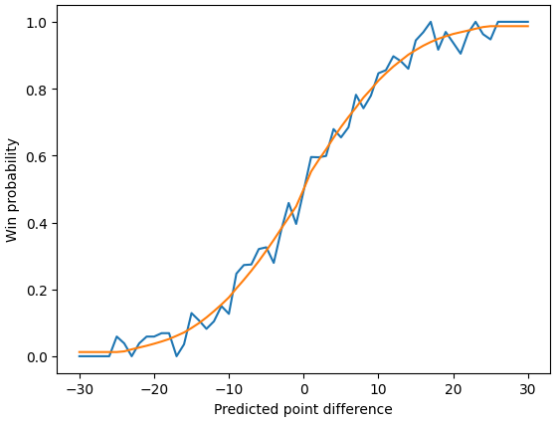

Figure 9.4: Win Prob vs Predicted Point Difference

Feature-Level Contributions Several design strategies significantly improved the model's predictive accuracy, particularly:

Seed Inclusion and Seed Differentials: The integration of official seed rankings and their differences between matchups emerged as the most powerful individual predictors. These features encoded both expert judgment and structural tournament constraints, enabling the model to approximate bracket logic effectively.

Averaged Statistical Metrics: Team-level season averages (e.g., point differential, scoring efficiency) helped reduce the impact of outlier games and stabilized performance across simulations. While not as influential as seed data alone, they provided complementary information that enhanced generalization.

Combined Feature Sets: The best results were obtained by combining historical statistics with seed-related features, confirming the hypothesis that both performance-based and expert-driven signals are necessary for robust tournament forecasting.

Performance Analysis (Simulation Summary)

Simulation	Feature Set	Kaggle Brier Score
Sim 1	Full statistical features	0.16903
Sim 2	Reduced statistical features	0.16256
Sim 3	Seeds and seed differences + Sim 2 variables set	0.11929
Sim 4	Combined all stats + seed-based features	0.11790

Table 9.1: Simulation Summary

This progression demonstrates that while a wide feature set alone does not guarantee performance, incorporating domain-specific information (e.g., seed rankings) substantially improves model output.

Chapter 10

Discussion

The analysis of the NCAA Men's and Women's Basketball Tournament systems revealed the inherent complexity of predicting tournament outcomes. The identification of key elements and their relationships highlights the numerous factors that can influence game results, ranging from team dynamics to external influences like fan behavior and referee decisions.

The high-level architecture designed for the prediction system provides a structured framework for addressing this complex problem. The modular design, with distinct modules for data ingestion, cleaning, model training, prediction generation, and output writing, promotes maintainability, scalability, and ease of debugging. This modularity also allows for flexibility in incorporating different machine learning models or data sources without requiring a complete system overhaul.

The strategies implemented to handle complexity and uncertainty are crucial for developing a robust prediction system. Encoding game locations, including seed and slot information, and averaging team statistics are all attempts to account for factors that introduce variability and potential bias into the predictions. Furthermore, the use of cross-validation is essential for mitigating overfitting and improving the model's ability to generalize to unseen data.
It is important to acknowledge the limitations of the system. The accuracy of the predictions is inherently constrained by the quality and availability of the input data, as well as the unpredictability of sports events. While the system design incorporates strategies to address these challenges, it cannot eliminate them entirely.

Overall, the designed system provides a strong foundation for predicting NCAA Men's and Women's Basketball Tournament outcomes. The modular architecture and the implemented strategies for handling complexity and uncertainty offer a comprehensive approach to this challenging problem.

And lastly, based on the results of the four simulations and the visual and statistical analysis performed, several key findings emerged:

1. **Full Stats Set (Simulation 1):** Using all average team statistics yielded a moderate performance (Kaggle score: 0.16903). It established a baseline but showed that quantity does not guarantee quality in feature selection.

2. **Reduced Stats Set (Simulation 2):** Limiting the variables to a smaller, curated set improved performance slightly (score: 0.16256), suggesting that some less obvious statistics may not provide as much value.

3. **Seeds and Seed Difference (Simulation 3):** Introducing seed data and seed difference resulted in a significant improvement (score: 0.11929). This reinforced the hypothesis that seed rankings are highly predictive in tournament outcomes.

4. **All Stats + Seeds (Simulation 4):** Combining full statistics with seed information gave the best result (score: 0.11790), confirming that both historical performance and tournament rank are essential for accurate predictions.

5. **Win Probability Correlation:** Graphs confirmed a strong relationship between point difference and win probability. This validated the use of `PointDiff` as a predictive target and justified its central role in model training.

6. **Seed Impact Visualization:** Visuals comparing seed values with point difference showed clear trends, highlighting the practical importance of seeding beyond just numerical input.

Chapter 11

Conclusions

This project presented the complete analysis, design, implementation, and evaluation of a forecasting system for the NCAA Men's and Women's Basketball Tournaments. Through the integration of systems engineering principles and machine learning methodologies, a robust and modular solution was developed to address the inherent complexity and stochasticity of tournament outcome prediction.

The systems analysis phase enabled a comprehensive decomposition of the NCAA tournament structure, identifying critical elements such as teams, seeds, matches, and contextual dynamics. By modeling the interactions among these components, the design process captured emergent behaviors and dependencies essential to building a high-fidelity predictive model.

The forecasting system was implemented as a modular data pipeline, composed of ingestion, feature engineering, modeling, prediction, and output modules. The architecture facilitated experimentation and adaptability, allowing for seamless modifications to the feature set, model type, or evaluation strategy.

Simulation results demonstrated the effectiveness of combining expert-informed variables (e.g., seed rankings) with aggregated historical performance data. This hybrid approach yielded the best predictive performance, confirming the value of both empirical signals and domain knowledge in complex systems.

While the system showed strong accuracy and stability, its performance is inherently limited by factors such as the unpredictability of live sports, incomplete data about player-specific conditions, and temporal drift across seasons. These challenges highlight the boundaries of data-driven models and the need for continual adaptation and contextual understanding.

Ultimately, the system achieved its goal of producing accurate, interpretable, and competition-compatible forecasts, providing a valuable proof of concept for the application of systems thinking and data science in sports analytics.

11.0.1 Future Work

- Refinement of Predictive Models: Explore and implement more advanced machine learning algorithms or ensemble methods to potentially improve the accuracy of the predictions. This could involve techniques such as deep learning, recurrent neural networks, or more sophisticated feature engineering.
- Incorporation of Additional Data Sources: Include other potentially relevant data sources, such as player statistics, coach statistics, or news articles, to provide a more comprehensive view of the factors influencing tournament outcomes.
- Analysis of Real-time Data: Investigate the use of real-time data during the tournament

to update predictions dynamically. This could involve incorporating information such as game momentum, player injuries, or crowd reactions.

- Evaluation of Model Performance: Conduct a thorough evaluation of the model's performance over multiple tournaments to assess its robustness and generalization capabilities.
- Comparison with Other Prediction Methods: Compare the system's predictions with those of other existing methods or experts to identify strengths and weaknesses.

References

[1] Nvidia, "What is XGBoost?," NVIDIA Data Science Glossary, 2024. https://www.nvidia.com/en-us/glossary/xgboost/

[2] Scikit-learn, "scikit-learn: Machine Learning in Python," Scikit-learn.org, 2024. https://scikit-learn.org/stable/

[3] "Kaggle," Kaggle.com, 2025.https://www.kaggle.com/competitions/march-machine-learning-mania-2025/overview (accessed May 16, 2025).

[4] V. Kanade, "What Is Logistic Regression? Equation, Assumptions, Types, and Best Practices," Spiceworks, Apr. 08, 2022. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/

[5] [1]"Educative Answers - Trusted Answers to Developer Questions," Educative. https://www.educative.io/answers/classification-using-xgboost-in-python

[6] GeeksforGeeks, "What is High Level Design – Learn System Design," GeeksforGeeks, Feb. 02, 2023. https://www.geeksforgeeks.org/what-is-high-level-design-learn-system-design/

Appendix A

System Analysis Diagram

This appendix contains the system element relationship diagram, originally presented in the Systems Analysis phase of the project. This diagram (Figure A.1) illustrates the key components of the NCAA Basketball Tournament systems and their complex interdependencies. Understanding these relationships is crucial for designing an effective prediction system.

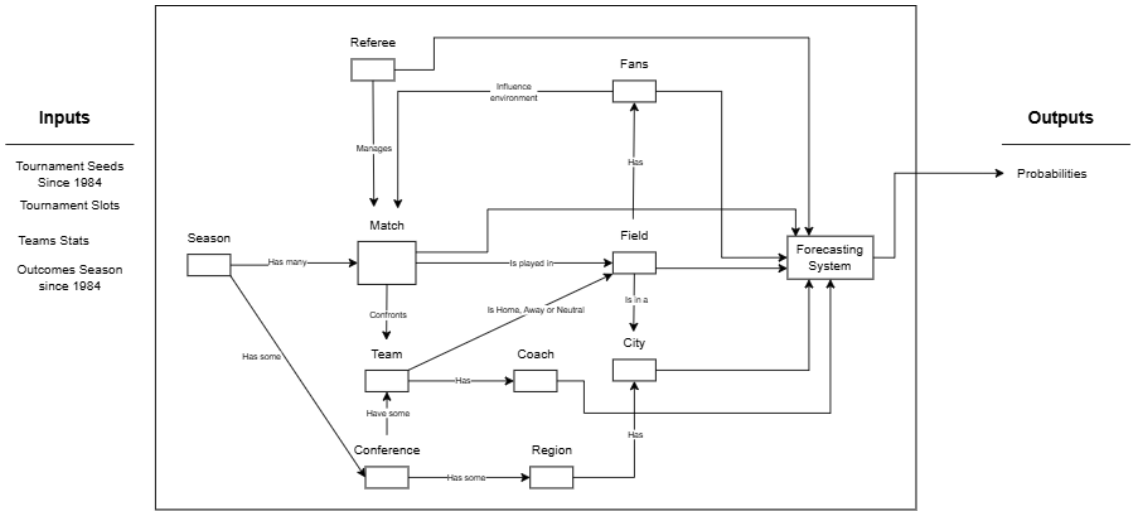

Figure A.1: System Element Relationship Diagram

Appendix B

High Level Architecture

This design illustrates the high-level architecture of the NCAA basketball tournament prediction system. The design follows a modular workflow consisting of key stages: data ingestion, preprocessing, feature engineering, model training, evaluation, and output generation. Each component interacts sequentially, ensuring data integrity and facilitating independent testing and updates.

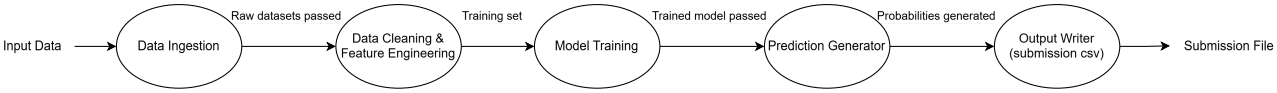

Figure B.1: High Level Architecture

Chapter 12

Acknwoledgements

The authors would like to extend their sincere appreciation to Eng. Carlos Andrés Sierra, M.Sc., for his continuous guidance, technical feedback, and critical insights throughout the development of this project. His expertise in systems analysis and software engineering provided a strong academic foundation and greatly enriched the quality of our methodology and implementation.

We are also grateful to the Universidad Distrital Francisco José de Caldas, particularly the Systems Engineering Department, for providing the infrastructure, academic support, and learning environment that enabled this project to be carried out with methodological rigor and interdisciplinary depth.

Additionally, we acknowledge the organizers of the "March Machine Learning Mania 2025" Kaggle competition, whose platform, datasets, and evaluation framework served as both a practical challenge and a real-world validation environment for our system. Their commitment to open data science played a pivotal role in shaping the applied dimension of this research.

Finally, we thank our peers and collaborators for their feedback, shared insights, and constructive discussions, which helped refine the scope and technical clarity of our work.

Chapter 13

Glossary

- NCAA (National Collegiate Athletic Association):The governing body for college athletics in the United States. It organizes the annual NCAA Men's and Women's Basketball Tournaments, which follow a single-elimination format.
- Kaggle: A platform for data science competitions and collaborative modeling. It hosts the "March Machine Learning Mania" competition, providing datasets and evaluation metrics for tournament prediction.
- Machine Learning (ML): A subfield of artificial intelligence focused on the development of algorithms that can learn patterns from data to make predictions or decisions without explicit rule-based programming.
- Tournament Seed: A ranking assigned to teams before the NCAA tournament begins, typically based on regular season performance. It reflects expected strength and determines placement in the tournament bracket.
- Tournament Slot: The specific position in the bracket assigned to a team, which dictates potential matchups and tournament progression path.
- Feature Engineering: The process of selecting, aggregating, and transforming raw data into informative variables (features) suitable for use in machine learning models.
- Overfitting: A modeling error where the model performs well on training data but fails to generalize to unseen data due to excessive sensitivity to noise or specific patterns.
- Cross-Validation: A resampling method used to evaluate a model's performance by partitioning the data into multiple training and validation sets, thereby testing robustness and generalization.
- Brier Score: A scoring rule used to measure the accuracy of probabilistic predictions. It calculates the mean squared error between predicted probabilities and actual binary outcomes.
- System Architecture: The high-level structural design of a software system, including its modules, data flow, and inter-component interactions.
- Sensitivity (in Systems): The degree to which small changes in inputs or environmental conditions lead to significant variations in system output.

- Chaos (in Systems): Describes unpredictable or emergent behavior resulting from the nonlinear interaction of system components, often beyond the reach of deterministic modeling.

Contents

List of Figures

List of Tables