Universidad Distrital Francisco Jose de Caldas

Faculty of Systems Engineering

Basketball Tournament Prediction System
Technical Report

Nicolas Romero Rodriguez - 20222020023
Carlos Andres Celis Herrera - 20222020051

Xiomara Salome Arias Arias - 20222020028

Professor: Carlos Andres Sierra Virgüez

Systems Engineering

July 12, 2025

Acknowledgements

We would like to express our most sincere thanks to our teacher Carlos Andres Sierra Virgüez. For his support throughout our learning process, we would like to highlight his guidance and tools provided to make this project a success.

Contents

List of Figures

List of Tables

Glosary

CSV	Comma-Separated Values. A file format used to store tabular data. In this project, 37 '.csv' files were used containing historical tournament statistics.
Pandas	A Python library used for data manipulation and analysis. It is used to load, clean, combine, and export data throughout the system.
scikit-learn	A Python machine learning library. It is used for data normalization and implementing the logistic regression model.
Logistic Regression	A statistical model that estimates the probability of one team winning over another based on historical performance variables.
StandardScaler	A tool from 'scikit-learn' used to normalize numeric features, reducing the impact of scale differences and data noise.
Feature Engineering	The process of selecting and creating relevant variables (features) used as input for the prediction model, such as seed difference or scoring averages.
Preprocessing	The stage where raw data is cleaned, filtered (e.g., removing years prior to 2015), and formatted for further processing.
Bracket Simulator	A module that simulates the progress of a single-elimination tournament, determining which teams advance in each round based on predicted outcomes.
Prediction Engine	The core system component that predicts match outcomes using logistic regression based on cleaned and processed data.
Abstract Factory	A design pattern used to generate flexible and scalable bracket structures that adapt to various team configurations and tournament formats.
Chaos Factors	Uncontrollable and unpredictable events (e.g., injuries, referee bias, delays) that may affect the real match results but are not modeled in the system.
Systems Thinking	A holistic approach that views the system as a set of interrelated components working together toward a shared goal.
Traceability	The ability to track data and processing flow across modules, ensuring transparency and making debugging and validation easier.
Seed	A ranking or classification assigned to a team based on its past performance, used to position teams in the tournament bracket.
Matchup	A simulated or real game between two teams in the tournament.
Elimination Tournament	A competition format in which teams are removed from the tournament after a single loss.

NCAA	National Collegiate Athletic Association.The organization that manages college sports tournaments in the United States, including the annual basketball championship.
March Madness	A popular name for the NCAA college basketball tournament, known for its unpredictability and high-stakes single-elimination format.
Win Rate	The percentage of games won by a team over a season or set of matches, used as a key feature in the prediction model.
Defensive Efficiency	Metrics that evaluate a team's performance on offense and defense, typically calculated per possession.
Complex System	A system composed of interconnected and interdependent components whose behavior as a whole cannot be simply predicted from the behavior of the individual parts.
Open System	A system that interacts with its environment and is influenced by external factors, making it adaptable but also susceptible to unpredictable changes.
Non-linear System	A system in which outputs are not directly proportional to inputs. Small changes in one component can produce large, sometimes chaotic, effects on the overall system.
Feedback Loop	A process where the system's outputs are fed back as inputs, influencing future behavior. In tournaments, match outcomes affect the structure of future matchups.
Adaptability	The system's ability to adjust to internal or external changes, such as new results or altered inputs, without breaking its logic or structure.
Deterministic Model	A model that behaves in a predictable, predefined way, producing the same output for a given input, with no randomness involved.
Linear System	A system where the relationship between inputs and outputs is direct and proportional, often assumed in simplified or ideal models.
Emergent Behavior	Unpredictable patterns or outcomes that arise from interactions between system components, common in complex or non-linear systems.

Abstract

The following technical report contains the project definition, the theoretical basis and technical aspects of the system analysis and design, the proposed solution, and the results. It also presents the design and implementation of a predictive system to estimate the results of university basketball tournaments, following the context of the Kaggle March Madness competition. The system simulates a single-elimination tournament with 64 teams using historical data from 1984 to 2024, distributed across 37 .csv files. A logistic regression model is used to estimate match results based on characteristics such as win percentage, average score, and ranking difference. The architecture is modular and scalable, and includes stages of data ingestion, preprocessing, feature engineering, prediction, and tournament simulation. Although the system does not include external chaotic factors such as injuries or referee bias, it effectively demonstrates the value of machine learning and systems thinking in modeling complex and dynamic environments.

Chapter 1

Introduction

March Machine Learning Mania is one of the hundreds of competitions, which can be found in Kaggle. A platform focused on data science and March Machine Learning. This competition consists of implementing a machine learning model capable of generating predictions of the results of the annual college basketball tournament organized by the National Collegiate Athletic Association using the tournament's historical data as the basis to generate the predictions. Specific requirements for the competition include that the system's output has to be given in terms of the probability of each team winning in a specific match-up and as such these probabilities ar given in terms of numbers between 0 and 1. Additionally the competition requires that the proposed solution generates predictions for every possible matchup of every team selectable for the tournament. This perdictions are evaluated by calculating the squared error between the predicted outcome and the actual outcome of the real life match

In terms of formatting, the competition gives the historical data in the form of CSV files that contain information dating from 1984 to 2024 of team-wide and individual performance of all teams that have played on the tournament, this information contains statistics such as number of defensive assists in a game, or number of tree point throws made in an overall season. These datasets also contain information on where was each match played or who was coaching which team in a given season. Such a large amount of data forces participants to make decisions on which parts of this historical information are useful to generate predictions and which should be discarded.

It's important to take into account that this tournament can be considered as a complex system, since it is constituted by a large amount of elements and therefore many interactions among them. Likewise, some of these elements are subsystems that build the tournament system, adding additional layers of complexity.

With this, the purpose of the project is to design and implement an autonomous prediction system, which can make an estimation to reality. With the use of historical data and learning techniques. Moreover, to make use of fundamental concepts related to systems theory, to achieve a system design capable of modeling and analyzing the behavior of complex systems. With respect to the above, this document seeks to present a clear perspective of the development of the project. Emphasizing the information references, the methodologies used, scope, limitations, conclusions and future work. In order that, any person who tries to replicate the work, can be guided by the information condensed here.

Chapter 2

Literature Review

For the development of the project we used information related complex systems, March Madness and Kaggle's competitiosns. With which we defined and contextualized the designs, descriptions and documentation. The concepts and articles we used were the following.

- **March Madness** by NCAA (2025): March Madness is the NCAA Division I men's basketball tournament, with a single-elimination format. Sixty-eight teams compete through seven rounds to crown a national champion. The schedule is divided as follows: First Four, first round (64), second round (32), Sweet 16, Elite Eight, Final Four, and championship. Teams are selected in two different ways:
  - **Automatic Selections:** Thirty-one teams are selected and receive an automatic berth in the tournament for winning their conference tournament. For example: The champion of the Big Ten, ACC, SEC, Pac-12, and other conferences earns a direct berth. This means that even if a team did not have a strong regular season, if it wins its conference tournament, it secures its entry.
  - **At-Large Selections:** The other 37 teams are chosen by the NCAA Selection Committee, which is made up of 10 members (commissioners and athletic directors from Division I conferences). This selection is not made by lottery, nor is it automated: it is an intensive deliberation based on performance metrics and qualitative analysis.

  March Madness is one of the most anticipated sporting events of the year, with massive media coverage, millions of brackets filled out, and upset games or "Cinderella runs" popular among fans and analysts alike.
  Detailed knowledge of the March Madness tournament team selection process is essential for the analysis and design of the predictive system, as it allows us to identify the key variables that influence each team's inclusion and positioning in the bracket. Understanding wins by quadrant, schedule strength, and other statistical indicators facilitates the design of a model that not only predicts game results but also anticipates which teams will be selected and in what position (seed) they will be placed. This allows the system to simulate the tournament from its earliest stages in an accurate and informed manner, integrating both quantitative and qualitative elements that reflect the actual behavior of the tournament and improve the quality of the predictions.

- **Complex Systems** by University of Waterloo (n.d.): A complex system is a network of interrelated elements whose collective behavior cannot be explained solely by the individual properties of its components. These systems exhibit:

  - **Non-linearity:** pequeñas causas pueden generar grandes efectos, y viceversa.
  - **Feedback:** Actions within the system can reinforce or counteract each other over time.
  - **Emergency:** New or unexpected properties arise from the interaction of the elements.
  - **Self-organization:** Systems can develop internal order without central control.
  - **Adaptation:** Systems evolve or adjust according to the environment or interactions.

  Rather than mechanistic or deterministic approaches (such as those used for simple systems), complex systems are best understood through models that capture their dynamic interactions and evolving patterns, recognizing that the whole is greater than the sum of its parts.
  Understanding complex systems, as detailed in University of Waterloo (n.d.), provides an essential conceptual foundation for designing a system to predict March Madness tournament results. This tournament can be interpreted as a complex system in which teams, coaches, performance metrics, tactical decisions, and contextual factors interact in a nonlinear manner, generating emergent behaviors such as upsets or Cinderella runs. Incorporating principles such as feedback, adaptation, and self-organization allows for the design of a flexible and realistic system that not only uses traditional statistics but also models the interaction between multiple variables in real time.

- **March Machine Learning Mania** by Kaggle (2025) The objective of the competition is to predict the results of all possible matchups in the 2025 NCAA men's and women's tournaments. Predictions are evaluated using the Brier Score, equivalent to the mean square error applied to probabilities, rewarding well-calibrated models. For this purpose, a dataset containing relevant information from previous tournaments and, more importantly, from the teams is made available.
  Includes 35 .CSV files with historical information on teams, seeds, regular season and tournament results (compact and detailed mode), geographic locations, and public rankings such as Massey, Sagarin, and RPI.
  The structure and breadth of the Kaggle "March Machine Learning Mania 2025" dataset provide a solid foundation for your predictive system, as they bring together essential data ranging from team seeds, game results and statistics, updated locations and rankings, to a standard format for training and validating probabilistic models using Brier Score.

Chapter 3

Background

A system can be defined as the a group of interacting or interrelated elements that function together as a whole to achieve a common purpose, within this concept some categories can be identified, one of these and the most relevant for the project presented is the category of complex systems.

Complex systems are characterized by having numerous interconnected components that interact with each other. This interactions often lead to unexpected outcomes and behaviors, these are known as emergent behaviors. Complex systems are often open, which means that the system interacts and is affected by the environment that surrounds it, which makes it so that is harder to predict the output of the system. Another important concept is that systems behavior changes with the time, those that fall into this category are known as dynamic systems. Finally it's also important to take into account that due to the large amount of elements and interactions that compose a complex system they tend to be very sensitive to initial conditions, this means that even if the initial output has a slight variation the output can have an enormous variation. Having all of these labels combined in a system makes it very difficult to understand its functioning and therefore makes it very difficult to predict the outcome of a given input. In order to understand these systems it becomes necessary to apply advanced analytical techniques that can capture patterns within these large systems.

Sports tournaments, such as the NCAA Division I Men's and Women's Basketball Championships (commonly referred to as March Madness), are examples of complex systems. These tournaments involve dozens of interacting elements, including teams with diverse characteristics, unpredictable match outcomes, and organizational rules that determine progression. Factors such as team composition, historical performance, seeding, and game location influence results in ways that are difficult to capture through simple statistical methods. Additionally, chaotic elements—such as last-minute injuries, referee decisions, or environmental conditions—contribute to non-determinism, further complicating predictive efforts.

Machine learning is a branch of artificial intelligence that focuses on training software to replicate the way human beings learn so that the model can complete tasks autonomously and improve its performance through exposure to data (IBM, s.f) Usage of this tool can be useful to find patterns on large volumes of data that can later be used to generate predictions. On this case Logistic regression is used as a classification algorithm that identifies and generates a prediction on which team would win and which would lose taking into account all variables given to the model. In order to increase the accuracy of the predictions, the data given to the system has to undergo a preprocessing stage, where information considered irrelevant for the

prediction is cut from the dataset, and information useful for the prediction is rearranged in a structured format that the model can process effectively.

Chapter 4

Objective

The primary objective of this research is to design and implement an autonomous prediction system capable of forecasting the outcomes of the NCAA March Madness basketball tournament using historical data. This system will take into account all of the possible match-ups between every eligible team for the tournament and generate the probabilities of each team winning in terms of numbers between 0 and 1

This system aims to use logistic regression as a predictive model, supported by preprocessing, feature engineering, and evaluation metrics to ensure reliable results. Once this initial system is implemented its performance will be measured by using the rating provided by the kaggle competition and comparing it to the rating of another sumbission implemented by a llm, using these metrics as a baseline for the comparison the project aims to identify differences that lead to one implementation performing better than the other.

More specifically this study seeks to explore the effectiveness of machine learning techniques focusing particularly on logistic regression for modeling a complex system such as a basketball tournament. Second, to develop a pipeline that processes raw historical data into structured inputs suitable for prediction, ensuring the model captures key performance indicators. By fulfilling these objectives, the research intends to demonstrate how system analysis and machine learning can be integrated to model and predict behaviors within complex, non-deterministic environments. The project also aims to showcase the value of systems engineering principles in building modular, mantainable, and extensible systems.

Chapter 5

Scope

This research focuses on the design, development, and evaluation of a prediction system capable of forecasting the results of games to be played in the NCAA basketball tournament, both in the women's and men's first divisions, in the 2025 edition of the tournament. In developing this study, concepts from general systems theory and systems engineering are explored and applied to model the behavior of a complex system such as the tournament. Historical data, rules, models, and conditions set forth by the competition are taken into account for the design and implementation of a machine learning model that uses logistic regression to generate predictions.

The scope of this project is limited to the use of historical data provided in .csv format, specifically that supplied by the March Machine Learning Mania 2025 competition hosted on Kaggle. The system considers statistical information at the team and player level, covering regular seasons and previous tournaments since 2015. It includes data ingestion, preprocessing, and transformation processes, as well as feature engineering based on these statistics. Subsequently, all possible combinations of matchups between teams eligible for the tournament are generated, and a logistic regression model is applied to estimate each team's probability of victory in a given matchup.

It should be noted that this research excludes any form of real-time updating and does not consider the integration of dynamic data or external sources such as social media, sports APIs, or sensors. Manual intervention in the predictions generated by the system is also not permitted. Furthermore, the study does not extend its application to sports, leagues, or competitions other than the 2025 NCAA basketball tournament.

Chapter 6

Assumptions and Limitations

To make the implementation of this project possible is necessary to make certain assumptions. First it is assumed that the historical data provided by the competition that includes both individual and team-wide statistics since 1998 are accurate and complete as it is not possible to verify or take new samples of this data and the competition doesn't allow to directly insert into the system data sources external to the files provided initially as inputs. This design also assumes that past teams and player performance is representative of the dynamics if the most recent matches and can serve as a reliable source of information to generate predictions despite the possible variability caused by player turnover, coaching changes, shifts in team strategies or player lesions

This research is also subjected to many limitations. For instance, this system is heavily reliant on historical data, which may not be completely accurate, contain outdated information or may fall short when capturing important recent developments such as injuries, roster changes or variations in team dynamics taking place in the current tournament. There are also many variants related to the tournament environment that can also affect matches results such as delays in schedule, resting time or time spent traveling and team's morale that are not possible to be taken into account. Which forces us to ignore most of the chaos theory related variables, limiting to a certain extent the precision of the model's predictions and the possibility of representing a complex system to its whole extent.

Chapter 7

Methodology

The methodology of this report aims to present the process carried out to solve the March Machine Learning Mania 2025 competition, proposed by the platform Kaggle (2025). A systems thinking approach was implemented, along with its corresponding concepts, as these ensure finding the best solution to the problem. This workflow could be summarized as an initial conceptual design that aims to generate understanding of the tournament's general functioning, an architecture design that describes the modules and interconnections necessary to implement a functional prediction system, a simulation process where the functioning of the proposed modules are tested and a final implementation that is evaluated with the Kaggle evaluation system and by comparing its performance with a llm-based solution

7.1 System Design

As a first approach to solving the proposed problem was to propose a system representation of a basketball tournament in general, this representation allows us to understand how each component of the tournament interacted with one another, in addition to identify which variables contribute to introduce randomness into the system's output, the knowledge collected in this phase would serve as the foundation of the architectural design and its later implementation.

7.1.1 Conceptual System Design

The initial development phase consisted of designing the tournament system from a functional and systemic perspective. This conceptual model was represented in a diagram that maps the key elements: teams, organizer, rules, match logic, prediction engine, and unpredictable or random external factors (such as player injuries or referee bias). These components interact dynamically and form a feedback loop in which each match result influences the progression of the tournament.

This design treats the tournament as a complex system in which multiple subsystems interact and influence each other. Elements such as the organizer and the match schedule have a cascading effect on other components. By understanding these interactions, the system can better simulate the tournament dynamics and define clear input-output relationships for predictive analysis.

In this proposed representation it's identified that the organizers define the general flow of the tournament, including the formation of brackets, number of players per team and general

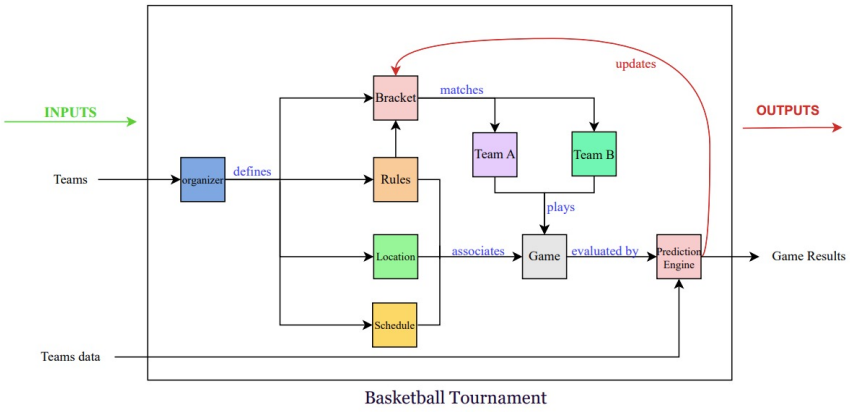

Figure 7.1: Basketball Tournament System Diagram

rules. They also determine the location and schedule of each individual match. This information is relevant to the system in the sense that these factors may affect the results of their respective match. Once all of these elements are determined the teams correspondent teams face each other in a match and said match produces a result where a team wins and the other loses.

The system diagram allows us to visualize the interconnectivity between all elements, showing how they are meaningfully and logically connected. It can also be stated that the system adapts to internal changes, as it constantly updates the brackets according to the results obtained. Finally, it is concluded that the system is open and may be non-linear given the context, as delays in the game's schedule caused by factors external to the tournament or personal struggles in a player's life can affect the matches' result despite being part of the system's environment. However, for the purposes of the competition, not all random factors that cause chaos are considered, and therefore the system is constrained to a linear model.

7.1.2 Architecture Design

The architectural design of the system was carried out following a modular, sequential, and scalable approach, with the aim of facilitating the implementation, traceability, and maintenance of each functional component. The architecture consists of a series of independent modules connected through a defined data flow, allowing for the processing of historical performance data and the generation of automated predictions for the results of a 64-team single-elimination tournament.

The proposed architecture (Figure 7.2) includes the following six main modules:

- **Data Ingestion Module:** Responsible for loading the historical `.csv` files into the system using the `pandas` library in Python. This module reads and structures the data for further processing.
- **Preprocessing Module:** Cleans the data by removing incomplete or irrelevant records (e.g., data from before 2015), normalizes numerical variables using `StandardScaler` from `scikit-learn`, and organizes data by division (men's/women's).
- **Feature Engineering Module:** Selects and constructs the most relevant predictive features, such as seed difference, win rate, and offensive/defensive efficiency. These features serve as input for the prediction model.

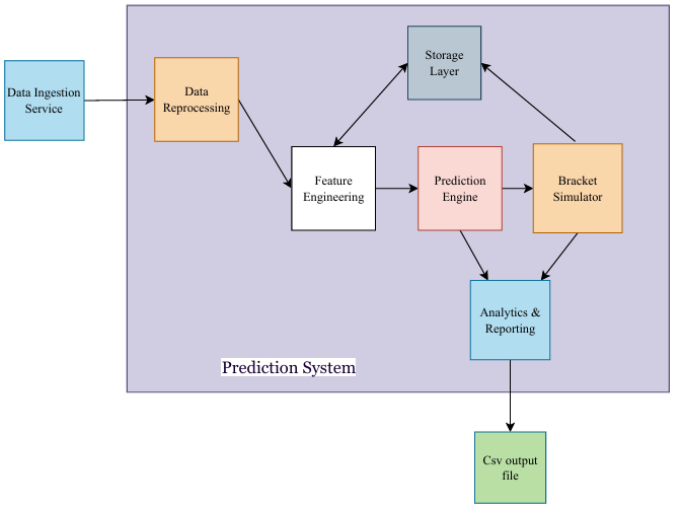

Figure 7.2: Prediction Architecture Diagram

- **Prediction Engine:** Implements logistic regression to estimate the probability of a team winning a given matchup, based on the processed features.
- **Bracket Simulator:** Organizes and simulates the tournament structure using the predicted outcomes. It was developed following the *Abstract Factory* design pattern, enabling dynamic creation of brackets with varying team numbers and tournament formats.
- **Output Module:** Collects the final results and exports them in `.csv` format for analysis and documentation purposes.

The interaction between modules follows a linear and traceable data pipeline. Each component operates independently but within an integrated system, allowing for future improvements or replacements without compromising the overall structure. The architecture is visually represented in a high-level diagram, which outlines the relationships and data flow between modules. This design ensures the system can be replicated, extended to other sports contexts, and remain aligned with sound systems engineering principles.

On the other hand, Figure 7.3 shows a representative workflow of the architecture. The diagram describes outlines the complete pipeline designed to transform raw historical data into actionable match predictions. This diagram complements the architectural breakdown by detailing the sequence of operations and data transformations that take place in the system. It begins with the ingestion of tournament datasets in .csv format, which are filtered and scoped according to division, year, and data completeness. The filtered data undergoes preprocessing to handle missing values, normalize features, and construct meaningful inputs for modeling. In the feature engineering stage, domain-specific metrics—such as win/loss ratios, seed differences, and scoring margins—are derived and curated. These features are then used to train a logistic regression model, which feeds into the prediction engine. From this point, a simulation module iteratively applies the predictions to all possible tournament matchups, generating outcomes round by round. Finally, the results are evaluated and exported in .csv format for further analysis or submission. The diagram also highlights the modular nature of the system and the external factors that may influence outcomes but are excluded from the model, such as referee bias or player injuries. This structured flow ensures that each transformation step is transparent, traceable, and aligned with the system's predictive goals.

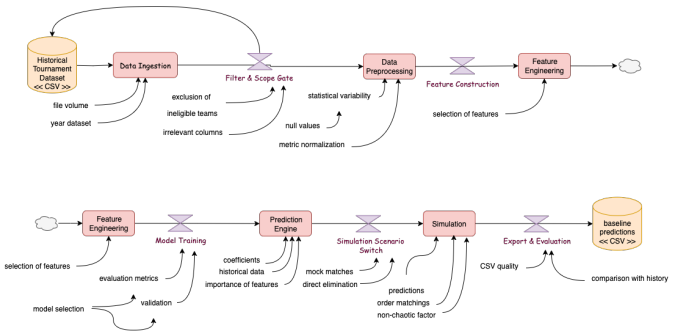

Figure 7.3: High-Level Data Processing and Prediction Workflow

7.2 Management & Data processing

First, the Python Pandas library was used as the main tool for manipulating .csv files during the data loading phase. Using the read_csv function, multiple data sources were incorporated into the work environment, facilitating their structured analysis.
Subsequently, in the preprocessing module, functions such as Pandas concat were used to join the different data sets into a single DataFrame, allowing an integrated analysis of the relevant variables. In addition, normalization techniques were incorporated using the scikit-learn preprocessing module, with the aim of reducing noise in the data and improving the quality of the predictions.

Feature	**Description**
Season	Year of the tournament
Gender	Gender category (M or W)
WTeamID	ID of the winning team in the matchup
LTeamID	ID of the losing team in the matchup
SeedA	Seed of Team A (winner in match record)
SeedB	Seed of Team B (loser in match record)
SeedDiff	Difference in seed values (SeedB - SeedA)
W_WinLoss	Win/Loss ratio of the winning team
L_WinLoss	Win/Loss ratio of the losing team
W_PPG	Average points scored per game by WTeam
L_PPG	Average points scored per game by LTeam
W_PtsAllowed	Avg. points allowed per game by WTeam
L_PtsAllowed	Avg. points allowed per game by LTeam
W_Margin	Scoring margin (avg. points scored - allowed) by WTeam
L_Margin	Scoring margin (avg. points scored - allowed) by LTeam
Diff_WinLoss	Difference in Win/Loss ratios (W - L)
Diff_PPG	Difference in points scored (W - L)
Diff_PtsAllowed	Difference in points allowed (W - L)
Diff_Margin	Difference in scoring margin (W - L)

Table 7.1: Summary of engineered features used for prediction

Table 7.1 shows all of the features taken into account at the moment of generating the tournament's predictions. Some important clarifications about these features are:

The WTeamID and LTeamID identify the winner and losing team of a singular matchup. The concept of seeds is determined by the NCAA organizers and are supposed to represent the overall performance of a team across the season, it takes into account elements such

win-loss ratio and difference in scores across games. In general the better the performance of the team the higher its seed number will be, thee numbers range from one to 16 and are assigned within each of the four regional brackets of the US (East, West, South, Midwest), following this logic the higher the difference between seeds the bigger the skill gap between teams and as such it serves as a very useful tool at the moment of generating predictions

Win/loss ratio is also a very useful feature as it allows to compare the overall performance of the team in recent matches against the rest of the teams in general without discriminating for a perceived level of skill as the seed difference does. The inclusion of this feature could help reduce the bias generated by by the usage of the seed feature.

Average points scored and allowed in a game are also relevant features, as they allow to visualize the offensive and defensive capabilities of each team. By combining both measures in the w_margin and L_margin allow for a direct comparison between their general capabilities as a team and internal synergies between attacking and defending scenarios. Other features used such as Diff_Winloss, Diff_PPG, Diff_PtsAllowed and Diff_margin have the function of comparing the previously mentioned features directly between each team.

7.3 Prediction Model

To implement the prediction model, we follow a workflow represented in Figure 3, which begins with a preliminary data preparation activity. First, we start by selecting the data sets that contain relevant information for generating predictions, which are provided by Kaggle (2025).

- **Data Ingestion** Due to the large amount of data provided by the competition it's important to discard outdated information and data that's difficult to incorporate in the model (such as the information related to coaches), for this it was decided to ignore all information previous to the year 2015 and from the remaining data only the following datasets where taken into account, in contrast to the 36 files provided by the competition.
  The data sets considered are as follows:
  - **MNCAATourneySeeds.csv and WNCAATourneySeeds.csv:** These files identify the pre-seeded teams in each NCAA tournament. The Seed field includes a region prefix (W, X, Y, Z) + number (01–16) + optionally a letter (a/b) for play-in games. The seed is the number assigned to a team within a tournament region, ranging from 1 (strongest) to 16 (weakest). It is useful for calculating seed difference, one of the most relevant features in prediction.
  - **MNCAATourneyCompactResults.csv and WNCAATourneyCompactResults.csv:** These files identify the results of the NCAA tournament game by game for all seasons of historical data. It contains NCAA tournament results exclusively and is useful for training classification models with actual "who won" results.
  - **MRegularSeasonDetailedResults.csv and WRegularSeasonDetailedResults.csv:** These files provide team statistics for many regular seasons with historical data. Statistics by team and by game: rebounds, steals, turnovers, free throws, etc. This is the key file for building performance features.

  - **MTeams.csv and WTeams.csv:** These files identify the different university teams present in the dataset. Columns such as FirstD1Season and LastD1Season help filter active teams.
  - **MSeasons.csv and WSeasons.csv:** These files identify the different seasons included in the historical data. With DayZero, you can convert DayNum into an actual date, facilitating temporal analysis. With this, you can standardize all seasons so that games coincide on the same scale of days.
  - **SampleSubmissionStage1.csv:** This file illustrates the submission format for the Stage 1 "preparation" competition. The ID field is in the format SSSS_XXXX_YYYY, where: SSSS = season XXXX, YYYY = team IDs.

- **Data Reprocesing** Now that we have the data we need to train the model and that is relevant for making predictions, using logistic regression, we continue to prepare these uploaded data sets.
  - First, we filter the datasets to include only seasons from 2015 onwards. This is to ensure consistency and relevance. This filter is applied to regular season games, tournament.
  - Second, Each team in the NCAA tournament is assigned a "seed" (e.g., 'W05', 'X12b') representing its rank within a region. To use this as a numeric feature, we extract only the numeric part of the seed (e.g., 'W05' $\rightarrow$ 5).

- **Feature Engineering** In this phase, variables (features) are created that capture the historical performance of teams in quantitative terms, so that the logistic regression model can use them to predict the outcome of matches. In this phase, we perform the following operations to obtain features:
  - **Compute seed difference** for each tournament game, this involves combining the ranking values of both teams in each game based on historical results from the NCAA tournament. The difference in ranking (RankingB - RankingA) will be one of the key features used for the prediction.
  - **Team-level statistics** are now generated based on each team's performance during the regular season. These statistics are calculated separately for each team, season, and gender, and will subsequently be merged into the tournament matchup dataset. The statistics include: Win-loss ratio (WinLossRatio), average points scored and conceded per game, total points scored and conceded, and point margin (average point difference).
  - **Calculating the statistics for each team in the regular season,** we now enrich the tournament matchup dataset by merging those statistics for both Team A (winner) and Team B (loser) in each historical match. In addition, we calculate derived features that capture the difference in performance between the two teams, such as: Difference in win-loss ratio, difference in point margin, and difference in average points scored. These comparative features are particularly useful for training classification models that predict the outcome of matchups based on historical indicators of strength.
  - **Assign a binary target variable "Winner"** y for balance each game is represented twice: Once with the winning team as TeamA (Winner = 1) y Once with the losing team as TeamA (Winner = 0)

	Season	Gender	TeamA	TeamB	SeedDiff	WinLossDiff	MarginDiff	PPGDiff	PtsAllowedDiff	Winner
0	2015	M	1214	1264	0	-0.108902	-3.223485	-2.723485	0.500000	1
1	2015	M	1279	1140	0	-0.093750	-3.781250	-9.218750	-5.437500	1
2	2015	M	1173	1129	0	0.015640	-1.632454	-1.658847	-0.026393	1
3	2015	M	1352	1316	0	-0.069404	-4.146628	-4.935484	-0.788856	1
4	2015	M	1112	1411	13	0.264706	17.088235	8.264706	-8.823529	1

Figure 7.4: High-Level Data Processing and Prediction Workflow

prepare a training dataset in which matchups between teams are structured neutrally and contain the features mentioned above. To train a predictive model, we restructured each historical match into a matchup between Team A and Team B, and assigned a binary target variable Winner. See Figure 7.4

- **Prediction Engine** Logistic regression (LogisticRegression) from the scikit-learn library is used to train a binary classification model. The objective is to predict whether the team wins or loses based on variables such as: SeedDiff, Diff_WinLoss, Diff_PPG, Diff_PtsAllowed, and Diff_Margin. These are generated in the Feature Engineering phase.

7.4 Scenarios

To evaluate the robustness and behavior of the prediction system, two distinct simulation scenarios were implemented in the notebook. The first, referred to as the Baseline Scenario, simulates all possible matchups between NCAA 2025 qualified teams using the trained logistic regression model. For each matchup, engineered features such as SeedDiff, WinLossDiff, MarginDiff, PPGDiff, and PtsAllowedDiff are computed and used as inputs to predict the probability of TeamA winning. Each matchup is mirrored (TeamA vs TeamB and TeamB vs TeamA) to ensure symmetric evaluation, and predictions are generated for all pairwise combinations. The output of this scenario is a full set of win probabilities under clean and stable input conditions, which is saved in a .csv file as a reference prediction set.

The second experiment, the Perturbed Scenario, explores the system's sensitivity to input noise. In this setup, random perturbations between $\pm 2\%$ and $\pm 5\%$ are applied to the SeedDiff feature while keeping the other inputs constant. This mimics real-world uncertainty and data variability. The model is then used to re-predict the outcomes using these slightly altered inputs. The analysis compares baseline and perturbed predictions, focusing on two metrics: (1) the average absolute difference in predicted probabilities, and (2) the proportion of matchups whose prediction changes significantly (by more than ± 0.10). A scatter plot is also generated to visually assess stability, where deviation from the diagonal line ($y = x$) indicates higher sensitivity. These experiments provide empirical insight into the resilience and reliability of the prediction engine when confronted with noisy or imperfect input data.

Once the model is completed trained and executed some metrics were implemented to evaluate its accuracy when making predictions, these include an accuracy score, a confusion matrix and a ROC curve.

The accuracy score is a simple metric that measures how many predictions were correct in comparison to the total amount of predictions made giving a score between 0 and 1.

The confusion matrix is a more complex measure since it represents a matrix that compares the amount of correct and wrong predictions and the respective context in which each situation happens. In this particular project the confusion matrix would identify out of all the

times that the model predicted that a team would win, how many times the correct answer was that a team would win and how many times the correct answer was that the team would lose and the same logic would apply for when the model predicts that a team would lose, this way it's possible to classify the system's predictions as true positives (the prediction was that the team won and the correct answer too), false positives (The prediction was that the team would win but the correct answer was that the team lost), false negatives (the prediction was that the team would lose but the team actually won) and true negatives (the prediction was that the team would lose and they actually lost). By using this metric it's not only possible to track the accuracy of the prediction but also allows to identify which scenarios are causing troubles to the prediction system, for example if the model is causing bias towards a particular team winning the confusion matrix would show a large amount of false positives, and by identifying these defects the model can be updated to improve its accuracy.

A ROC curve, or Receiver Operating Characteristic curve, is a graphical tool used to evaluate the performance of a binary classification model. It shows how the true positive rate varies with the false positive rate across different decision thresholds. For this graphic a perfect score shows a curve that tends to the upper left corner, showing a perfect classification of of true positives and true negatives, the curve always starts an the (0;0) point and ends in the (1; 1) point. By calculating the area under this curve we obtain the auc, which summarizes in a single number the information proportioned by the ROC curve, on this metric a number of 0.5 means that the system predicts randomly and 1 is a perfect classification score.

7.5 Large Language Model (LLM) Integration

To explore the integration of large language models (LLMs) into the NCAA tournament prediction pipeline, we initially aimed to compare different state-of-the-art models (e.g., OpenAI's GPT, Google Gemini, Claude) to assess their ability to generate insights or support decisions based on structured sports data. However, due to API limitations and restricted execution environments, only Google's Gemini 1.5 Flash model was successfully integrated using the Kaggle notebook environment. Through prompt-based interaction, Gemini was tasked with analyzing simulated match data (including features such as seed differences, win-loss ratios, and scoring margins) to provide explanations, justifications, and feature-based reasoning behind match outcomes. This qualitative analysis supported the model's predictions and offered a human-like narrative component that complements quantitative approaches.

Although the initial goal was to integrate a large language model (LLM) directly into the predictive pipeline to generate outcome probabilities or simulate tournament results, this approach was not feasible. Due to the nature of LLMs and platform limitations, Google's Gemini 1.5 Flash could not be used to directly perform numerical predictions from tabular data. Instead, the LLM was incorporated within the Kaggle environment to analyze individual match features and provide qualitative insights. While it did not replace the machine learning model, Gemini proved useful in highlighting feature importance, validating results, and suggesting potential new variables. This human-like reasoning adds interpretability to the model and may inform future iterations of the predictive system.

```
1 row = df.iloc[1]
2
3 prompt2 = f"""
```

```
You're analyzing a historical NCAA match simulation from Season {row['
    Season']} ({row['Gender']} division).

Teams:
- Winner: Team {row['WTeamID']} (Seed {row['SeedA']})
- Loser: Team {row['LTeamID']} (Seed {row['SeedB']})

Feature differences:
- Win/Loss ratio diff: {row['Diff_WinLoss']}
- Margin diff: {row['Diff_Margin']}
- PPG diff: {row['Diff_PPG']}
- Points allowed diff: {row['Diff_PtsAllowed']}

Tasks:
1. Does the winner make sense given this data?
2. Which feature seems most predictive in this case?
3. Suggest one additional stat that might improve the model's ability to
    predict match outcomes.
"""

response2 = flash.generate_content(prompt2)
print(response2.text)
```

Listing 7.1: Prompt sent to Gemini for qualitative match analysis

Chapter 8

Results

By generating the confusion matrix for the logistic regression model that implements all of the proposed features the general accuracy of the predictions appears to be acceptable, as most f the cases initially tested seem to fall in the true positive and true negative labels with 174 and 181 matches respectively out of 473 test cases, despite this the amount of false positives and false negatives is considerable as there were 55 and 63 of these cases respectively.

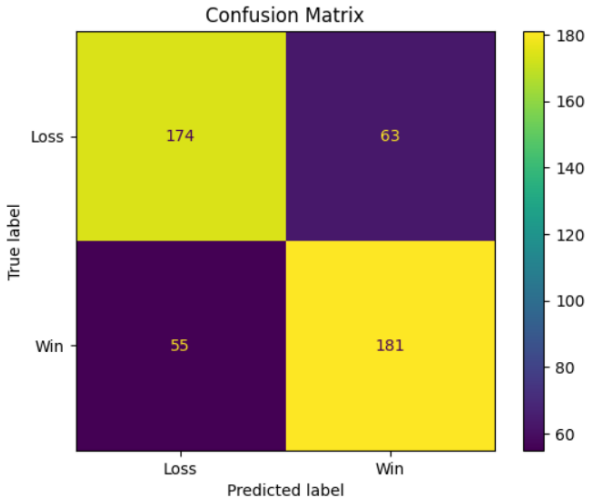

Figure 8.1: Confusion matrix for the complete model

The accuracy metric for this version of the model leads to a similar result with an accuracy of 0,7505 which means that in general the accuracy is good, but there's still room for improvement. The AUC score of 0.8157 offers a similar insight, the prediction is not perfect, but the accuracy is decent

On the other hand the ROC curve also proves a similar insight, the prediction is not perfect, the curve tends to the side of being more accurate but it's still not quite on the optimal form of the curve

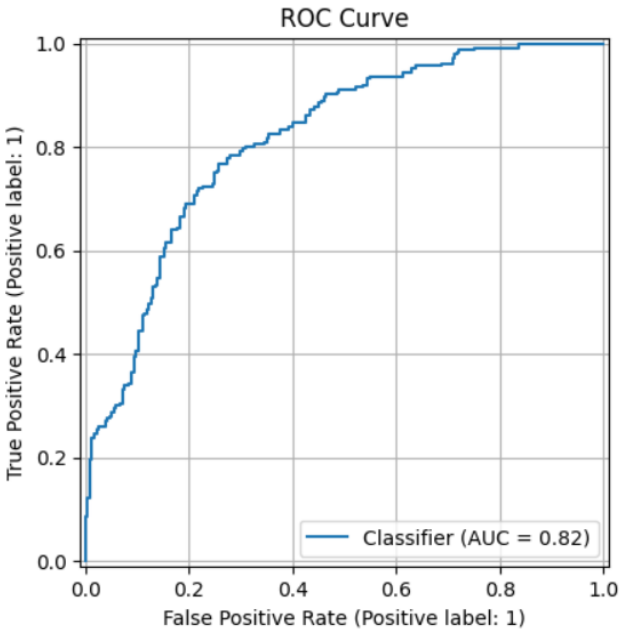

Figure 8.2: ROC curve for the complete model

8.0.1 Prediction Shift Analysis

To evaluate the sensitivity of the model to input perturbations, a scenario-based analysis was conducted where small amounts of noise were injected into the `SeedDiff` feature. This process simulates real-world uncertainty in data, where input values may not always be perfectly measured or recorded. Specifically, for each 2025 tournament matchup, a random noise ranging between $\pm 2\%$ and $\pm 5\%$ was applied to the original `SeedDiff` values, while keeping the remaining features unchanged.

Two quantitative metrics were used to assess the model's robustness:

- **Average Change**: The mean absolute difference between baseline and perturbed predicted probabilities. A low value suggests high stability.
- **Shift Proportion**: The proportion of matchups in which the predicted probability shifted by more than ± 0.10. This indicates how frequently noise leads to significantly different predictions.

Figure 8.3 provides a visual summary of this analysis through a scatter plot that compares baseline predictions (x-axis) against perturbed predictions (y-axis). Each dot represents a tournament matchup. The red dashed diagonal line ($y = x$) denotes perfect consistency—matchups for which predictions remained unchanged despite the perturbation. The degree of dispersion around this line reflects the model's sensitivity to noise.

The results reveal that the majority of predictions fall very close to the diagonal line, indicating that the model is generally resilient to minor variations in input. The average change observed was minimal, and only a small percentage of matchups experienced a shift greater than ± 0.10 in predicted probability. This suggests that the logistic regression model, trained with carefully engineered features, maintains consistent output under small-scale feature noise, supporting its reliability for decision-making in realistic, imperfect data conditions.

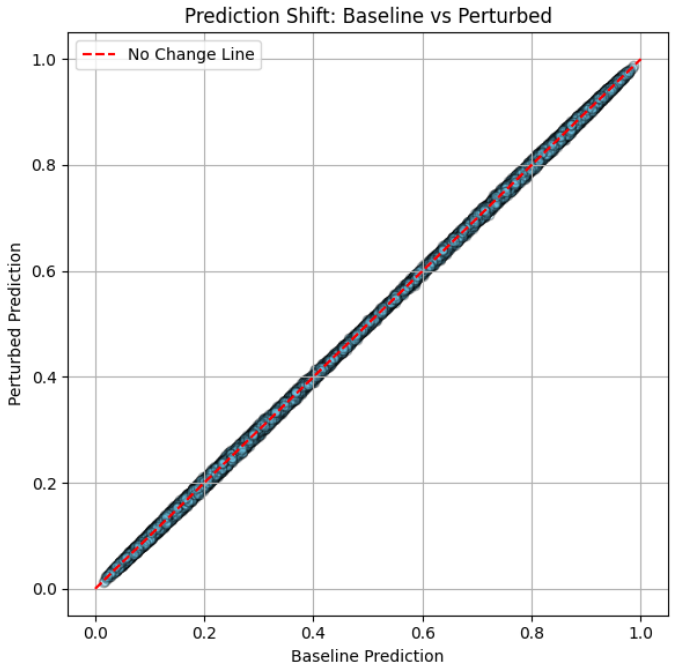

Figure 8.3: Prediction Shift Analysis

8.0.2 Multi-Feature Perturbation Scenario

To further investigate the resilience of the predictive system, a multi-feature perturbation experiment was conducted. This simulation introduces controlled noise into several engineered input variables simultaneously, in order to evaluate the model's behavior under more realistic conditions where multiple metrics may be subject to uncertainty or measurement error.

The perturbed features include:

- **SeedDiff**: Difference in seed rankings between teams
- **WinLossDiff**: Difference in win/loss ratio
- **MarginDiff**: Difference in average scoring margin
- **PPGDiff**: Difference in average points per game
- **PtsAllowedDiff**: Difference in average points allowed

For each feature, random noise ranging from $\pm 2\%$ to $\pm 5\%$ of the absolute value of the original feature was added. This models realistic imperfections in data such as inconsistencies in team records, inaccurate inputs, or rounding errors in statistics. The objective is to understand how small simultaneous shifts in multiple inputs may impact prediction outcomes.

The trained logistic regression model was then reapplied to the perturbed dataset, and the resulting predictions were compared against the baseline probabilities. Figure 8.4 presents a scatter plot comparing baseline predictions (x-axis) to perturbed predictions (y-axis). Each point represents a tournament matchup, and the red dashed diagonal line ($y = x$) indicates perfect prediction consistency.

As observed, the majority of points lie very close to the diagonal line, indicating high consistency between the perturbed and baseline predictions. Although the perturbation affects multiple features at once, the resulting variation in predicted probabilities remains minimal.

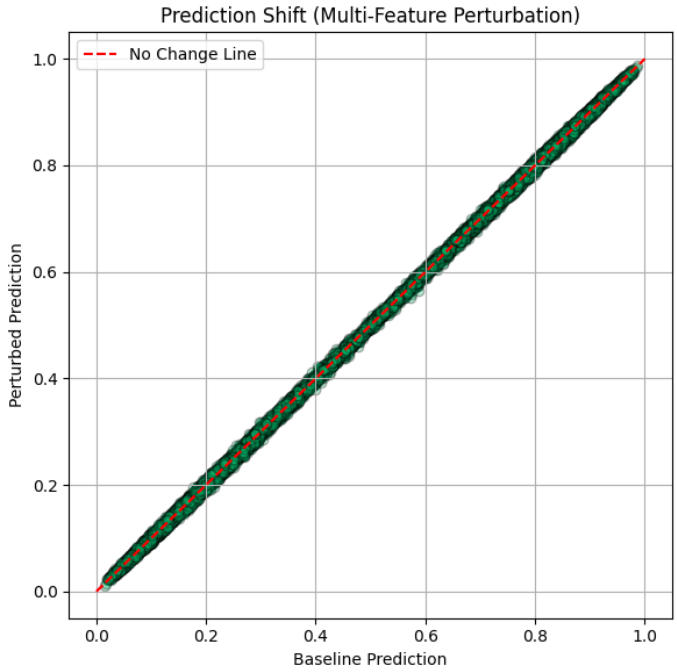

Figure 8.4: Perturbed Scenario (Multi-Feature)

This implies that the logistic regression model is relatively robust to simultaneous input fluctuations, suggesting that the engineered features and their influence on the model are well balanced.

This analysis provides additional evidence that the system can maintain predictive reliability under noisy input conditions, supporting its applicability in real-world scenarios where data imperfections are common.

Results from LLM Integration

Although Gemini 1.5 Flash was not used to make numerical predictions, it was successfully integrated into the Kaggle environment to generate qualitative analyses. Using prompt-based interaction, the model produced textual explanations for a subset of simulated NCAA tournament matches. These explanations were based on features such as seed differences, win-loss ratios, scoring margins, and points allowed.

The generated outputs provided narrative justifications for why a particular team was more likely to win. In many cases, Gemini's reasoning aligned with the outcomes predicted by the machine learning model, confirming the relevance of key features. These outputs were stored alongside the original dataset to support interpretability and post hoc analysis. An example of a generated response is shown in Figure 8.5.

1. **Does the winner make sense given this data?**

No, the winner doesn't entirely make sense given the data. All four feature differences are negative, indicating that Team 1140 (the loser) performed significantly better across the board than Team 1279 (the winner). A negative win/loss ratio difference means Team 1140 had a better win-loss record, a negative margin difference means they won by a larger average margin, and negative PPG and points allowed differences show they scored more points per game and allowed fewer. These suggest Team 1140 was the stronger team. The fact that Team 1279 won despite these disadvantages suggests that other unmeasured factors played a significant role in determining the outcome of this specific match. Random chance, a particularly strong performance by Team 1279 in that game, or even a significant injury impacting Team 1140 could explain the unexpected result.

2. **Which feature seems most predictive in this case?**

The **PPG diff (-9.21875)** seems the most predictive in this *specific* instance. A difference of nearly 10 points per game is substantial. While all differences point to Team 1140's superiority, the magnitude of the PPG difference is largest, indicating a significant scoring disparity between the two teams that should, generally speaking, translate to wins for the team with higher PPG.

3. **Suggest one additional stat that might improve the model's ability to predict match outcomes.**

Turnover differential would be a valuable addition. Turnovers (both forced and committed) significantly impact game outcomes. A team that forces many turnovers and commits few is likely to win more games, irrespective of other statistics. Including turnover differential would help capture the impact of these crucial plays on the overall result and provide a more nuanced understanding of team performance.

Figure 8.5: Example of a textual explanation generated by Gemini for a simulated match.

Chapter 9

Discussion

9.1 Discussion of Results

The evaluation metrics obtained for the logistic regression model suggest an overall acceptable level of predictive performance. With an accuracy of 0.7505 and an AUC score of 0.8157, the system demonstrates the ability to identify relevant patterns in historical NCAA tournament data and convert them into reasonably reliable predictions. The confusion matrix, shown in Figure 8.1, further reveals that true positives and true negatives dominate the prediction outcomes, although the presence of 118 misclassified cases (55 false positives and 63 false negatives) indicates potential areas for improvement.

These results align with the expected behavior of statistical models applied to complex systems, such as sports tournaments, where inherent uncertainty and emergent interactions limit full predictability. While the model does not reach perfect precision, it performs consistently better than chance, which validates both the engineered features and the logistic regression framework adopted.

The robustness of the model was further validated through sensitivity analysis. The single-feature perturbation experiment, targeting only `SeedDiff`, revealed that predictions are largely stable even under slight modifications to input data. This resilience was confirmed by two key metrics: a low average shift in prediction probabilities and a small percentage of cases exceeding the ± 0.10 change threshold.

Expanding this analysis, a multi-feature perturbation scenario simulated more complex, real-world noise by altering five input features simultaneously. Despite this, the predictions remained closely aligned with baseline values (Figure **??**), indicating that the model's structure effectively absorbs minor fluctuations in inputs without significantly altering outcomes. This is a strong indicator of model reliability in environments where data imperfections are common.

Nonetheless, the evaluation process also highlighted an operational limitation: the system failed to produce a valid submission for the Kaggle competition due to errors in matchup ID formatting. While this does not compromise the internal performance of the model, it does underscore the importance of end-to-end compliance with external specifications, particularly in competitive or production settings.

Finally, the integration of a large language model (LLM) as a qualitative explanation tool adds an additional interpretability layer. The generated narratives helped contextualize model predictions and confirm the relevance of key features such as seed differences and scoring margins. This component, though not directly involved in prediction, strengthens the transparency and explainability of the system's outputs.

Overall, the findings demonstrate that the proposed system performs well under controlled and perturbed conditions, is structurally robust, and benefits from complementary tools that

improve its interpretability. However, the limitations encountered—in prediction formatting and misclassification rates—highlight the need for further refinements in both implementation and validation workflows.

Discussion on the Use of LLMs

While large language models (LLMs) like Gemini are not designed to replace traditional machine learning algorithms for numerical prediction tasks, their integration into the modeling pipeline offered several benefits. In this project, Gemini was not used to predict match outcomes directly, but rather to provide natural language explanations of simulated results. This interpretive layer added human-like reasoning that complemented the statistical rigor of the predictive model.

The qualitative insights generated by Gemini helped validate the relevance of certain features (such as point differential and seed gap) and highlighted cases where the model's predictions may not align with intuitive interpretations. This supports the use of LLMs as post hoc explanatory tools or as advisors in the feature engineering process.

However, several limitations were identified. The LLM's outputs are not quantifiable or directly verifiable, and they depend heavily on prompt design. Furthermore, due to computational and access constraints, only one LLM (Gemini 1.5 Flash) was used, limiting the possibility of comparative analysis.

Nonetheless, this preliminary integration suggests that LLMs can play a valuable support role in enhancing model interpretability, guiding improvements, and enriching the overall understanding of data-driven decisions in sports analytics.

Chapter 10

Conclusions and Future Work

10.1 Conclusion

This project developed a predictive system for the NCAA 2025 college basketball tournament using a logistic regression model supported by historical data and engineered features. The model achieved solid performance, with an accuracy of 75.05% and an AUC of 0.8157, indicating its ability to capture meaningful patterns in team performance metrics such as seed difference, win-loss ratio, and scoring margins.

Robustness testing through input perturbations confirmed the model's stability under noisy or incomplete data, validating its reliability for structured sports prediction. Additionally, the system's design was informed by systems thinking principles, treating the tournament as a dynamic, interconnected environment rather than a static dataset—an approach that enhanced both interpretability and modularity.

Although the model performed well, two key limitations emerged: first, the system could not produce a valid submission to the Kaggle competition due to formatting issues; and second, the integration of a large language model (Gemini) was limited to generating qualitative explanations rather than direct prediction. Despite this, the LLM proved valuable in supporting interpretability and offering contextual insights that complemented the quantitative results.

In conclusion, the project establishes a strong baseline for outcome prediction in complex sports environments. Future work should focus on improving data integration, ensuring compliance with competition standards, exploring more advanced predictive models, and enhancing the role of LLMs in both interpretability and simulation tasks.

10.2 Future Work

Several limitations encountered during this project have opened clear avenues for future work. First, the system did not generate a valid submission for the official Kaggle competition due to formatting issues and incomplete matchup data. Resolving these technical gaps—such as aligning with Kaggle's submission schema and enriching the dataset with full tournament brackets—would enable participation in future editions and facilitate direct benchmarking against other models.

Second, while a large language model (LLM) was successfully integrated to provide qualitative insights, its use was limited to textual interpretation. The original goal of leveraging an LLM for direct prediction, simulation, or model augmentation could not be achieved within the current constraints. Future work could explore more advanced LLM integration strategies, such as using structured prompts with embeddings, connecting to external knowledge bases

via Retrieval-Augmented Generation (RAG), or fine-tuning models for domain-specific tasks.

Moreover, the dataset could benefit from additional features—such as recent performance trends, injury reports, or player-level statistics—to improve predictive accuracy. Expanding the model beyond logistic regression, incorporating ensemble methods or neural networks, may also enhance performance and robustness under uncertainty.

Finally, a comparative analysis between multiple LLMs (e.g., Gemini, GPT-4, Claude) would help assess their relative effectiveness in sports analytics contexts. This would require stable API access and a modular framework that allows switching between models efficiently.

In summary, while this project established a foundational prediction system and demonstrated the complementary value of LLM-based interpretation, future work should focus on data completeness, advanced model integration, and broader system validation to fully realize its potential.

References

Kaggle (2025), 'March machine learning mania 2025', https://www.kaggle.com/competitions/march-machine-learning-mania-2025/data. Accessed: May 15, 2025.

NCAA (2025), 'What is march madness? the ncaa tournament explained'. Accessed: 2025-05-16.
URL: *https://www.ncaa.com/news/basketball-men/bracketiq/2025-01-22/what-march-madness-ncaa-tournament-explained*

University of Waterloo (n.d.), 'What are complex systems?'. Accessed: 2025-05-16.
URL: *https://uwaterloo.ca/complexity-innovation/about/what-are-complex-systems*