NCAA 2025 Tournament prediction

1st Carlos Andrés Celis Herrera
Ingeniería de Sistemas
Universidad Distrital
Bogotá, Colombia
20222020014

2nd Nicolas Romero
Ingeniería de Sistemas
Universidad Distrital
Bogotá, Colombia
20222020023

3rd Xiomara Salome Arias Arias
Ingeniería de Sistemas
Universidad Distrital
Bogotá, Colombia
20222020014

***Abstract*—This project presents a solution to the Kaggle competition on NCAA basketball tournament predictions, using a computational and systems-oriented approach to forecast outcomes in both the men's and women's divisions. The solution is built upon historical performance data, distributed across 37 structured CSV files. A logistic regression model serves as the core of the prediction engine, estimating match probabilities based on variables such as win-loss ratios and average team performance metrics. The system is developed with a modular architecture that supports scalability and maintainability, and it is implemented using Python libraries such as Pandas and Scikit-learn. The study also emphasizes the inherent sensitivity and complexity of the tournament environment, where small variations in input data or external factors can lead to significant shifts in outcomes. Overall, this work demonstrates the practical application of systems thinking and machine learning techniques in a dynamic, data-intensive sports context.**

***Index Terms*—Prediction, artificial intelligence, LLm, architecture, modules**
t

I. INTRODUCTION

Throughout its history, one of the primary goals of Artificial Intelligence (AI) has been the identification of patterns within large datasets—patterns that would otherwise take humans a significant amount of time to uncover without technological assistance. These patterns, extracted from historical data, can be used to generate predictions about the behavior of various systems by applying forecasting techniques that align with the available data.

Among the most challenging domains for prediction are *complex systems*. These systems are characterized by being composed of numerous interconnected elements whose interactions produce emergent behaviors. In addition to internal dynamics, they are influenced by external environmental factors. Complex systems often contain multiple subsystems, introducing additional layers of interdependence. A failure in one component may propagate through the system, affecting others—a phenomenon known as systemic interdependency. Another key characteristic of complex systems is their non-determinism: the same inputs may yield different outputs at different points in time. This randomness does not imply a complete lack of structure, but rather introduces a level of variability that makes behavior less predictable and more nuanced. Despite this, statistical patterns can still be identified and leveraged.

This document presents the development of a machine learning model designed to predict the behavior of such a complex system while accounting for the properties described above. The selected case study is the annual NCAA college basketball tournament—commonly known as *March Madness*—which serves as the foundation of the Kaggle prediction challenge. The goal is to design and implement an autonomous prediction system that uses historical tournament data, including both team-level and individual statistics for all eligible participants. This raw data is ingested, preprocessed, and transformed to meet the requirements of the predictive model. A logistic regression algorithm is then applied to generate outcome probabilities for all possible matchups in the tournament. Finally, the predictions are evaluated by comparing them with the real outcomes of the 2025 tournament games.

The project also aims to illustrate the relevance and applicability of systems analysis and systems engineering principles in the development of predictive systems that address real-world problems in a structured and scalable manner.

II. DEVELOPMVENT

Competition overview

The system proposed for this project aims to generate predictions of the results of this year's collegiate basketball tournaments of both women's and men's division using historical data of this league past's games as inputs. The system would generate every possible matchup between eligible teams for the tournament an then give the probabilities of each team winning in each matchup as output.

Basketball tournament as a system

Figure 1 shows a system diagram of the tournament as a whole, serving to identify each component relevant to the system's operation, as well as the interrelations and potential random factors that influence the system's output. The functioning of these components is as follows:

The organizer is responsible for defining the structure of the tournament, including how brackets are formed, the rules

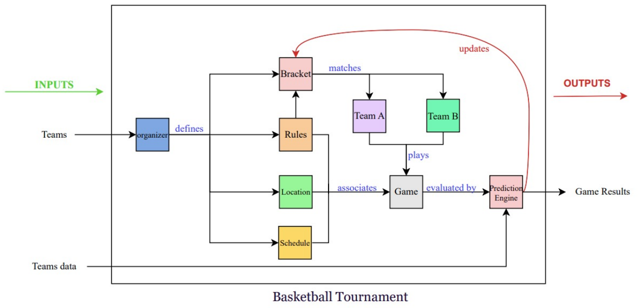

Fig. 1. System representation

that govern how the game is played, the locations where matches are held, and the schedule that determines when each game takes place. Brackets determine the order and progression of matchups, allowing the tournament to unfold in a structured manner. Rules establish the framework under which players compete and determine how teams are ranked and eliminated. Locations refer to the physical venues where teams play, and the schedule dictates the timing and sequence of the matches. The game component itself represents the individual matchups in which Team A and Team B face off as part of the tournament, with all its parameters being shaped by the aforementioned components.

This system is inherently non-deterministic and is influenced by various random factors that introduce variation in game outcomes. For example, the personal performance of a player can be unexpectedly affected by injuries or unforeseen personal circumstances. Similarly, incorrect referee decisions or bias can directly alter the outcome of a match. Unanticipated delays, such as weather events or logistical issues, can disrupt the game's rhythm and impact results without prior warning. Furthermore, even if a team has demonstrated consistently strong or weak performance in past tournaments, there remains the possibility of a sudden change in their level of play, introducing unpredictability despite the use of structured prediction models. Because of the difficulty of adding these random factors to the systems design it was decided to not take them into account.

Figure 2 represents a system diagram with the architecture proposed for designing the prediction system, the most important elements of this architecture function as follows:

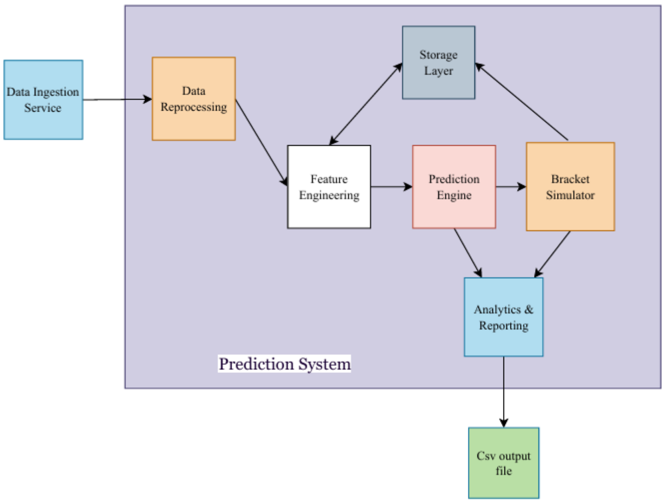

Fig. 2. Architecture diagram

The data ingestion service is responsible for collecting historical data to feed the prediction system. Although the dataset provided by the competition includes information dating back to 1984, it was decided to limit the scope to data from 2015 onward. From this filtered subset, only team-wide statistics and game locations were selected as relevant features. This data is then passed to the data preprocessing module, where it undergoes several cleaning operations. During this stage, the selected data is normalized to ensure uniform scale, formatting inconsistencies are corrected, missing or incomplete values are handled appropriately, and irrelevant or redundant records are removed to improve data quality.

More specifically the model only takes into account data from 2015 onward, and the features selected for the process were the following:

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 I
TABLE 1: SUMMARY OF ENGINEERED FEATURES USED FOR PREDICTION

Following preprocessing, the cleaned data is passed to the feature engineering module. This module transforms the raw information into meaningful variables that can effectively support the prediction process. Once the data has been engineered

into its final form, it is stored in a dedicated storage layer, from which the prediction engine retrieves it to compute the outcome probabilities for every possible matchup in the tournament. The resulting predictions are compiled into a CSV file, which serves as the system's primary output.
On the other hand, Figure 3 prediction workflow of the system. The diagram outlines the sequential architecture followed from the ingestion of historical data to the generation of match outcome predictions. Each module is shown alongside the key factors that influence its behavior, such as data volume, normalization, statistical variability, and the presence of null values. The architecture begins with data ingestion from structured CSV files and proceeds through a filtering and preprocessing pipeline, where irrelevant or incomplete records are excluded and metrics are normalized. Feature construction and engineering modules transform raw inputs into informative variables, which are then used by the prediction engine trained using logistic regression. This helps us to better understand the internal dynamics and how uncertainty propagates through the system.

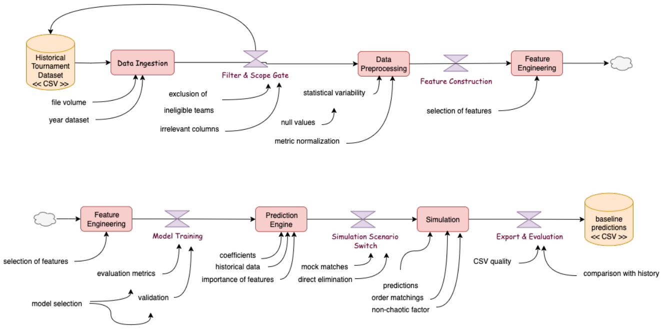

Fig. 3. High-Level Data Processing and Prediction Workflow

To support future improvements and system tuning, the architecture also proposes the integration of a monitoring component capable of generating visual reports. These reports would track prediction accuracy and system performance over time, enabling continuous evaluation and optimization of the model.

Figure 4 showcases a causal loop diagram that justifies the development of a module that tracks and takes metrics of the system performance. This diagram shows that feedback received trough this means can help to improve refine the system, design, which in turn can help improve the data quality to make more relevant and effective features that would eventually lead to generate better predictions, leading to generate more feedback and completing the proposed loop.

For the purpose of making this module the following metrics were implemented: accuracy measure, confusion matrix, ROC diagram and AUC. All of these are metrics that evaluate the accuracy of the predictions made by the system.

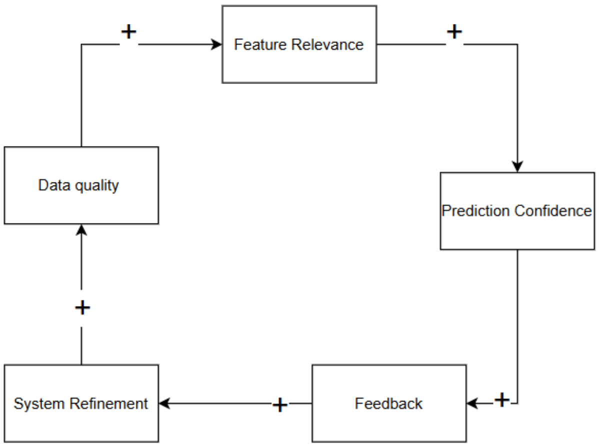

Fig. 4. Causal loop diagram applied to feedback

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 allows to identify the exact amount of each possible result generated by the system in each context, these results can be classified as true positives, false positives, false negatives and true negatives. A true positive occurs when the system predicts that a team should win in a specific matchup, and the actual result of the match was that the team indeed won, a false positive occurs when the system predicts that a team should win, but in the actual match said team lost. This same logic applies for the remaining cases.

The ROC diagram puts these same results in therms of a graphic, offering a different optic on the same metric, the AUC summarizes the information of the ROC diagram in a single numerical value, in order to interpret this metric it's important to understand that a score of 1 is a prefect classification and a score of 0.5 means that the system is guessing randomly

III. Results

By applying the ROC diagram, the confusion matrix and the AUC on the model that applies all of the selected features the results are mostly similar, all of them show that the predictions are not perfect, but in general the predictions tend to be accurate more often than not. Specifically the AUC score has a value of 0.8157 and the ROC diagram and confusion matrix produce the following outputs.

The confusion matrix displays the results of 437 test cases where 355 of them where sorted correctly.

In addition to these metrics, testing on various possible scenarios for the simulation was implemented, the following

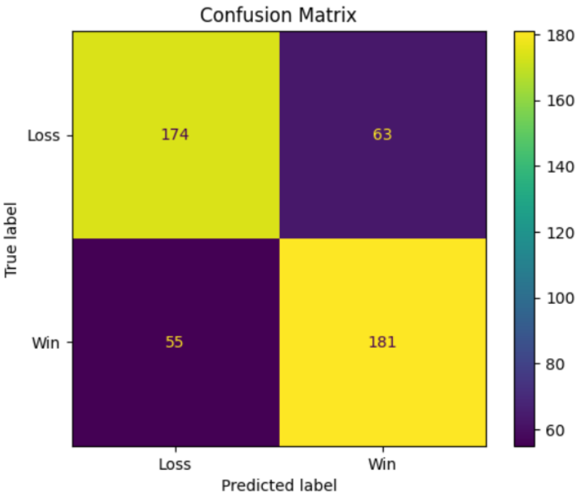

Fig. 5. confusion matrix

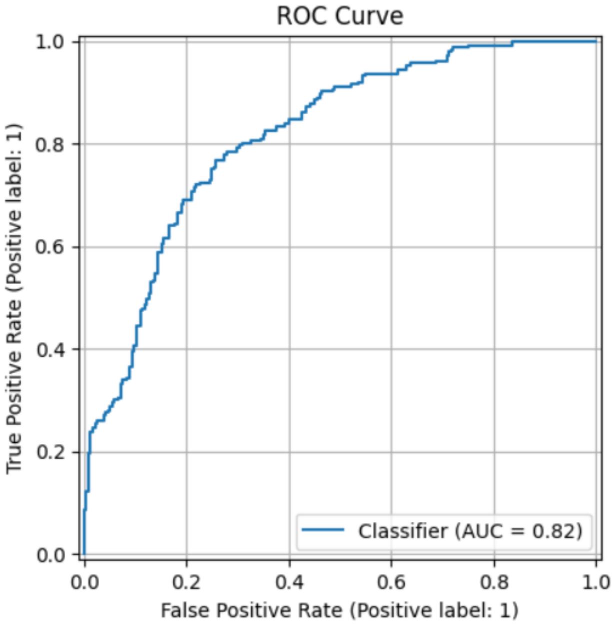

Fig. 6. ROC diagram applied on the complete model

shows the variation between the baseline case proposed and a perturbed scenario where there was added noise to the dataset, despite this, there was no visible change on the predictions.

IV. CONCLUSIONS

Despite the successful development and implementation of the prediction system, the final submission to the Kaggle competition could not be scored. This result was due to a lack of alignment between the predictions generated and the format required by the competition, specifically in the generation and identification of possible pairings. The system failed to reproduce the exact number of predictions required by the competition. As a result, the competition platform was unable to process the predictions and return a valid evaluation score.

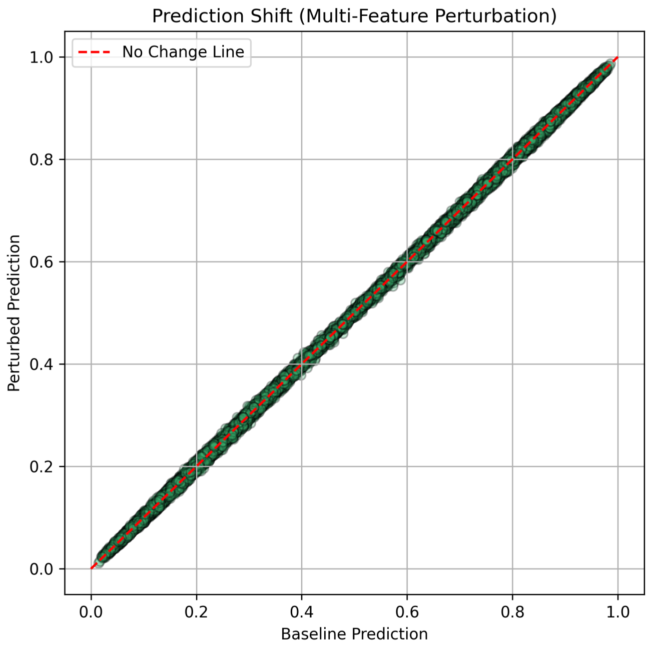

Fig. 7. Perturbed prediction and baseline case comparation

The problem was not due to a failure in the central prediction logic, the feature engineering process, or the performance of the logistic regression model itself. Instead, it originated in the simulation and export stages, where the expected team pairing logic was not replicated with complete accuracy.

Despite this, metrics produced within the system indicate that the predictions that were generated were accurate in general, with some room for improving at the moment of selecting features and algorithms for the predictions.

REFERENCES

[1] Kaggle (s.f) https://www.kaggle.com/competitions/march-machine-learning-mania-2025/code