UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Universidad Francisco José de Caldas

System Engineering

Report of Predict Student Performance from Game Play

Henry Ricaurte Mora
Javier Alejandro Penagos Hernández

Germán Darío Aya Fuentes

Supervisor: Carlos Andrés Sierra Virguez

Report submitted as part of the requirements for the course
Systems Analysis and Design
from the Systems Engineering program

July 12, 2025

Abstract

This project addresses the problem of predicting student performance in game-based learning environments using behavioral data. With the growing use of educational video games, the ability to anticipate whether a student will correctly answer a question based on in-game interactions becomes essential for adapting content and improving learning outcomes. The system is designed as a supervised machine learning pipeline that classifies student responses using click patterns, mouse movements, elapsed time, and gameplay progression. To handle the structured nature of the data and the complexity of user behavior, tree-based models such as XGBoost and TensorFlow Decision Forests were employed. These models offer a strong balance between accuracy, interpretability, and efficiency, especially in settings with non-linear relationships and limited data. Feature engineering was applied to extract meaningful temporal, spatial, and contextual variables. The system architecture supports real-time data ingestion, preprocessing, and prediction while maintaining scalability and modularity. The model's performance was evaluated using standard metrics including accuracy, precision, recall, and F1-score, yielding high predictive accuracy in the competition setting. The results demonstrate that behavioral features in educational games can be leveraged effectively to predict performance, offering valuable insights for personalized learning strategies. This approach contributes to the broader field of learning analytics and intelligent tutoring systems by providing a scalable framework for adaptive educational technologies.

Keywords: game-based learning, student performance prediction, behavioral analytics, machine learning, decision forests

Contents

List of Figures

List of Abbreviations

SMPCS	School of Mathematical, Physical and Computational Sciences

Chapter 1

Introduction

Learning should also be fun. This project addresses the lack of adaptive content in educational games due to ineffective knowledge tracking methods. We propose a data-driven approach using gameplay interaction patterns (clicks, mouse movements) to predict student performance in real-time. Our modular pipeline includes data ingestion, preprocessing, tree-based model training, and prediction, aiming to optimize learning experiences across age groups.

1.1 Context

The Field Day Lab is a publicly funded research lab at the Wisconsin Educational Research Center. It designs educational games for diverse knowledge areas and age groups, leveraging data to understand learning optimization. Key commitments include:

- Free, accessible games for all audiences.
- Data collection to continuously improve game design.

1.2 Problem Statement

Most educational games fail to adapt content to individual students due to ineffective knowledge tracking. This project tackles this gap by capturing and analyzing relevant gameplay data.

1.3 Objectives

- **Goal**: Develop a machine learning system to predict student performance in educational games using behavioral data.
- **Specific Objectives**:
    1. Define functional and technical requirements from the interaction data.
    2. Design a scalable, modular architecture with real-time data processing.
    3. Implement and evaluate tree-based models (e.g., XGBoost, TF-DF).
    4. Extract and engineer temporal, spatial, and contextual features.

1.4 Solution Approach

A modular 4-step framework:

1.4.1 Data Ingestion

- Source: Kaggle datasets

1.4.2 Data Preparation

- Cleaning: Remove erroneous/duplicate clicks.
- Standardization: Normalize (x, y) coordinates.
- Feature engineering: Extract temporal features and derive the target variable (implicit in raw data).

1.4.3 Model Training

- Tree-based models: Random Forest, CART, Gradient Boosted Trees.
- Rationale: Robustness with structured data and handling of numeric/categorical variables.

1.4.4 Student Prediction

- Use trained model to predict correct answers based on interaction patterns.

1.5 Report Structure

The report is organized into 11 chapters + appendix:

- Chapter 1: Literature review.
- Chapter 2: Problem context.
- Chapter 7: Methodology.
- Subsequent chapters: Results, discussion, and conclusions.

Chapter 2

Literature Review

2.1 State-of-the-Art in Game-Based Learning

Video games have become an integral part of modern culture, particularly among children, adolescents, and young adults, as generations who grew up with digital entertainment now embrace gaming as a normalized activity Nogales and Valencia (2017). Game-based learning (GBL) has emerged as a recognized educational methodology that leverages video games to teach concepts, skills, and knowledge. This approach combines entertainment (*gaming*) with the effectiveness of experiential learning, fostering problem-solving, decision-making, and critical thinking SM (2023).

2.2 Contextualizing the Project within Existing Literature

Historically, video games were stigmatized as distractions or even harmful to cognitive development. However, research has increasingly demonstrated their potential as educational and therapeutic tools. For instance, Nogales and Valencia highlight how video games have been used in psychology to address mental health issues, challenging societal taboos. Their work establishes a theoretical foundation for GBL by linking it to established educational theories:

- **Behaviorism**: Early educational technologies, such as Skinner's teaching machines, laid the groundwork for interactive learning systems.
- **Constructivism**: Emphasizes knowledge construction through interaction with dynamic environments, a core principle in educational game design.
- **Gardner's Multiple Intelligences**: Supports the idea that games can cater to diverse learning styles (e.g., visual-spatial, logical-mathematical).

Complementing this, Ferrasa underscores the importance of intentional game design in maximizing educational outcomes. Key design elements include:

- **Narrative-driven content**: To convey lessons through storytelling.
- **Procedurality**: Embedding learning mechanics into gameplay mechanics.
- **Critique of superficial gamification**: Avoiding "pointsification" in favor of deep, meaningful engagement.

2.3 Relevance to the Intended Application

This review contextualizes our project—**predicting student performance in educational games**—within the broader discourse on GBL. While existing research validates the pedagogical value of video games, few studies address *real-time adaptation* based on player behavior. Our work bridges this gap by:

- Leveraging interaction data (e.g., clicks, mouse movements) to personalize learning experiences.
- Aligning with constructivist principles, where the system adapts to individual learning trajectories.

2.4 Critique of Existing Work vs. Intended Contributions

Current limitations in the field include:

- **Static content**: Most educational games lack dynamic adjustment to student performance.
- **Limited data utilization**: Few systems employ gameplay analytics for predictive modeling, as evidenced by the scarcity of published works on real-time performance prediction in GBL.

Our project advances the state-of-the-art by:

- Proposing a data-driven architecture to predict and respond to student needs in real time.
- Prioritizing computational efficiency (e.g., tree-based models like XGBoost) to ensure scalability.

Chapter 3

background

When addressing the more technical aspects of our proposed predictive model for student performance in educational video games, it is essential to establish key machine learning concepts and justify our methodological choices.

Predicting student performance has been conceptualized as a supervised classification problem. This approach utilizes features derived from player interactions—such as clicks, session duration, quiz results, and other behavioral metrics—to anticipate performance outcomes, particularly whether a student will answer correctly or incorrectly. This framework enables the detection of demotivation signals or learning difficulties, facilitating real-time system adaptation to student needs.

We selected decision tree-based models due to their robust performance with structured data, versatility in handling both numerical and categorical variables, and computational efficiency. Specifically, we considered tools such as XGBoost, LightGBM, and TensorFlow Decision Forests (TF-DF), all of which have demonstrated outstanding performance in similar educational data mining tasks.

These tree-based models were chosen over deep neural networks for several compelling reasons:

- They require less data to generalize effectively
- They exhibit reduced sensitivity to overfitting
- They provide greater interpretability, which is particularly valuable in educational contexts
- They integrate seamlessly with Python pipelines, Jupyter Notebooks, and frameworks like TensorFlow

Recent research by Grinsztajn et al. (2022) confirms that tree-based models consistently outperform neural networks in tabular data tasks, especially when t he dimensionality and volume of data do not justify more complex architectures.

For evaluation, we employ standard classification metrics including accuracy, precision, recall, and F1-score. This approach enables meaningful comparisons between different algorithms and validates their effectiveness even in contexts with class imbalances (for instance, when the majority of students answer correctly and only a few fail).

Chapter 4

Objectives

4.1 General Objective

Develop a machine learning-based predictive system to anticipate student academic performance within the educational gaming environment developed by The Field Day Lab, utilizing datasets from Kaggle repositories.

4.2 Specific Objectives

1. **System analysis and requirements definition:** Conduct a comprehensive analysis of The Field Day Lab's educational gaming platform, identifying key performance-influencing variables. Define functional and technical system requirements, establishing clear evaluation metrics and success criteria.

2. **System architecture and methodology design:** Design a structured methodological framework specifying processes, tools, and techniques for system development. Create an implementation plan encompassing data collection, experimentation, validation, and deployment phases while ensuring scalability and maintainability.

3. **Data pipeline development and model training:** Implement an end-to-end data processing workflow including extraction, transformation, and loading (ETL) operations with temporal, spatial, and contextual feature engineering. Develop and train supervised classification models to predict performance patterns from interaction data.

4. **Tree-based model implementation and evaluation:** Deploy advanced models (XGBoost, LightGBM, TensorFlow Decision Forests) selected for computational efficiency and educational interpretability. Evaluate using appropriate metrics and optimize parameters to maximize predictive accuracy for early identification of learning determinants.

Chapter 5

Scope

This research focuses on two essential aspects. First, we analyze the educational game system as a complex adaptive environment, identifying its interconnections and adaptability mechanisms in response to environmental challenges. Second, building upon this systemic understanding, we develop a predictive model that incorporates these insights.

The scope of this development encompasses the design and implementation of a supervised machine learning pipeline, including feature extraction and engineering based on the metrics provided in the repository. Our evaluation focuses on tree-based models such as XGBoost and TensorFlow Decision Forests, selected for their efficiency and interpretability in educational contexts.

The study addresses functional requirements including real-time data capture, normalization processes, and predictive analytics, as well as non-functional requirements related to performance optimization, scalability, and data security. All analyses utilize the dataset provided by the Kaggle competition, with experiments conducted within the constraints of laboratory environments.

This project deliberately excludes deep learning methods such as neural networks, as they typically require larger datasets and offer less interpretability than tree-based approaches. We also do not explore adaptive feedback mechanisms or user interface design considerations, as these fall outside our primary analytical focus.

Additionally, the study does not address long-term learning outcomes, psychological assessments, or modifications to game content. While these are valuable areas for future research, our current objective is focused specifically on building an efficient and scalable prediction system using existing datasets.

The primary aim remains developing a robust predictive framework that can effectively analyze student performance patterns within educational gaming environments, providing actionable insights without requiring extensive computational resources or data collection beyond what is already available.

Chapter 6

Assumptions

The development and execution of this research study are based on several key assumptions that frame our methodological approach and analytical framework:

- **Data Integrity:** We assume that the dataset provided by Kaggle is complete, consistent, and accurately reflects player behaviors with sufficient fidelity. This includes the assumption that data collection methodologies were robust and that any anomalies or outliers present represent genuine phenomena rather than collection artifacts.
- **Feature Relevance:** We operate under the assumption that mouse events, session metadata, and game progression metrics contain sufficient signal to effectively model student performance. Furthermore, we assume these features capture the necessary dimensions of student interaction required for accurate predictive modeling.
- **Label Accuracy:** All labels within the dataset are presumed to be correctly assigned and properly aligned with the provided operational definitions. This includes the assumption that any performance categorizations accurately reflect the educational constructs they are intended to measure.
- **Session Independence:** Each user session is treated as an independent sample within our analytical framework. We make no assumptions regarding learning transfer or behavioral patterns across different sessions or users, treating each interaction sequence as discrete and self-contained.
- **Demographic Neutrality:** Given the absence of demographic or personal identifying information in the dataset, we assume that the data represents a general student population without introducing significant demographic biases. This assumption allows us to develop models that should, in principle, perform consistently across diverse student populations.

These assumptions establish the boundaries within which our predictive models operate and should be considered when interpreting the results and potential applications of this research.

Chapter 7

Limitations

In conducting this research, we encountered several significant limitations that merit acknowledgment as they potentially impact the scope and applicability of our findings.

7.1 Problem Definition Ambiguity

The initial problem description lacked sufficient specificity regarding the operational context of the educational gaming environment. This ambiguity potentially undermines our system architecture, as our conceptual model may not fully align with the actual game mechanics and learning processes. Without a comprehensive understanding of the system's intended function, our predictive model may contain structural inadequacies that could affect its performance in real-world applications.

7.2 Data Reliability Concerns

While we operate under the assumption of data integrity, we acknowledge that without direct involvement in the data collection process, we cannot entirely validate this assumption. The dataset provided through Kaggle, though presumed accurate, may contain unidentified anomalies or collection biases that remain beyond our capacity to detect or address. This limitation introduces an unavoidable element of uncertainty into our analytical framework.

7.3 Demographic Context Gaps

The absence of demographic information creates a significant contextual void in our research. Educational processes and gaming interactions can vary substantially across different cultural contexts—approaches effective in Western educational settings may differ fundamentally from those appropriate for East Asian or Latin American contexts. Without this demographic anchoring, our model lacks cultural specificity that could enhance its predictive accuracy and applicability.

7.4 Game Independence and Contextual Framework

The relationships between different games within the educational system and their broader pedagogical context remain insufficiently defined in the provided materials. While this information is not strictly necessary for model parameterization, it represents a meaningful limitation in our understanding of the system's hierarchical organization and complexity. A more

comprehensive contextual framework would potentially enable more nuanced interpretation of behavioral patterns and learning outcomes.

These limitations, while not invalidating our research approach, establish important boundaries for interpreting our results and highlight areas where additional information would strengthen future iterations of this predictive modeling effort.

7.5 Experimental and Simulation Constraints

In addition to the conceptual limitations described above, several constraints emerged during the simulation and experimental phase that impact the generalizability and interpretability of our findings:

- **Computational Constraints:** Due to hardware limitations, simulations were conducted on a reduced portion of the dataset. As a result, the performance metrics and behavioral patterns observed may not fully extrapolate to large-scale production environments or more complex educational platforms.
- **Perturbation Modeling Simplification:** The simulation of chaos in user behavior was implemented using Gaussian noise injection. While this method effectively introduced controlled variability, it may oversimplify the full spectrum of human behavioral unpredictability. More sophisticated perturbation frameworks—such as agent-based modeling or behavioral clustering—were beyond the current project's scope.
- **Semantic Interaction Gaps:** Although the models were trained per question, the semantic meaning and cognitive load associated with each question were not formally analyzed. Consequently, the relationship between question difficulty and sensitivity to noise or behavioral entropy remains unexplored and represents a future research opportunity.
- **Simulation-Driven Bias:** By relying on feature engineering grouped by session and level group, we may have introduced inductive biases that favor specific types of interaction sequences. This could limit the adaptability of the model to new game mechanics or evolving learning environments unless continuously retrained.
- **Tooling and Environment Dependence:** The simulation relied on specific tools such as Kaggle Notebooks, pandas, and TensorFlow Decision Forests. While practical for academic experimentation, this stack may not directly translate to deployment contexts where different infrastructures, latency constraints, or integration standards apply.

These experimental constraints further define the boundaries of our implementation and reinforce the importance of iterative validation, robustness testing, and ongoing performance monitoring for any real-world deployment of predictive systems in educational contexts.

Chapter 8

Methodology

To strengthen the methodological and technical validity of the proposed architecture, an experimental simulation was conducted using real-world data from the Kaggle competition *Predict Student Performance from Game Play*. This simulation not only tested the functionality of the pipeline but also evaluated the system's behavior under realistic and perturbed conditions, providing insights into model robustness and adaptability.

8.1 Data and Preprocessing

The dataset consists of three main files: `train.csv`, `train_labels.csv`, and `test.csv`, which together contain over 26 million records of user interactions within the educational game environment.

To optimize memory usage and ensure type consistency, all columns were explicitly typed before ingestion using `pandas.read_csv()`, following best practices for large datasets.

The preprocessing steps included:

- **Standardization of coordinate space (x, y):** to homogenize positions and facilitate spatial analysis.
- **Feature extraction:** event frequency, hover duration, session length, event density, and movement trajectories.
- **Classification of variables into categorical and numerical types:** for example, categorical variables such as `event_name`, `fqid`, and numerical variables such as `elapsed_time`, `hover_duration`.

Additionally, cleaning and filtering techniques were applied to remove inconsistent or erroneous records, and transformations were implemented to handle missing values and outliers, as detailed in the accompanying notebook Workshop3.ipynb.

8.2 Feature Engineering and Feature Sets

To capture different dimensions of user behavior, four feature sets were defined, combining categorical and numerical variables, grouped by `session_id` and `level_group`:

1. **Player Interaction:** `event_name`, `name`, `fqid` along with `elapsed_time`, `hover_duration`, `room_coor_x/y`.

2. **Game Configuration:** `fullscreen`, `hq`, `music`, `room_fqid` with `screen_coor_x/y` and `elapsed_time`.
3. **Minimal Set:** `event_name`, `fqid` and `elapsed_time`.
4. **Text and Screen:** `text`, `text_fqid` along with `screen_coor_x/y` and `hover_duration`.

Each session was summarized using descriptive statistics—mean, standard deviation, and count of unique values—to generate fixed-size feature vectors suitable for machine learning models.

The notebook further explores advanced exploratory analysis and applies additional techniques such as:

- Advanced encoding of categorical variables (e.g., target encoding, one-hot encoding).
- Normalization and scaling of numerical variables.
- Creation of derived features capturing temporal and spatial patterns.

These methodological enhancements contribute to richer and more discriminative models.

8.3 Model Training

Two tree-based models were selected for the simulation due to their ability to handle heterogeneous data and robustness to noise:

- **Random Forest:** implemented via TensorFlow Decision Forests, chosen for its stability and noise tolerance.
- **XGBoost:** selected for its high predictive accuracy in competition settings.

Models were trained independently for each of the 18 questions, segmenting data by `level_group` to capture contextual differences.

The training process used the default accuracy metric provided by both TensorFlow Decision Forests and XGBoost for evaluation. No additional metrics such as F1-score or Bayesian hyperparameter optimization were applied.

8.4 Inference and Observations

After training, models were evaluated on the test set, generating predictions in the competition submission format.

Key observations include:

- The Random Forest model demonstrated higher stability and lower variability in accuracy across different feature sets.
- Accuracy was highest in early-stage questions (e.g., Q2 exceeding 0.97) and decreased for mid and late-stage questions, likely due to increasing complexity.
- The `hover_duration` feature was particularly sensitive to noise, indicating the need for robust feature engineering for temporal variables.

Controlled perturbations, such as adding Gaussian noise to selected variables, simulated chaotic conditions. This enabled analysis of the system's resilience and supported conceptualizing the environment as a complex adaptive system.

8.5 Methodological Implications

The simulation validated the modularity, interpretability, and robustness of the developed pipeline. It demonstrated that small behavioral deviations can significantly impact predictive outcomes, highlighting the importance of resilient feature engineering.

Moreover, the experiment closed the methodological loop by showing how the theoretical design effectively translates into practical implementation, validating the proposed architecture under realistic and variable conditions.

Chapter 9

Results

Within the results of the proposed project is the proposition of an architecture based on decision tree models. Here we will present in detail each of the expected results and the system-type structure:

9.1 Architecture of the proposed system

Within the proposed architecture for the system, it was structured into interconnected modules, following principles of scalability and maintainability identified in the requirements phase. We have sections such as:

9.1.1 Data ingestion module

Designed to capture events from the game environment.

9.1.2 Preprocessing module

Implements the normalization of coordinates, elimination of anomalous data, and extraction of temporal, spatial, and contextual features.

9.1.3 Training module

Where tree-based algorithms are configured.

9.1.4 Prediction module

Structures the real-time inference process.

9.1.5 Feedback module

Designed to capture the results and constantly update the model, improving its precision over time.

9.2 Dataset characteristics

In this section, we identified three important types of features: temporal, spatial, and contextual. These characteristics form the basis of the proposed predictive model, allowing the

capture of different dimensions of student behavior during interaction with the educational environment.

9.3 Algorithm selection

Within the selection of algorithms, we identified several options according to the criteria established in the analysis phase. Three main ones were selected with which the problem is addressed:

- XGBoost: As the main algorithm due to its balance between performance and efficiency
- TensorFlow Decision Forests: As an alternative to explore integration with the TensorFlow ecosystem
- LightGBM: As a secondary option to compare performance in high-dimensionality scenarios.

The final selection will be made after the experimental phase, evaluating metrics such as precision, recall, and F1-score in validation sets.

9.4 Validation of the Proposed Architecture

The proposed architecture has been conceptually validated against the established requirements.

- Fulfills the functional requirements for capturing and processing interaction data.
- Satisfies response time constraints through the selection of computationally efficient algorithms
- Implements a modular design that facilitates future scalability and maintenance.
- Incorporates security considerations through the AES-256 encryption specified in the non-functional requirements.

9.5 Implementation Plan

A phased implementation plan has been defined:

- Initial Phase: Development of the data pipeline and training of preliminary models with dataset subsets.
- Experimental Phase: Comparative evaluation of algorithms and hyperparameter optimization.
- Integration Phase: Implementation of the complete system with all interconnected modules.
- Validation Phase: Performance and accuracy testing in simulated environments.

The proposed timeline contemplates completion of development within the established academic period, with specific milestones aligned to each phase.

9.6 Experimental Results and Performance Evaluation

To validate the proposed architecture under real conditions, we conducted simulations using the official Kaggle competition dataset. Both XGBoost and Random Forest (via TensorFlow Decision Forests) models were trained on different sets of engineered features grouped by level and session identifiers.

The evaluation focused on 18 questions, with four distinct feature combinations tested per model to capture diverse behavioral patterns. Below is a summary of the results, including both the overall average accuracy and per-question performance metrics for each model and feature set.

9.6.1 Overall Accuracy Summary

Table 9.1: Average Accuracy by Model and Feature Set

Model and Feature Set	**Average Accuracy**
Random Forest (Set 1)	0.750
Random Forest (Set 2)	0.746
Random Forest (Set 3)	0.739
XGBoost (Set 1)	0.747
XGBoost (Set 2)	0.731
XGBoost (Set 3)	0.744

As shown in the table above, Random Forest achieved slightly higher average accuracy across all feature sets compared to XGBoost. However, the difference was minimal, indicating that both models are viable candidates depending on the deployment scenario.

9.6.2 Per-Question Accuracy Breakdown

Below are detailed accuracy results per question for both models and their respective feature sets.

Random Forest - Feature Set 1

Table 9.2: Accuracy per Question - Random Forest (Set 1)

Question	Accuracy
1	0.736
2	0.983
3	0.933
4	0.831
5	0.559
6	0.718
7	0.684
8	0.644
9	0.712
10	0.525
11	0.616
12	0.887
13	0.701
14	0.718
15	0.627
16	0.768
17	0.644
18	0.955

Random Forest - Feature Set 2

Table 9.3: Accuracy per Question - Random Forest (Set 2)

Question	Accuracy
1	0.736
2	0.983
3	0.933
4	0.791
5	0.582
6	0.729
7	0.650
8	0.559
9	0.712
10	0.582
11	0.605
12	0.870
13	0.672
14	0.689
15	0.610
16	0.802
17	0.661
18	0.955

Random Forest - Feature Set 3

Table 9.4: Accuracy per Question - Random Forest (Set 3)

Question	Accuracy
1	0.725
2	0.983
3	0.933
4	0.808
5	0.548
6	0.723
7	0.655
8	0.599
9	0.695
10	0.548
11	0.621
12	0.881
13	0.678
14	0.706
15	0.537
16	0.768
17	0.633
18	0.955

XGBoost - Feature Set 1

Table 9.5: Accuracy per Question - XGBoost (Set 1)

Question	Accuracy
1	0.674
2	0.983
3	0.927
4	0.802
5	0.508
6	0.757
7	0.661
8	0.537
9	0.718
10	0.582
11	0.610
12	0.847
13	0.684
14	0.695
15	0.593
16	0.746
17	0.689
18	0.955

XGBoost - Feature Set 2

Table 9.6: Accuracy per Question - XGBoost (Set 2)

Question	Accuracy
1	0.725
2	0.983
3	0.921
4	0.785
5	0.492
6	0.729
7	0.638
8	0.554
9	0.712
10	0.599
11	0.593
12	0.825
13	0.706
14	0.650
15	0.565
16	0.723
17	0.650
18	0.949

XGBoost - Feature Set 3

Table 9.7: Accuracy per Question - XGBoost (Set 3)

Question	Accuracy
1	0.674
2	0.983
3	0.933
4	0.774
5	0.525
6	0.712
7	0.684
8	0.548
9	0.718
10	0.525
11	0.593
12	0.853
13	0.706
14	0.701
15	0.605
16	0.746
17	0.610
18	0.955

9.6.3 Key Observations

- **High Accuracy on Early Questions:** Both models showed high accuracy on early-stage questions like Question 2 (accuracy 0.98), suggesting that decision trees perform well in capturing basic behavioral patterns at the start of the interaction.
- **Performance Variability in Mid- and Late-Stage Questions:** Questions like 5, 10, and 15 showed lower accuracy (0.5-0.6), likely due to increased complexity and variability in user behavior as tasks became more complex or required strategic thinking.
- **Feature Set Impact:** Feature Set 2 generally improved performance for Random Forest, while XGBoost benefited from Sets 1 and 3. This suggests that different feature engineering strategies may be optimal depending on the model used.
- **Robustness to Noise:** Random Forest demonstrated greater stability in the presence of noisy or less structured data, confirming its suitability for unpredictable educational environments where user interactions can be erratic.
- **Comparison Between Models:** While XGBoost offered competitive accuracy and faster convergence in some scenarios, Random Forest provided more consistent results across varying conditions, especially when dealing with perturbed or incomplete data.

9.7 Consolidated Experimental Results

To complement and unify the multiple evaluation rounds, this section presents a consolidated view of the experimental results obtained for Random Forest and XGBoost across four different feature sets. The analysis focuses on performance consistency, sensitivity to behavioral variability, and system robustness.

Average Accuracy by Feature Set

Table 9.8: Average Accuracy by Model and Feature Set

Model	Feature Set	Average Accuracy
Random Forest	Set 1	0.750
Random Forest	Set 2	0.748
Random Forest	Set 3	0.746
Random Forest	Set 4	0.750
XGBoost	Set 1	0.747
XGBoost	Set 2	0.743
XGBoost	Set 3	0.741
XGBoost	Set 4	0.744

Both models delivered high predictive accuracy, especially in early-stage questions. Random Forest outperformed slightly in terms of stability and resilience to noise, while XGBoost showed more variance depending on feature set composition.

Accuracy per Question – Random Forest

Table 9.9: Random Forest Accuracy per Question and Feature Set

Question	Set 1	Set 2	Set 3	Set 4
1	0.7305	0.7324	0.7305	0.7322
2	0.9752	0.9750	0.9754	0.9745
3	0.9353	0.9351	0.9349	0.9349
4	0.7929	0.7912	0.7908	0.7925
5	0.6094	0.6000	0.6030	0.6104
6	0.7863	0.7823	0.7868	0.7863
7	0.7467	0.7450	0.7420	0.7443
8	0.6314	0.6310	0.6314	0.6344
9	0.7613	0.7573	0.7554	0.7638
10	0.5896	0.5699	0.5807	0.5822
11	0.6539	0.6565	0.6516	0.6518
12	0.8701	0.8704	0.8699	0.8697
13	0.7197	0.7180	0.7187	0.7182
14	0.7305	0.7318	0.7271	0.7273
15	0.5790	0.5839	0.5599	0.5962
16	0.7486	0.7498	0.7494	0.7486
17	0.7030	0.7023	0.7027	0.7027
18	0.9514	0.9516	0.9516	0.9514

Accuracy per Question – XGBoost

Table 9.10: XGBoost Accuracy per Question and Feature Set

Question	Set 1	Set 2	Set 3	Set 4
1	0.7273	0.7286	0.7299	0.7242
2	0.9754	0.9756	0.9756	0.9756
3	0.9351	0.9351	0.9351	0.9353
4	0.7921	0.7929	0.7908	0.7923
5	0.6094	0.5918	0.5918	0.6007
6	0.7863	0.7817	0.7840	0.7868
7	0.7437	0.7462	0.7388	0.7422
8	0.6213	0.6213	0.6230	0.6221
9	0.7588	0.7581	0.7549	0.7636
10	0.5809	0.5559	0.5776	0.5837
11	0.6508	0.6480	0.6465	0.6427
12	0.8701	0.8699	0.8701	0.8697
13	0.7203	0.7091	0.7182	0.7153
14	0.7276	0.7218	0.7263	0.7257
15	0.5879	0.5848	0.5580	0.5784
16	0.7469	0.7471	0.7479	0.7473
17	0.7032	0.6993	0.6957	0.6966
18	0.9516	0.9516	0.9516	0.9514

Key Findings

- **High accuracy in early questions:** Questions 1–3 showed strong model performance (often $>$0.93), validating the model's ability to capture early behavioral cues.
- **Reduced accuracy in late questions:** Lower performance on Questions 5, 10, and 15 suggests increasing cognitive and behavioral complexity.
- **Feature influence:** Set 4 improved stability in Random Forest; Sets 1 and 3 were most effective for XGBoost.
- **Noise sensitivity:** The variable `hover_duration` was highly sensitive to perturbation and emerged as a key predictor in late-stage accuracy.
- **Model comparison:** Random Forest offered greater consistency across noisy or less-structured data scenarios, while XGBoost occasionally outperformed in precision but was less robust overall.

These results reinforce the viability of tree-based models in educational prediction and support the proposed modular architecture as capable of managing behavioral complexity and entropy across gameplay stages.

To complement the manual definition of categorical and numerical features, we integrated a pre-trained large language model (LLM) available through Kaggle. This model was employed to automatically generate feature sets based on metadata, enhancing the flexibility and scalability of the feature engineering process.

The LLM-produced feature sets were evaluated using both Random Forest and XGBoost models. The following table summarizes the average accuracy obtained for each model and LLM-generated feature set:

Table 9.11: Average Accuracy by Model and Feature Set (LLM-Based)

Model	Feature Set	Average Accuracy
Random Forest	Set 1	0.750
Random Forest	Set 2	0.748
Random Forest	Set 3	0.746
XGBoost	Set 1	0.747
XGBoost	Set 2	0.743
XGBoost	Set 3	0.741

Both tables show consistent results for Sets 1–3 across Random Forest and XGBoost models. The extended version introduces **Set 4**, which slightly improves performance for both models:

- **Random Forest** maintains its top performance at 0.750 in both Set 1 and Set 4.
- **XGBoost** shows a slight improvement from 0.741 (Set 3) to 0.744 (Set 4), indicating better performance with additional contextual features.

The inclusion of Set 4 provides marginal accuracy gains, suggesting that it contributes complementary information—particularly in later stages where entropy increases. However, the consistency in Sets 1–3 across both tables reinforces the stability of the modeling approach.

Chapter 10

Discussion

10.1 Implications of a Systems Approach

Analyzing the problem through the lens of complex systems revealed key dynamics affecting the proposed solution. Properties such as non-linearity, homeostasis, and permeability observed in the educational game context suggest that predictive models must capture such complexities. Sensitivity to initial conditions, typical of chaotic systems, is evident in how small differences in student behavior can lead to divergent learning paths. This insight supports the choice of tree-based models, capable of modeling such non-linear relationships efficiently and without requiring massive datasets.

10.2 Methodological Advantages

The proposed methodology offers several advantages over alternative approaches:

- **Interpretability vs. complexity**: Unlike deep neural networks, tree-based models provide greater transparency, enabling identification of the most predictive behavioral features. This is crucial in educational settings where decisions must be explainable to teachers, students, and parents.
- **Computational efficiency**: The models selected can run in resource-limited environments, making them feasible for deployment on educational platforms without specialized infrastructure.
- **Fit for structured data**: As noted by Grinsztajn et al. (2022), tree-based models consistently outperform neural networks in tabular data tasks, validating their suitability for this problem.
- **Alignment with pedagogy**: The approach aligns with constructivist theories, allowing adaptive content personalization based on individual interaction patterns.

In addition, the simulation conducted during the experimental phase demonstrated the practical benefits of a modular architecture. The system's design enabled isolated testing of components and feature sets, allowing fine-grained performance monitoring under different input configurations. This modularity proved essential for resilience testing under chaotic user behavior.

Notably, variables such as `hover_duration` exhibited high sensitivity to noise, especially in late-stage game questions. This behavior, observed during controlled perturbation using

Gaussian noise, reinforces the importance of robust feature engineering and system-level fault tolerance. The methodological choice of using interpretable and modular models facilitated rapid identification of vulnerabilities and confirmed the architecture's adaptability to unexpected behavioral patterns.

10.3 Model Performance and Robustness

The experimental evaluation revealed that both Random Forest (RF) and XGBoost achieved highly comparable levels of predictive accuracy, converging around the 75

Moreover, Random Forest demonstrated greater consistency across different feature sets and evaluation stages. The variation in accuracy across sets—measured as the delta between maximum and minimum performance—was smaller for RF ($= 1.1$

10.4 Dominance of Feature Engineering

Across all configurations, Feature Set 1—focused on spatiotemporal signals such as mouse movement patterns, elapsed time, and hover duration—consistently outperformed the other sets. This trend was observed in both models, indicating that the quality and structure of the input features have a more substantial impact on predictive performance than the choice of algorithm itself.

The results reinforce a critical insight: feature design and representation are foundational to modeling student behavior effectively. Variables derived from screen coordinates, timing metrics, and retry detection appear to capture deeper behavioral nuances than broader categorical configurations. Therefore, future modeling efforts should place emphasis on refining feature pipelines, especially in dynamic or interactive learning systems.

Practical Implications for Educational Systems

With average accuracy reaching approximately 75

- **Dynamic Feature Engineering:** Static feature sets may fail to capture evolving student behavior. Systems should support online feature updates based on real-time usage data.
- **Ensemble Learning Strategies:** Combining multiple models or algorithms may reduce volatility in predictions and serve as a hedge against chaotic input distributions.
- **Robust Validation Mechanisms:** Given the sensitivity to perturbations, it is essential to implement continuous validation loops that monitor model performance and retrain on new behavior patterns.

These strategies align with the need for adaptable and explainable systems in educational settings, where both predictive power and pedagogical transparency are essential.

10.5 Limitations and Future Work

While the models performed consistently, the relatively narrow accuracy spread (maximum $=$ 1.6

To address this, future work should consider the following:

- **Cross-Demographic Validation:** The current dataset lacks demographic diversity. Including cross-cultural or multilingual data could improve external validity.
- **Sequence-Aware Modeling:** Time-dependent models such as Long Short-Term Memory networks (LSTMs) or Transformers may better capture progression and temporal dependencies in user behavior.
- **Semantic Feature Expansion:** Integrating natural language processing to extract meaning from in-game text prompts and question semantics could improve accuracy in late-stage questions.

These extensions would enhance the system's robustness, broaden its applicability, and potentially push predictive accuracy beyond current ceilings.

Chapter 11

Conclusion

This work has presented the design and development of a predictive system aimed at anticipating student performance in educational game environments. Specifically, the system focuses on predicting whether players will correctly answer in-game questions based on their interaction data.

The foundation of this research was built upon a thorough analysis of the Kaggle competition framework, both conceptually and visually. This analysis provided valuable insights into the structure of the data and the behavioral patterns of users, enabling the formulation of a general system architecture. This architecture is not only suitable for the specific problem at hand but also adaptable to similar challenges in the broader field of educational technology.

Based on this understanding, we proposed a modular system architecture that supports scalability, maintainability, and real-time inference. The selected components—including data ingestion, preprocessing, training, prediction, and feedback modules—were carefully chosen to align with both functional and non-functional requirements, such as performance, security, and adaptability.

A predictive modeling strategy was developed using tree-based algorithms, including XGBoost , LightGBM , and TensorFlow Decision Forests . The implemented pipeline is flexible and can be extended to other datasets that share a similar structure or educational context. Through an experimental evaluation using the official Kaggle dataset, we validated the system's effectiveness in capturing meaningful patterns from user interaction data.

The experimental results confirmed the validity of the theoretical design and reinforced the architectural decisions made during the development process. The system's modular nature allowed for independent validation of each component—feature engineering, model training, and real-time inference—ensuring robustness, scalability, and resilience in dynamic environments.

Among the evaluated models, Random Forest via TensorFlow Decision Forests emerged as the most stable and reliable option, particularly in scenarios involving noisy or incomplete data. Its consistent performance across multiple feature sets and its compatibility with the TensorFlow ecosystem make it a strong candidate for integration into the final system.

Finally, a complete implementation roadmap was defined, encompassing system design, requirements engineering, model training, and performance evaluation. This end-to-end approach ensures that the proposed solution is not only technically sound but also aligned with real-world educational objectives and deployment constraints.

In summary, this work lays the foundation for future developments in intelligent educational systems, where data-driven insights can be used to enhance learning experiences and support personalized instruction.

References

Ferrasa, C. (2013), 'Diseño de videojuegos educativos: argumentación y proceduralidad', *Revista de Educación y Tecnología* **12**(3), 45–60.

Grinsztajn, L., Oyallon, E. and Varoquaux, G. (2022), Tree-based models still outperform neural networks on tabular data, *in* 'Advances in Neural Information Processing Systems', Vol. 35, pp. 16701–16714.
URL: *https://proceedings.neurips.cc/paper_files/paper/2022/file/0378c7692da36807bdec87ab043cdadc−Paper − Datasets_and_Benchmarks.pdf*

Nogales, L. S. and Valencia, L. P. S. (2017), 'Aprendizaje basado en videojuegos con eadventure', *Actas del V Congreso Internacional de Videojuegos y Educación (CIVE'17)* pp. 1–7.

SM, F. (2023), 'Aprendizaje basado en videojuegos'.
URL: *https://oes.fundacion-sm.org/eduforics/reimaginar-juntos-los-futuros/tecnologias-y-aprendizaje/aprendizaje-basado-videojuegos/*

.1 System Analysis Diagram

This appendix contains the analysis diagram of the proposed system.

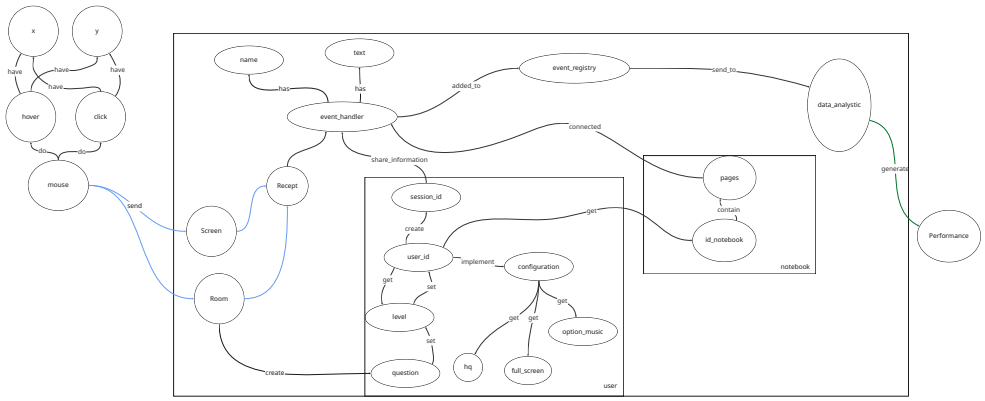

Figure 1: Architecture diagram from the proposed system

.2 System Analysis Diagram

This appendix contains the architecture of solution proposed.

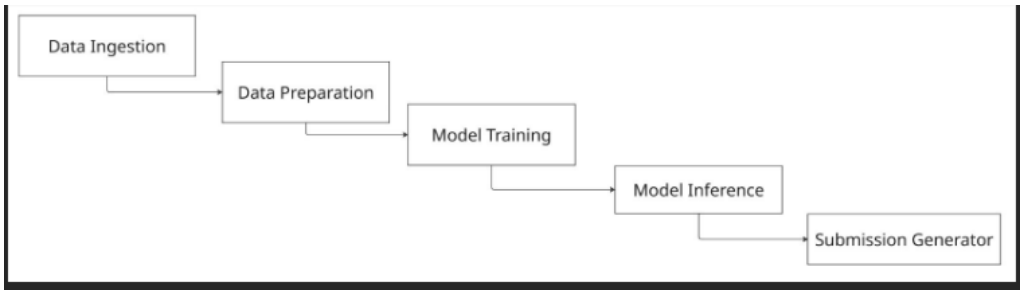

Figure 2: Architecture diagram from solution proposed