

Workshop 2

Henry Ricaurte Mora 20221020084
Germán Darío Aya Fuentes 20232020091
Javier Alejandro Penagos Hernández 20221020028

1 Review Workshop #1 Findings

2 Define System Requirements

2.1 Functional Requirements

2.1.1 Data Capture and Storage

-
- RF-001** Capture all user interactions including hovers, clicks, and drags with their respective `x,y` coordinates and timestamp.
 - RF-002** Assign a unique `session_id` per game session.
 - RF-003** Link all interactions to a specific `user_id`.
 - RF-004** Store user configuration settings including `full_screen`, `hq`, and `music_volume`.
 - RF-005** Save the `level_group` and question progress.

2.1.2 Data Processing and Normalization

- RF-006** Remove erroneous or duplicate clicks (condition: ≥ 90 ms between clicks).
- RF-007** Normalize `x,y` coordinates by standardizing to a key resolution.
- RF-008** Extract temporal features: time between events, response speed.
- RF-009** Extract spatial features: movement patterns like trajectories.
- RF-010** Extract contextual features: difficulty level and number of retries.

2.1.3 Prediction Model

- RF-011** `screen_coord_x/y`: Mouse position at critical questions.
- RF-012** `event_name`: Actions like `cutscene_click` or `map_click`.
- RF-013** `elapsed_time`: Cumulative time in session.
- RF-014** `hover_duration`: Time spent on interactive elements.

2.2 Non-Functional Requirements

2.2.1 Performance

- RNF-001** The system must capture and store user interactions in real time without affecting the user experience.
- RNF-002** Data preprocessing and normalization must not exceed 500 ms per batch of captured events.

2.2.2 Reliability

RNF-003 The system must guarantee 99.9% availability during gameplay sessions.

RNF-004 An integrity check must be implemented for each processed data block.

2.2.3 Security

RNF-005 All sensitive data (`user_id`, `session_id`) must be stored and transmitted using AES-256 encryption.

RNF-006 The system must implement role-based access control for data management and visualization.

2.2.4 Ease of Use

RNF-007 The user configuration interface must be accessible and intuitive, allowing easy adjustments to parameters such as volume, resolution, and fullscreen mode.

RNF-008 Key metrics and model outputs must be visualized through interactive dashboards supported by tools such as **Grafana**, facilitating interpretation by non-technical users.

2.2.5 Interoperability

RNF-009 The system must integrate seamlessly with analysis platforms such as **Jupyter**, **Apache Kafka**, **Grafana**, and relational or NoSQL databases.

RNF-010 The predictive model must be exportable in ONNX format or similar, ensuring portability across languages and frameworks.

3 High-Level Architecture

4 Addressing Sensitivity and Chaos

In this section, we address the sensitive variables and chaotic factors identified during the system analysis in Workshop 1. The student's performance in this context is modeled based on their interactions with the educational game. Multiple sources of variability were identified, including unpredictable user behavior, variable initial conditions, and random in-game events.

In terms of user behavior, we observed fast or erratic clicks, irregular mouse movements, and non-linear navigation paths. Regarding initial conditions, factors such as emotional state, motivation level, and familiarity with the game can significantly affect user interaction. Random events may include variations in question difficulty or game elements that introduce noise into the dataset.

The mitigation strategies implemented to address these challenges include:

1. Data Normalization and Preprocessing:

- Elimination of erroneous and duplicate clicks.
- Standardization of x, y coordinate space.
- Extraction of temporal features.

2. Use of TensorFlow Decision Forests (TF-DF):

TF-DF is robust to noisy data and can handle both categorical and numerical features without requiring additional encoding. The ensemble nature of decision forests enables the model to capture non-linear and complex relationships among variables, thus improving generalization capabilities.

3. Monitoring of Unanticipated Conditions:

Implementation of observation routines to detect unusual patterns or anomalies in the interaction data.

5 Technical Stack and Implementation Sketch