Decoding Data Science(DDS) Academy
AI Application Building Challenge

Idea Submission

Project Title:

AI-Driven System for Student Outcomes Assessment and Continuous Improvement

Concept Summary:

AI-Driven System for Student Outcomes Assessment and Continuous Improvement integrates artificial intelligence to streamline the evaluation of student outcomes (SOs) in educational institutions. By automating data analysis, applying rule-based decision-making, and generating actionable insights, the system ensures accurate, consistent, and transparent assessments. It not only evaluates SOs but also identifies underperforming areas, providing targeted improvement recommendations to enhance curriculum design and academic performance.

Target Audience:

Educational Administrators: Seeking data-driven tools for curriculum assessment and improvement.
Instructors: Aiming for consistent and objective evaluation of student outcomes.
Students: Benefiting indirectly through targeted curriculum enhancements and improved learning experiences.

Key Features:

Rule-Based Expert System: Uses predefined logic to evaluate Performance Indicators (PIs) and student outcomes.

Automated Data Processing: Normalizes and analyzes assessment data to generate meaningful insights.

Targeted Recommendations: Offers specific plans for underperforming PIs and SOs to drive improvement.

Dynamic Reports: Generates detailed achievement summaries with categorized performance levels (e.g., Needs Improvement, Satisfactory, Excellent). **Scalability**: Applicable across programs, campuses, and varying academic structures.

Technical Approach:

1. Knowledge Base: Stores rules and facts for evaluating PIs and SOs, including predefined scoring thresholds, rubrics, and performance benchmarks.
2. Inference Engine: Employ forward-chaining reasoning to assess outcomes and generate targeted recommendations for improvement.
3. KNIME Integration: The system uses KNIME as a no-code/low-code platform for advanced data workflows, including preprocessing assessment data, implementing rule-based logic, and generating predictive analytics for continuous improvement.
4. Database: PostgreSQL for structured data storage, enabling efficient queries and analytics.
5. Visualization: KNIME's built-in visualization capabilities, along with tools like Matplotlib and Power BI, are used to create insightful dashboards and reports that highlight areas of excellence and improvement.

Expected Challenges:

Data Integration: Difficulty in gathering standardized assessment data from diverse systems.

- **Solution**: Develop APIs and data mapping techniques for seamless integration. **AI Transparency**: Gaining trust in AI-driven evaluations. • **Solution**: Incorporate explainable AI methods to clarify decision-making processes.

Submission Format:

- Detailed project proposal outlining the system's design and implementation.
- A presentation showcasing the problem, solution, and impact.
- Interactive demo with sample data to illustrate the system's capabilities.

Expected Outcome:

- A functional AI-driven prototype capable of assessing student outcomes and generating actionable recommendations.
- Improved efficiency and accuracy in evaluating educational programs.

- A roadmap for integrating AI in continuous academic improvement processes.

Additional Notes (Optional):

Future extensions could include real-time integration with LMS platforms and predictive analytics for student performance.

Environment Setup

1. Infrastructure Preparation: o Server/Cloud: Set up a cloud environment (e.g., AWS, Azure, or GCP) or a local server with sufficient storage and compute power for hosting KNIME and PostgreSQL.
o Database Setup: Install and configure PostgreSQL for storing assessment data, rubrics, and benchmarks.

o Development Tools:

- Install KNIME for workflow creation and data processing.
- Set up Python for advanced analytics and integration with Matplotlib or other libraries.
- Install Power BI or Tableau for visualization needs.

2. Integration Framework: o Develop APIs to enable seamless data transfer between the system, LMS platforms, and other academic tools.
o Define data mapping techniques to standardize diverse assessment data formats. Initial Development

1. Data Collection and Preparation:

- Gather sample assessment data for Performance Indicators (PIs) and Student Outcomes (SOs).
- Preprocess data using KNIME, ensuring it is clean, normalized, and ready for analysis.

2. Knowledge Base Creation:

- Define the rules and thresholds for evaluating PIs and SOs.
- Store these rules in the knowledge base within KNIME or PostgreSQL.

3. Inference Engine:

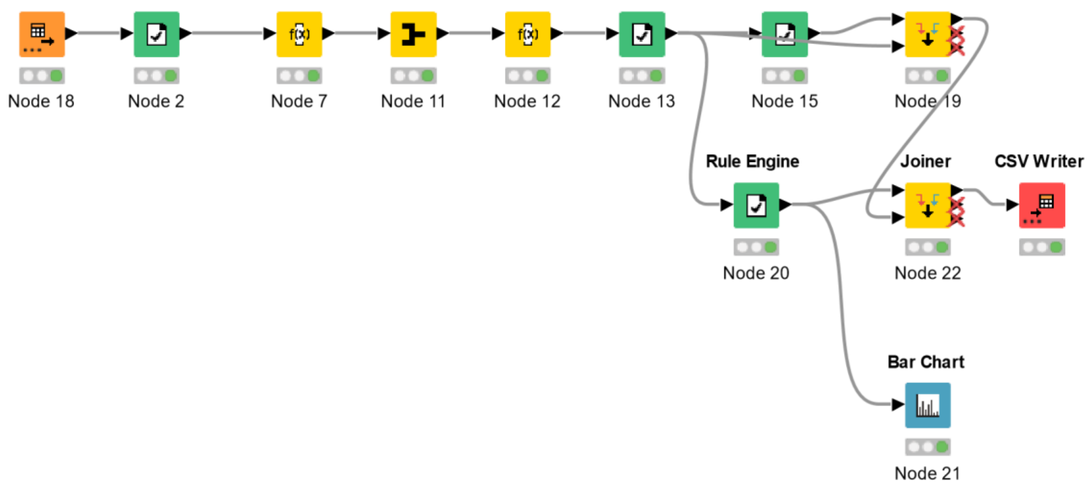
- Build forward-chaining logic in KNIME to evaluate PIs and generate recommendations.
- Test the engine with sample data to validate its decision-making capabilities.

4. Visualization and Reporting:

- Develop interactive dashboards and reports using KNIME and Power BI.
- Include categorized performance levels (Needs Improvement, Satisfactory, Excellent) to provide actionable insights.

5. Prototype Development: ○ Create a functional prototype integrating all components: data processing, rule-based evaluations, and reporting.

○ Ensure the system is modular for scalability and future extensions.



Dataset

1. **Source**: Assessment data from educational institutions.
2. **Structure**:

○ Performance Indicators (PIs) mapped to Student Outcomes (SOs). ○ Predefined scoring thresholds and rubrics.

○ Student performance data categorized by academic programs, courses, and individual achievements.

3. **Challenges**:

○ Standardization of data formats from diverse systems.

○ Integration of historical and current datasets.

Project Report

1 Project Title:

AI-Driven System for Student Outcomes Assessment and Continuous Improvement

2 Introduction

Student Outcomes (SOs) serve as critical benchmarks for evaluating the effectiveness of educational programs in preparing students for professional and academic success. Traditional methods for assessing SOs often involve labor-intensive processes, subjective evaluations, and limited scalability. In recent years, the advent of AI tools has provided an unprecedented opportunity to transform these practices by offering automated, accurate, and actionable insights into student performance and program effectiveness. This paper investigates how AI tools can revolutionize SO assessment by enabling data-driven decision-making and continuous improvement. Specifically, the study examines the practical application of AI technologies in analyzing diverse student outputs, providing feedback, and informing the needed adjustments.

The proposed SOA employs principles of expert systems by utilizing rule-based decision-making to evaluate the performance indicators (PIs) and align them with SOs. Specifically, the use of rubrics and weighted scoring criteria mirrors the inference mechanisms of expert systems, enabling automated, consistent, and transparent assessments. This system, like functionality supports the identification of areas requiring improvement, generates actionable recommendations, and ensures that decisions are aligned with predefined educational objectives.

Fig. 1 shows the workflow of SOA; stages of the expert system in the AI-Driven System for Student Outcomes Assessment (SOA)

1.1 **Knowledge Base:**

Stores a structured set of rules and facts regarding Performance Indicators (PIs) and Student Outcomes (SOs). For example, rules define thresholds and scoring logic for achieving specific PIs or SOs.

1.2 **Logic Implementation:**

Inference Engine processes the rules and facts to draw logical conclusions about whether specific PIs and SOs have been achieved. It uses the Forward Chaining reasoning method, it starts with known facts and apply rules to infer new facts or conclusions. For instance, it evaluates if scores meet the PI threshold to determine whether a PI is achieved. The system starts with known data inputs (e.g., scores, thresholds, weights). It applies predefined rules to assess whether specific Performance Indicators (PIs) are achieved. Conclusions (e.g., PI_Achieved = True/False, recommendations) are derived step by step from the facts. SOA iterates through the dataset, applying rules to each PI and aggregating results to evaluate Student Outcomes (SOs). This progression mirrors the forward chaining mechanism. Each rule determines intermediate outcomes (e.g., whether a PI meets its threshold). These outcomes contribute to achieving higher-level goals, such as meeting SO thresholds or generating tailored improvement plans. This method supports systematic and unbiased decision-making making it well-suited for the structured, rule-driven evaluation processes highlighted in AI-driven SOA system.

Rule Application: Checks if a student's score meets the defined thresholds for PIs and calculates weighted scores to assess the overall achievement of SOs. Score Normalization: Aggregates scores to normalize the contributions of various PIs, ensuring consistency in evaluation. Recommendation Generation: Based on the results, generates targeted improvement plans or recommendations, such as workshops or curriculum adjustments.

1.3 Output:

SOA Provides a detailed summary of SO achievements, including a categorization of performance levels (e.g., Needs Improvement, Satisfactory, Excellent). It generates specific recommendations for underperforming PIs and SOs to guide continuous improvement.

3 Experiments

The integration of rule-based logic in SO assessment aligns with expert system methodologies, where predefined rules evaluate performance data and provide targeted feedback. For example, the automated classification of performance into 'Needs Improvement,' 'Satisfactory,' or 'Excellent' mirrors the inference engine of an expert system, ensuring systematic and unbiased decisionmaking. By incorporating these elements, the assessment framework effectively bridges the gap between AI technologies and traditional expert system principles, enhancing both accuracy and applicability.

1.4 Define Student Outcomes (SOs): The following SOs as an example:

SO1: Apply problem-solving skills to design IT solutions. SO2: Develop secure network systems.

- Break Down SOs into Performance Indicators (PIs): SO1:
  PI1.1: Analyze a problem and define requirements.
  PI1.2: Design algorithms to address a given problem.
  SO2:
  PI2.1: Analyze network vulnerabilities and propose appropriate security measures. PI2.2: Design and implement a secure network system using industry-standard protocols and tools.

- Assign Assessment Tools to PIs: SO1:
  PI1.1: Assignment 1 (Requirements Analysis).
  PI1.2: Midterm Exam (Algorithm Design).
  SO2:
  PI2.1: Project Report (Secure Network Design). PI2.2: Final Exam (Practical Implementation).

1.5 Input Student Assessment Data:

The student assessment data stage involves recording detailed scores for each task or question related to various assessments in the curriculum as shown in Table1. This data serves as the foundation for evaluating Performance Indicators (PIs) and subsequently determining the achievement of Student Outcomes (SOs). The process ensures accurate and systematic evaluation by mapping individual assessments to corresponding PIs and SOs. Key Components of the Data

- SO_Code: Identifies the specific Student Outcome (SO) being evaluated. Each SO represents a broader educational goal tied to program objectives.
- PI_Code: Denotes the Performance Indicator (PI) linked to the respective SO. PIs are measurable elements that collectively contribute to the achievement of an SO.
- Weight: Represents the relative importance of the task or question within the overall assessment of the PI. It is used to calculate weighted scores.
- Assessment_Code: Specifies the unique identifier for the assessment (e.g., assignment, exam) used to evaluate the PI.
- Task/Question: Details the specific task or question within the assessment that aligns with the PI.
- PI_Threshold: The minimum score required to achieve the PI, indicating whether the student meets the predefined performance standard.
- SO_Threshold: The minimum score required to achieve the overall SO. • Score: The raw numeric score obtained by the student for the task or question**.**

Table 1. Performance Indicator (PI) Dataset

SO_Code	PI_Code	Weight	Assessment_Code	Task/Question	PI_Threshold	SO_Threshold	**Score**
SO1	PI1.1	0.4	Assignment 1_IT112	Task1	75	80	70
SO1	PI1.2	0.6	Midterm _IT341	Question1	75	80	85
SO2	PI2.1	0.5	Project_IT321	Task1	75	85	80
SO2	PI2.2	0.5	Final Exam_IT112	Question4	80	85	88
SO3	PI3.1	0.5	Presentation_IT234	Task1	80	80	92
SO3	PI3.2	0.5	Quiz2_IT341	Question 2	80	80	87
SO4	PI4.1	0.4	Final Exam_IT321	Question 1	80	75	78
SO4	PI4.2	0.6	Assignment 2_IT123	Task2	80	75	83

1.6 Aggregate PI Results to Measure SO Achievement

Each PI Score is weighted based on its importance in contributing to the overall SO achievement. The weighted score for each PI is calculated using the formula:

Weighted_Score = Score * Weight (1)

After calculating the weighted scores for all PIs associated with a particular SO, the total weighted score is normalized to account for the varying weights of each PI. This normalization ensures that the contributions of all PIs are properly scaled.

Normalized_Weighted_Score = Sum (Weighted_Score) / Sum (Weight) (2)

The Normalized_Weighted_Score is compared against a pre-defined PI Threshold to determine whether the SO is achieved:

- If the Normalized_Weighted_Score meets or exceeds the PI Threshold, the SO is considered achieved, represented as True.

$Normalized_Weighted_Score$ >= $PI_Threshold$ =>"True"

(3)

- If it falls below the threshold, the SO is not achieved, represented as False

$Normalized_Weighted_Score$ < $PI_Threshold$ =>"False"

(4)

Once the SO is achieved, its performance level is categorized based on the value of the **Normalized_Weighted_Score**. This categorization helps in assessing the overall quality of the outcome:

$Normalized_Weighted_Score$>= 90=> "Excellent"
$Normalized_Weighted_Score$< 90 AND $Normalized_Weighted_Score$>= 80 =>"Satisfactory"
$Normalized_Weighted_Score$< 80 AND $Normalized_Weighted_Score$ >= 70=> "Needs Improvement"

1.7 Apply Rule-based decision-making: Logic for underperforming PIs (Enhancement Plan):

$PI Achieved$ = 1 AND PI_Code = "PI1.1" => "Enhance existing workshops by including advanced identification techniques and hands-on projects for hardware assembly. Introduce interactive simulations with progress tracking for better engagement."

$PI Achieved$ = 1 AND PI_Code = "PI1.2" => "Expand mathematical problem-solving competitions to include interdisciplinary challenges. Provide advanced case studies and foster peer collaboration to strengthen problem-solving capabilities."

$PI Achieved$ = 1 AND PI_Code = "PI2.1" => "Scale hackathons to cover more complex realworld problems and include industry mentors for guidance. Incorporate reflective sessions for students to analyze their problem decomposition strategies."

$PI Achieved$ = 1 AND PI_Code = "PI2.2" => "Integrate advanced tools and frameworks into the curriculum through specialized workshops. Conduct scenario-based learning activities where students justify tool selection and application."

$PI Achieved$ = 1 AND PI_Code = "PI2.3" => "Introduce peer-reviewed group projects focused on innovative computing solutions. Facilitate student-led brainstorming sessions to promote collaborative evaluations."

$PI Achieved$ = 1 AND PI_Code = "PI3.1" => "Expand prototyping bootcamps to include realworld case studies and client interactions. Establish partnerships with local industry to provide realtime feedback on student prototypes."

$PI Achieved$ = 1 AND PI_Code = "PI3.2" => "Enhance the peer-review system with structured rubrics and automated feedback tools. Provide additional resources for debugging, including casebased examples and expert tutorials."

$PI Achieved$ = 1 AND PI_Code = "PI4.1" => "Upgrade cybersecurity training series by introducing real-world attack simulations and response strategies. Partner with industry experts to provide guest lectures and certification opportunities."

$PI Achieved$ = 1 AND PI_Code = "PI4.2" => "Expand cybersecurity drills to include penetration testing and vulnerability analysis exercises. Develop case studies that emphasize compliance with industry standards and best practices."

Logic for Underperforming PIs (Improvement Plan):

$PI Achieved$ = 0 AND PI_Code = "PI1.1" => "Initiate interactive, hands-on workshops to teach students to identify and categorize various hardware components effectively. Develop an online simulation platform where students can virtually assemble hardware and get real-time feedback."

$PI Achieved$ = 0 AND PI_Code = "PI1.2" => "Design advanced mathematical problem-solving competitions and include real-world case studies in assignments. Provide weekly problem-solving sessions focused on computational techniques."

$PI Achieved$ = 0 AND PI_Code = "PI2.1" => "Conduct hackathons where students practice breaking down complex computing problems into manageable tasks. Introduce guided tutorials on modular problem-solving methodologies."

$PI Achieved$ = 0 AND PI_Code = "PI2.2" => "Organize tool-specific training sessions to familiarize students with advanced computing tools. Provide real-world case studies requiring students to select and justify the most suitable tools for given problems."

$PI Achieved$ = 0 AND PI_Code = "PI2.3" => "Conduct peer reviews for evaluating solution feasibility."

$PI Achieved$ = 0 AND PI_Code = "PI3.1" => "Launch a prototyping bootcamp where students develop innovative software solutions based on client requirements. Establish mentorship programs with industry professionals to refine prototype design skills."

$PI Achieved$ = 0 AND PI_Code = "PI3.2" => "Introduce a structured peer-review system for prototype testing in class. Develop a checklist-driven testing framework for students to use while debugging their prototypes"

$PI Achieved$ = 0 AND PI_Code = "PI4.1" => "Organize an advanced training series on implementing cybersecurity measures, including encryption and firewalls. Develop team-based projects where students analyze and reinforce security protocols in simulated environments."

$PI Achieved$ = 0 AND PI_Code = "PI4.2" => "Conduct cybersecurity drills where students identify and mitigate vulnerabilities in existing systems. Provide access to industry-standard tools for vulnerability assessment to enhance technical skills."

Output of SOA in Table2. categorizes achievements for each PI and SO, generating actionable plans for improvement.

Table2. Output of SOA

PI_Code	**PI1.1**	**PI1.2**	**PI2.1**	**PI2.2**
PI_Threshold	75	75	75	80
SO_Threshold	80	80	85	85
PI Achieved	0	1	1	1
Sum(Weighted_Score)	28	51	40	44
Sum(Weight)	0.4	0.6	0.5	0.5
Normalized_Weighted_Score	70	85	80	88
PI_Achieved	FALSE	TRUE	TRUE	TRUE
Student Outcome (SO) Achievement Status	Needs Improvement	Satisfactory	Satisfactory	Satisfactory
Recommendation and Improvement Plan	Initiate interactive, hands-on workshops to teach students to identify and categorize various hardware components effectively. Develop an online simulation platform where students can virtually assemble hardware and get real-time feedback.	Enhancement Plan_Expand mathematical problem-solving competitions to include interdisciplinary challenges. Provide advanced case studies and foster peer collaboration to strengthen problem-solving capabilities.	Enhancement Plan_Scale hackathons to cover more complex real-world problems and include industry mentors for guidance. Incorporate reflective sessions for students to analyze their problem decomposition strategies.	Enhancement Plan_Integrate advanced tools and frameworks into the curriculum through specialized workshops. Conduct scenario-based learning activities where students justify tool selection and application.
PI_Code	**PI3.1**	**PI3.2**	**PI4.1**	**PI4.2**
PI_Threshold	80	80	80	80
SO_Threshold	80	80	75	75
PI Achieved	1	1	0	1
Sum(Weighted_Score)	46	43.5	31.2	49.8

Sum(Weight)	0.5	0.5	0.4	0.6
Normalized_Weighted_Score	92	87	78	83
PI_Achieved	TRUE	TRUE	FALSE	TRUE
Student Outcome (SO) Achievement Status	Excellent	Satisfactory	Needs Improvement	Satisfactory
Recommendation and Improvement Plan	Expand prototyping bootcamps to include real-world case studies and client interactions. Establish partnerships with local industry to provide real-time feedback on student prototypes.	Enhancement Plan_Enhance the peerreview system with structured rubrics and automated feedback tools. Provide additional resources for debugging, including case-based examples and expert tutorials.	Organize an advanced training series on implementing cybersecurity measures, including encryption and firewalls. Develop teambased projects where students analyze and reinforce security protocols in simulated environments.	Enhancement Plan_Expand cybersecurity drills to include penetration testing and vulnerability analysis exercises. Develop case studies that emphasize compliance with industry standards and best practices.

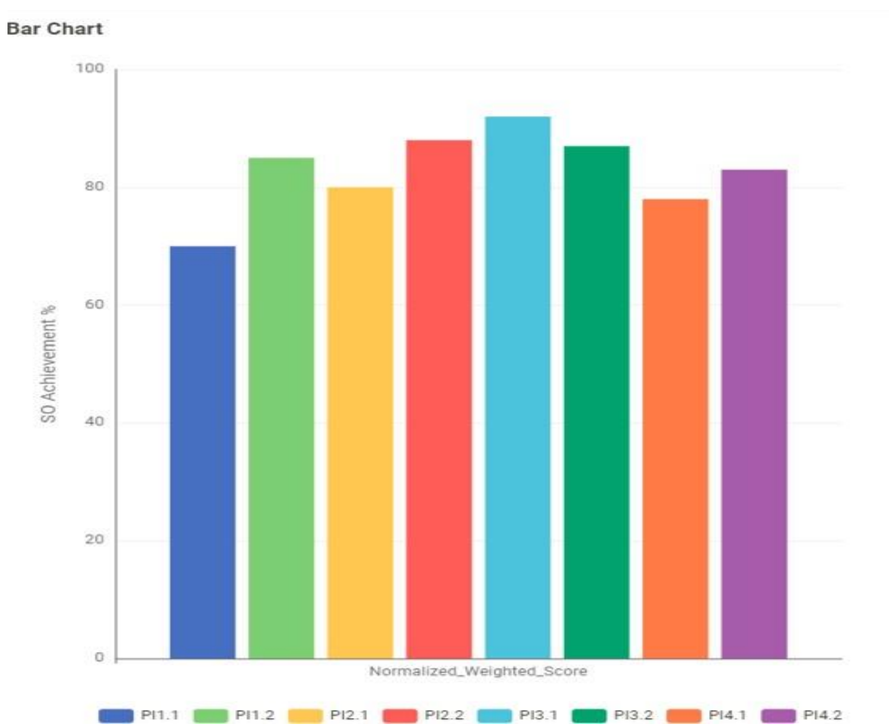


Fig.2 SOs Achievements Chart; it showcases SO achievement levels, highlighting areas of excellence and improvement needs.

The results demonstrate the transformative potential of AI tools in enhancing SO assessment and continuous improvement. The practical implications of this assessment example highlight the value of using AI tools for SO analysis, providing actionable insights for continuous improvement.

AIdriven system for Program Assessment and Continuous Improvement ensures systematic and unbiased decision-making. By incorporating these elements, the assessment framework effectively bridges the gap between AI technologies and traditional expert system principles, enhancing both accuracy and applicability.

4 Conclusion

The AI-driven SOA system represents a significant leap toward achieving educational excellence. By aligning assessments with predefined objectives, it promotes sustainable and data-driven improvements, equipping institutions to meet evolving student and stakeholder needs effectively. SOA bridges AI technology with traditional expert system principles, ensuring systematic, unbiased, and transparent assessments. It enables institutions to identify improvement areas, generate targeted action plans and continuously enhance academic programs.

SEE
invideo AI - Turn ideas into videos - AI video creator