Objective evaluation of performances in case of Students based on similarity analyses and Moodle-logs

Shagai Turtogtokh
Kodolányi János University and MY-X research team Hungary

László Pitlik
Kodolányi János University and MY-X research team Hungary

László Pitlik (Jr)
Kodolányi János University and MY-X research team Hungary

ABSTRACT

The increasing adoption of e-learning platforms has revolutionized educational practices and generated rich log data that can be harnessed to evaluate student performance objectively. This study proposes a comprehensive model that leverages 29 distinct attributes extracted from Moodle (e-learning platform) log data to provide a multifaceted/objective evaluation of student performance.

These attributes, which capture aspects such as diligence (e.g., posting frequency, active days, etc.), understanding (e.g., topic relevance, citation usage), and interaction dynamics (e.g., reply time, response length), are organized into an Object Attribute Matrix (OAM), where each attribute's type and direction (e.g. the less is the reply-time, the better is the performance, etc.) are defined.

To quantify topic alignment, the all-MiniLM-L6-v2 model is used to generate sentence embeddings, measuring semantic coherence between student responses and instructor posts via cosine similarity. Suspected AI-generated content is identified using the roberta-base-openai-detector, which assigns a 1–10 score (10 indicating high probability). The framework utilizes the COCO Y0 engine (https://miau.my-x.hu/myx-free/index_en.php3) — a computational analysis tool — to assess attribute impacts and rank students based on composite performance metrics.

This data-driven approach offers educators an objective framework for assessing student performance while delivering personalized feedback to enhance learning outcomes. Future research will validate the model's objectivity and effectiveness across diverse educational contexts.

Keywords: AI, anti-discrimination, optimization, automation

INTRODUCTION

The rapid growth of e-learning platforms has transformed how individuals acquire knowledge and develop skills, generating vast amounts of data that offer valuable insights into learning behaviors. According to Statista[i], the global e-learning market is projected to reach nearly $400 billion by 2026, doubling from $200 billion in 2019. This expansion reflects the increasing reliance on digital learning solutions and highlights the potential for data-driven insights to enhance educational outcomes. By examining behaviors such as engagement, consistency, collaboration, and comprehension, educators can personalize learning experiences, identify gaps in understanding, and assess the effectiveness of course design.

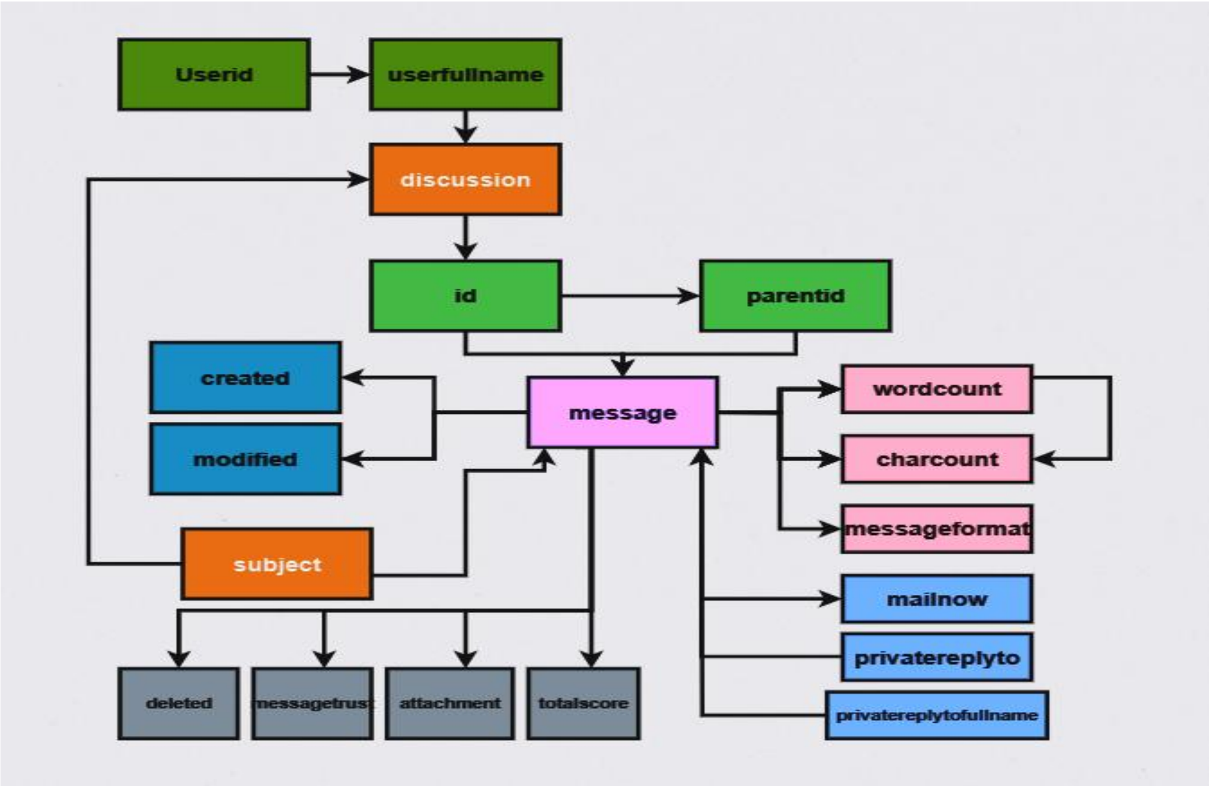
However, analyzing such data presents challenges. The large size and complexity of learning data makes it difficult to extract meaningful insights without advanced analytical methods. Additionally, concerns around privacy, data integrity, and algorithmic fairness require careful consideration to ensure evaluations are both accurate and ethical.

This research seeks to address these challenges by utilizing Moodle log data, advanced natural language processing (NLP) models, and an anti-discriminative engine to analyze student performance with 29 different attributes that reveal multiple dimensions of student activity. This approach aims to uncover meaningful insights into engagement patterns and learning behaviors, offering a data-driven framework for improving educational strategies and outcomes. As educational technology continues to evolve, such insights will play a vital role in shaping the future of learning, empowering educators to better support students in achieving their full potential.

MATERIALS AND METHOD

Moodle eLearning platform provided us valuable log data of the students' discussions and activities in the platform (c.f. Figure-1):~~.~~

Figure 1. The structure of the raw log data from the Moodle platform - (source: own presentation)~~.~~



Used raw log data https://miau.my-x.hu/miau/315/moodle/discussion.xlsx

With such complex raw data being provided, relevant attributes for each student can be extracted using queries and other technologies. We extracted 29 chosen attributes that are divided into two categories (24 for diligence and 5 for understanding):~~.~~

Diligence/ quantitative attributes:

1. total_posts - Total number of posts
2. active_days - Number of days that student actively participated in discussions
3. total_replies_to_professor - Number of replies to professor's post
4. total_characters - Total chars of all posts of the student
5. total_words - Total words of all posts of the student
6. avg_words - Average words per post
7. unique_interactions - The number of unique interactions with other students for each user
8. unique_discussions - the number of unique discussions for each user

9. engagement_rate - The attribute measures student engagement by dividing their replies by the professor's total posts, showing how actively they participate in the professor's discussions.
10. normanlized_score - The used query normalizes total replies and average reply time for all students, assigning a higher weight (70%) to replies and a lower weight (30%) to reply time. This weighting emphasizes the importance of engagement (quantity of replies) over responsiveness (speed of replies) when calculating an overall performance score. The normalized scores ensure fair comparisons regardless of differing scales for the two metrics.
11. deadline_exceeded(Quasi exam I) - Posts created after given deadline in quasi exam I
12. deadline_exceeded(Quasi exam II) - Posts created after given deadline in quasi exam II
13. deadline_exceeded(Quasi exam III) - Posts created after given deadline in quasi exam III
14. consistency_score - How consistently a student participates in discussions over time. standard deviation based on number of replies per week.
15. max_streak - Maximum number of activity streak
16. avg_reply_time – Average reply time to the professor
17. avg_charcount - Avarage char count of all post
18. max_charcount – The longest post char count
19. min_charcount -
20. modification_count - How many times student modified his post
21. avg_modified_time_minutes - How many minutes did it take to modify their post after posting
22. average_posts_per_day - Average number of posts per day
23. response_time_in_hours(Task-I) - response_time_in_hours(Task-I)
24. response_time_in_hours(Task-II) - Amount of time to answer Task II

Understanding/ qualitative attributes:

1. Pattern_followed(quasi exam i) - The number of posts where a student followed a specific pattern provided by the professor (with a maximum of 2 and a minimum of 0)
2. avg_AI_involvedMsg_score - The average AI involvement score is calculated using the "roberta-base-openai-detector" model, which detects AI-generated text. It assigns a score for each response (scaled from 1-10 the more the probability of AI involvement), and averaging these scores indicates the overall prevalence of AI-generated content.
3. topic_relevance_score - The Topic Relevance Score for every student's reply to the professor's post using a pre-trained NLP model (all-MiniLM-L6-v2) and cosine similarity.
4. citation_count - How often students use external references (e.g., links, citations) in their posts to measure research-oriented behavior.

5. valid_response - Quasi exam task I has the correct answer 37 and Task II has the correct answer 5. If the student answers the question correctly for both of the questions then 1 else 0

All these actionable insights are gathered through various queries, computations, and model interpretations. For instance, the query for attribute "deadline_exceeded(Quasi exam II)" is shown in Figure 2:.

Figure 2. Query for extracting the attribute "deadline_exceeded(Quasi exam II)"

```
SELECT
   userid, userfullname, COUNT(*) AS posts_after_deadline
FROM
   discussions_m
WHERE
   subject LIKE '%Quasi Exam II%' -- Filter for posts related to "Quasi Exam III"
   AND created > '2024-11-08 24:00:00' -- Filter posts made after the deadline
GROUP BY
   userid, userfullname
ORDER BY
   userfullname;
```

All other used queries: https://miau.my-x.hu/miau/315/moodle/QUERIES.docx

Most of the attributes that are about diligence can be extracted using queries and codes but we need to analyze text inputs to get the more detailed insights of the student. To get such insights following NLP models are utilized.

Topic relevance score: all-MiniLM-L6-v2

all-MiniLM-L6-v2 is a pre-trained language model specifically designed for generating high-quality sentence embeddings. This means it can convert sentences and paragraphs into numerical representations (vectors) that capture their semantic meaning. Additionally, its fast inference speed allows for scalable analysis of large volumes of student submissions.

Steps:

Importing the libraries:

```
import sqlite3
import pandas as pd
from sentence_transformers import SentenceTransformer
from sklearn.metrics.pairwise import cosine_similarity
```

Prepare the Data:

- Extract the professor's posts (parent IDs and messages).
- Extract student replies corresponding to each parent post (message where parent = prof_id).

```
prof_query = "SELECT id AS prof_id, message AS prof_message FROM discussions_m
WHERE userid = 34004"  # Professor's posts
reply_query = "SELECT id, parent, userid, userfullname, message AS
student_message FROM discussions_m WHERE parent IN (SELECT id FROM
discussions_m WHERE userid = 34004)"  # Student replies

prof_posts = pd.read_sql_query(prof_query, conn)
student_replies = pd.read_sql_query(reply_query, conn)
```

Embed the Messages:

- Use a pre-trained sentence embedding model (e.g., SentenceTransformer).
- Generate embeddings for both professor's posts and student replies.

```
prof_posts['embedding'] = prof_posts['prof_message'].apply(lambda x:
model.encode(x) if isinstance(x, str) and x.strip() else None)

# Filter out replies with invalid parent IDs
student_replies =
student_replies[student_replies['parent'].isin(prof_posts['prof_id'])]
```

Compute Similarity:

- For each student reply, calculate the cosine similarity between the reply's embedding and the professor's post's embedding.

```
student_replies['topic_relevance_score'] = student_replies.apply(
    lambda x: calculate_similarity(x, prof_posts) if x['student_message'] else
None, axis=1
)
```

Store and Output Scores:

- Save the relevance score for each student reply in the database or output the results as a report.

```
merged_df[['id', 'topic_relevance_score']].to_sql('relevance_scores', conn,
if_exists='replace', index=False)
conn.close()
```

Full source code:https://miau.my-x.hu/miau/315/moodle/Topic%20relevance%20score(code).docx

Predicted AI-generated text: roberta-base-openai-detector

Nowadays, lots and lots of educational institutions use AI detector tools for their system so including this metric in this learning assessment is a must. Therefore, the roberta-base-openai-detector model is used for student AI text prediction due to its strong contextual understanding, fine-tuning for AI-generated content detection, and reduced false positive rates.

Steps:

1. Prepare the Environment – getting all important Python libraries. Load and Preprocess Data.
2. Function to clean text (Remove punctuation, Normalize whitespace)
3. Analyze Text Complexity (Calculate readability metrics)
4. Predict AI probability
5. Combine Attributes into a Unified Score
6. Scale to a 1-10 rating
7. Pull the result into the database

Full source code: https://miau.my-x.hu/miau/315/moodle/(Code)Detecting%20AI%20generated%20text%20using%20pretrained%20model.docx

With all 29 attributes extracted using queries and model interpretations, we can create an Object attribute matrix (OAM, c.f. Figure-3):~~.~~

Figure 3. OAM (24 objects, 29 attributes) - (source: own presentation)

	direction	0	0	0	0	1	1	0	
	type	x	x	x	x	x	x	x	
	Value	integer	integer	integer	integer	integer	integer	decimal	
	attribute_id	A1	A2	A3	A4	A27	A28	A29	
userid	username/attribute_name	totalPosts	activeDays	total_replies_to_prof	total_characters	response_time_in_hours(Task-I)	response_time_in_hours(Task-II)	average_posts_per_day	Y
1	student_1	16	3	0	8697	0	0	5.3	1000
2	student_2	16	5	13	5493	34	34	3.2	1000
3	student_3	10	3	8	1841	32	32	3.3	1000
4	student_4	22	8	19	14790	12	16	2.8	1000
5	student_5	8	4	7	3666	15	0	2.0	1000
6	student_6	37	10	23	10778	14	14	3.7	1000
7	student_7	14	6	11	3696	12	12	2.3	1000
8	student_8	18	5	15	7140	14	15	3.6	1000
9	student_9	11	4	8	3015	15	16	2.8	1000
10	student_10	17	5	10	2191	13	13	3.4	1000
11	student_11	3	1	2	383	0	0	3.0	1000
12	student_12	14	7	9	3422	0	0	2.0	1000
13	student_13	2	1	2	1452	0	0	2.0	1000
14	student_14	3	3	2	1308	0	0	1.0	1000
15	student_15	27	12	15	5429	34	35	2.3	1000
16	student_16	17	6	14	8260	10	15	2.8	1000
17	student_17	21	8	17	6484	10	16	2.6	1000
18	student_18	12	4	9	2852	16	13	3.0	1000
19	student_19	16	6	11	7209	18	18	2.7	1000
20	student_20	20	6	17	7590	11	15	3.3	1000
21	student_21	17	6	14	4696	14	15	2.8	1000
22	student_22	20	4	17	9436	7	8	5.0	1000
23	student_23	4	3	1	414	0	0	1.3	1000
24	student_24	1	1	1	51	0	0	1.0	1000

Reference: https://miau.my-x.hu/miau/315/moodle/OAM_student_analysis_Moodle.xlsx

OAM rules:

Each of the 29 attributes in our model is assigned a directional value to indicate its impact on performance. A direction value of 0 signifies that more is better, while a value of 1 indicates that less is better. For example, attributes such as the number of posts or replies to the professor are positively correlated with better performance, so they are assigned a direction of 0. Conversely, the AI involvement score, which reflects the likelihood of machine-generated responses, negatively impacts performance and is thus assigned a direction of 1.

To standardize performance measurement, each student begins with a baseline performance index of 1000. This starting point ensures that performance scores can increase or decrease without introducing negative values, which the linear programming tool we employed is less suited to handle. While higher baseline values, such as 10,000 or 1,000,000, could offer even finer granularity, the choice of 1000 provided sufficient precision for our analysis while maintaining clear, interpretable results.

This structured scoring system enables the creation of a ranked table using Excel Rank() function derived from the Object Attribute Matrix (OAM), which is then processed using the COCO Y0 anti-discriminative model to assess students individually and compare student performances (c.f. Figure-4):~~.~~

Figure 4. Ranked values compared to a performance index: 1000 (norm-value) - (source: own presentation)

userid	direction	0	0	0	0	1	1	0	Y
	type	x	x	x	x	x	x	x	
	Value	integer	integer	integer	integer	integer	integer	decimal	
	attribute_id	A1	A2	A3	A4	A27	A28	A29	
	username/attribute_name	totalPosts	activeDays	total_replies_to_prof	total_chara	response_time_in_hours(Task-I)	response_time_in_hours(Task-II)	average_posts_per_day	
1	student_1	11	18	24	4	1	1	1	1000
2	student_2	11	11	10	10	23	23	8	1000
3	student_3	18	18	16	19	22	22	6	1000
4	student_4	3	3	2	1	12	18	13	1000
5	student_5	19	14	18	14	18	1	19	1000
6	student_6	1	2	1	2	15	13	3	1000
7	student_7	14	6	11	13	12	10	17	1000
8	student_8	7	11	6	8	15	14	4	1000
9	student_9	17	14	16	16	18	18	13	1000
10	student_10	8	11	13	18	14	11	5	1000
11	student_11	21	22	19	23	1	1	9	1000
12	student_12	14	5	14	15	1	1	19	1000
13	student_13	23	22	19	20	1	1	19	1000
14	student_14	21	18	19	21	1	1	23	1000
15	student_15	2	1	6	11	23	24	18	1000
16	student_16	8	6	8	5	9	14	11	1000
17	student_17	4	3	3	9	9	18	16	1000
18	student_18	16	14	14	17	20	11	9	1000
19	student_19	11	6	11	7	21	21	15	1000
20	student_20	5	6	3	6	11	14	6	1000
21	student_21	8	6	8	12	15	14	11	1000
22	student_22	5	14	3	3	8	9	2	1000
23	student_23	20	18	22	22	1	1	22	1000
24	student_24	24	22	22	24	1	1	23	1000

Reference: https://miau.my-x.hu/miau/315/moodle/OAM_student_analysis_Moodle.xlsx

COCO Y0 Engine (https://miau.my-x.hu/myx-free/index_en.php3):

The COCO Y0 engine is a computational tool designed to evaluate and compare objects based on multiple attributes while ensuring objective and anti-discriminative analysis. By leveraging data organized in an Object Attribute Matrix (OAM), the engine calculates performance indices through a set of predefined conditions to maintain fairness and consistency.

In addition to the direction vector (0,1) and integer-based Y-values, which have already been discussed, the COCO Y0 engine operates under the following key principles[ii]:

- No Preset Weight Relations: The model does not require predefined relationships between attribute weights or fixed intervals between performance levels (referred to as "stairs"), ensuring unbiased calculations.
- Additive Logic: If an attribute has an extreme value (e.g., zero), the corresponding Y-value will not be zero. Instead, the system maintains additive logic, where each attribute contributes proportionally to the overall score.
- Column Filtering: Identical attribute columns are excluded to improve computational efficiency without compromising the overall evaluation.
- Genetic Potential Interpretation: The sum of the initial values in each attribute column can be interpreted as a form of 'potential,' ensuring all evaluated entities start on an equal footing.

Adhering to these principles makes the COCO Y0 engine suitable for analyzing complex data structures like student performance metrics (c.f. Figure-5):~~.~~

Figure 5. Error-free estimations (source: own presentation)

userid	username	Estimation	Fact+0	Delta	Delta/Fact
1	student_1	1000	1000	0	0
2	student_2	1001	1000	-1	-0.1
3	student_3	999	1000	1	0.1
4	student_4	1001	1000	-1	-0.1
5	student_5	1000	1000	0	0
6	student_6	1000	1000	0	0
7	student_7	999	1000	1	0.1
8	student_8	1001	1000	-1	-0.1
9	student_9	1000	1000	0	0
10	student_10	999	1000	1	0.1
11	student_11	1000	1000	0	0
12	student_12	999	1000	1	0.1
13	student_13	1001	1000	-1	-0.1
14	student_14	1000	1000	0	0
15	student_15	1000	1000	0	0
16	student_16	1001	1000	-1	-0.1
17	student_17	1000	1000	0	0
18	student_18	999	1000	1	0.1
19	student_19	1001	1000	-1	-0.1
20	student_20	1001	1000	-1	-0.1
21	student_21	1000	1000	0	0
22	student_22	1001	1000	-1	-0.1
23	student_23	1000	1000	0	0
24	student_24	1000	1000	0	0

Reference: https://miau.my-x.hu/miau/315/moodle/OAM_student_analysis_Moodle.xlsx

In Figure 5, the COCO Y0 engine produces results where all evaluated objects receive the same score, often matching the pre-defined baseline value (e.g., 1000 points). This outcome suggests that the model's anti-discriminative logic has functioned effectively to ensure fairness, but it also indicates that some critical differences between objects remain obscured. In such cases, attribute exclusion becomes necessary to refine the evaluation.

The model's anti-discriminative logic may hide certain attributes that had minimal or no influence in the initial run. These are often attributes that contribute evenly across all objects. The next step is to intentionally remove the attributes that were previously dominant in the initial evaluation. By excluding those attributes, the focus shifts to the previously obscured characteristics, which often carry the subtle yet crucial differences between objects.

To do that, In Figure 6 we can see the stairs2 table and the s1 row values starting from 23 in certain attributes are the obscured attributes (c.f. Figure-6):~~.~~

Figure 6. COCO result (stairs2 table hidden attributes) (source: own presentation)

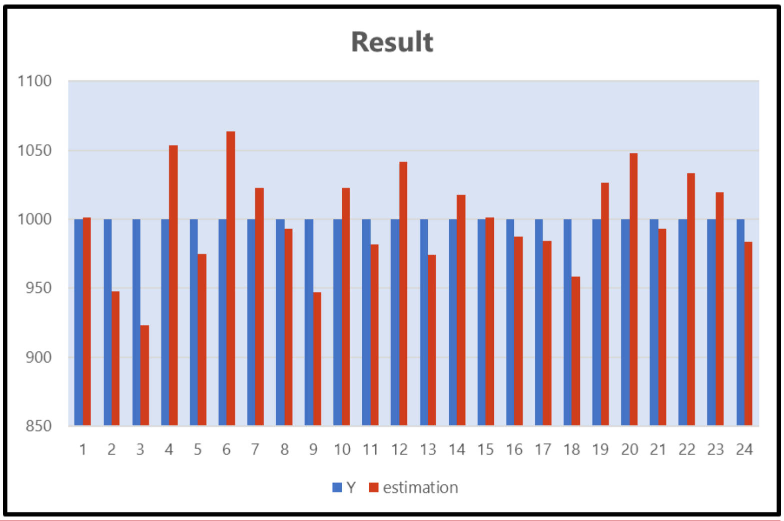
stairs(2)	X(A1)	X(A2)	X(A3)	X(A4)	X(A5)	X(A6)	A23)	X(A24)	X(A25)	X(A26)	X(A27)	X(A28)	X(A29)
S1	28	98	23	23	137	182	23	23	23	268	23	23	50
S2	27	56	22	22	136	181	22	22	22	267	22	22	49
S3	26	55	21	21	135	180	21	21	21	266	21	21	48
S4	25	54	20	20	134	179	20	20	20	172	20	20	47
S5	24	53	19	19	133	178	19	19	19	171	19	19	46
S6	23	18	18	18	132	177	18	18	18	170	18	18	45
S7	22	17	17	17	131	176	17	17	17	31	17	17	17
S8	16	16	16	16	130	175	16	16	16	30	16	16	16
S9	15	15	15	15	129	151	15	15	15	29	15	15	15
S10	14	14	14	14	128	150	14	14	14	28	14	14	14
S11	13	13	13	13	127	149	13	13	13	13	13	13	13
S12	12	12	12	12	126	12	12	12	12	12	12	12	12
S13	11	11	11	11	125	11	11	11	11	11	11	11	11
S14	10	10	10	10	124	10	10	10	10	10	10	10	10
S15	9	9	9	9	123	9	9	9	9	9	9	9	9
S16	8	8	8	8	122	8	8	8	8	8	8	8	8
S17	7	7	7	7	121	7	7	7	7	7	7	7	7
S18	6	6	6	6	120	6	6	6	6	6	6	6	6
S19	5	5	5	5	5	5	5	5	5	5	5	5	5
S20	4	4	4	4	4	4	4	4	4	4	4	4	4
S21	3	3	3	3	3	3	3	3	3	3	3	3	3
S22	2	2	2	2	2	2	2	2	2	2	2	2	2
S23	1	1	1	1	1	1	1	1	1	1	1	1	1
S24	0	0	0	0	0	0	0	0	0	0	0	0	0

Reference: https://miau.my-x.hu/miau/315/moodle/OAM_student_analysis_Moodle.xlsx

Once we exclude the attributes, we run the COCO engine for the second time and get the refined results as shown in Figure 7:~~.~~

Figure 7-8. Result and rankings after attribute exclusion (source: own presentation)~~.~~

userid	username	Y	estimation	rank
1	student_1	1000	1001.3	11
2	student_2	1000	947.7	22
3	student_3	1000	923.2	24
4	student_4	1000	1053.4	2
5	student_5	1000	974.8	19
6	student_6	1000	1063.4	1
7	student_7	1000	1022.8	7
8	student_8	1000	993.3	13
9	student_9	1000	947.2	23
10	student_10	1000	1022.8	7
11	student_11	1000	981.8	18
12	student_12	1000	1041.3	4
13	student_13	1000	974.3	20
14	student_14	1000	1017.8	10
15	student_15	1000	1001.3	11
16	student_16	1000	987.3	15
17	student_17	1000	984.3	16
18	student_18	1000	958.7	21
19	student_19	1000	1026.3	6
20	student_20	1000	1047.9	3
21	student_21	1000	993.3	13
22	student_22	1000	1033.3	5
23	student_23	1000	1019.3	9
24	student_24	1000	983.3	17



Reference (Figure-7): https://miau.my-x.hu/miau/315/moodle/OAM_student_analysis_Moodle.xlsx

Reference (Figure-8): https://miau.my-x.hu/miau/315/moodle/OAM_student_analysis_Moodle.xlsx

Validation of Results

To ensure the reliability and accuracy of our evaluation results, we applied a validation process based on symmetry effects. This method verifies whether the differences in attribute values between students align consistently with their performance rankings, reinforcing the model's predictive reliability.

The validation process:

Reverse Ranking: We first reverse the original ranking order using the formula:

$$Reversed\ rank = Number\ of\ Objects - Original\ Rank + 1$$

Re-Evaluation with COCO Engine: After creating this flipped ranking table, we ran the COCO Y0 engine on the adjusted data.

Delta Calculation: We compute a key metric known as the product of the original delta values and the inverted delta values. This metric acts as a critical indicator of the model's consistency.

$$1st\frac{delta}{fact} * Inverted\frac{delta}{fact}$$

The result is interpreted according to the following rule:

If the product of the two delta values is zero or less, the model's results are confirmed to be valid and reliable. If the product is greater than zero, it indicates potential inconsistencies. This could signal errors in certain students' data or weaknesses in the model itself.

With such a validation process we improve our confidence in the model's accuracy and ensure that the ranking outcomes reflect meaningful distinctions in student performance therefore in Figure 9 we can see all objects are valid.

Expected Outcome for Valid Objects: Direct-ranking vs. Inverse-ranking should produce inverted results with differences centered around the norm value 1000 (c.f. Figure-9):~~.~~

Figure 9. Validation results (source: own presentation)

Important error: standard row-headers must always be given in case of all OAMs!

Example: (userid = O(1)-O(n) = student1-studentn, etc.)!!!

userid	Y	estimation	validation	rank
00001	1000	1001.3	1	11
00002	1000	947.7	1	22
00003	1000	923.2	1	24
00004	1000	1053.4	1	2
00005	1000	974.8	1	19
00006	1000	1063.4	1	1
00007	1000	1022.8	1	7
00008	1000	993.3	1	13
00009	1000	947.2	1	23
00010	1000	1022.8	1	7
00011	1000	981.8	1	18
00012	1000	1041.3	1	4
00013	1000	974.3	1	20
00014	1000	1017.8	1	10
00015	1000	1001.3	1	11
00016	1000	987.3	1	15
00017	1000	984.3	1	16
00018	1000	958.7	1	21
00019	1000	1026.3	1	6
00020	1000	1047.9	1	3
00021	1000	993.3	1	13
00022	1000	1033.3	1	5
00023	1000	1019.3	1	9
00024	1000	983.3	1	17

Reference: https://miau.my-x.hu/miau/315/moodle/OAM_student_analysis_Moodle.xlsx

After symmetry effect analysis we verified the reliability of the COCO system by giving wrong inconsistent too high or low inputs since it is the key tool for the evaluation (c.f. Figure-10-11-12-13):~~-~~

Figure 10. Incorrect inputs: too high values~~-~~(source: own presentation)

totalPosts	activeDays	total_replie	total_chara	total_words	avg_words	....
200	0	2222222	8697	1473	92.06	....
16	0	13	5493	234234234	78.94	....
10	0	8	1841	352	47.20	....
22	0	19	14790	2432	121.00	....
8	0	7	3666	528	67.88	....
10	10	10	10	10	10	....
0	0	11	3696	610	64.50	....
18	0	15	7140	1163	102.00	....
11	0	8	3015	433	62.36	....
222323232	0	10	2191	352	38.65	....
3	0	222222	383	74	26.33	....
14	0	9	3422	518	61.29	....
2	0	2	1452	275	137.50	....
3	0	2	1308	118	50.67	....
2324342	0	15	5429	911	60.81	....
17	0	14	8260	1373	99.47	....
21	0	17	6484	1216	71.71	....
4432323	0	9	2852	364	41.67	....
16	0	11	7209	1123	141.44	....
10000000	10000000	10000000	10000000	10000000	10000000	....
17	0	14	4696	758	55.76	....
20	0	17	9436	1551	94.85	....
23232333	0	1	414	84	59.50	....
1	0	1	51	12	12.00	....

Between the figures, it is necessary to have explanations!

Important errors:

column-headers must always be visible/readable AND

standardized in case of ALL attributes in all figures! (Units for all attributes must always be given!

Row-headers must always be given!

Figure 11. outputs of Figure 10. (source: own presentations)

userid	userfullname	Y	estimation	validation	rank
47141	Aadi Rajesh	1000	1002.5	1	5
46681	Amgalanbaatar Am	1000	958.4	1	24
46668	Amin-Erdene Ankhb	1000	1002.5	1	5
46671	Ariunbold Munkhjar	1000	1002.5	1	5
46666	Battuguldur Tuyatse	1000	1002.5	1	5
47139	Benjámin Honti	1000	1046.6	1	2
46683	Bilegt Gankhuyag	1000	1002.5	1	5
46674	Boldsukh Ganzorig	1000	973.9	1	22
46667	Dulguun Sukh-Ochi	1000	1002.5	1	5
46677	Ganbat Bayanmunk	1000	1051.1	1	1
45297	Gábor Kosdi	1000	1013	1	4
48825	Gülsah Öztürk	1000	1002.5	1	5
44166	István Siposs	1000	992	1	18
42461	Japheth Jerry Dangi	1000	1002.5	1	5
44444	Muhammad Khuran	1000	1002.5	1	5
46672	Munkh-Orgil Batbay	1000	1002.5	1	5
45293	Márk Zsigmond Léva	1000	1002.5	1	5
46673	Namjiljav Tsetsegsu	1000	981.5	1	20
46678	Nurbol Byekbolat	1000	970.4	1	23
46675	Shagai Turtogtokh	1000	1002.5	1	5
46682	Yaruu-Aldar Enkhtu	1000	975.4	1	21
46680	Zandangarav Nyam	1000	1002.5	1	5
45764	Zoltán Lehrer	1000	1021.6	1	3
45304	Zoltán Sváb	1000	983.5	1	19

We do never need names!

In Figure 11, the result is not consistent and does not deliver a clean output due to too high incorrect values being given as inputs.

Figure 12. Incorrect inputs: one person gets too high constant value (source: own presentation)

Standardized row-headers!!! (e.g. Student(1)-Student(n))

totalPosts	activeDays	total_replie	total_chara	total_words	avg_words	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
10000000	10000000	10000000	10000000	10000000	10000000	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....
0	0	0	0	0	0	....

Figure 13. Outputs of Figure 12. (source: own presentation)

userid	userfullname	Y	estimation	validation	rank
47141	Aadi Rajesh	1000	1000	1	1
46681	Amgalanbaatar Am	1000	1000	1	1
46668	Amin-Erdene Ankhb	1000	1000	1	1
46671	Ariunbold Munkhjar	1000	1000	1	1
46666	Battuguldur Tuyatse	1000	1000	1	1
47139	Benjámin Honti	1000	1000	1	1
46683	Bilegt Gankhuyag	1000	1000	1	1
46674	Boldsukh Ganzorig	1000	1000	1	1
46667	Dulguun Sukh-Ochi	1000	1000	1	1
46677	Ganbat Bayanmunk	1000	1000	1	1
45297	Gábor Kosdi	1000	1000	1	1
48825	Gülsah Öztürk	1000	1000	1	1
44166	István Siposs	1000	1000	1	1
42461	Japheth Jerry Dangi	1000	1000	1	1
44444	Muhammad Khuran	1000	1000	1	1
46672	Munkh-Orgil Batbay	1000	1000	1	1
45293	Márk Zsigmond Léva	1000	1000	1	1
46673	Namjiljav Tsetsegsu	1000	1000	1	1
46678	Nurbol Byekbolat	1000	1000	1	1
46675	Shagai Turtogtokh	1000	1000	1	1
46682	Yaruu-Aldar Enkhtu	1000	1000	1	1
46680	Zandangarav Nyam	1000	1000	1	1
45764	Zoltán Lehrer	1000	1000	1	1
45304	Zoltán Sváb	1000	1000	1	1

In Figure 13, all the results are identical due to incorrect inputs that one person gets too high a constant value.

In Figures 10, 11, 12, and 13, we can see that inconsistent low-quality data inputs lead to poor results from the COCO module. However, the potential of real-world/correct data for insightful output is evident in Figures 8 and 9.

RESULTS (=summary) ~~AND~~ DISCUSSION (=weakest elements in the entire own logic?)

In this study, we created OAM consisting of 24 objects and 29 attributes that are categorized into two categories Diligence, and Understanding. In the first run on the COCO Y0 engine, all the students got identical performance scores indicating the anti-discriminative logic of the engine, which ensures fairness by avoiding preset weight relations in Figure 5. However, this uniformity hid some attributes and differences in how well things performed. To get the hidden results, attribute exclusion was performed based on the stairs2 table (Figure 6), which identified attributes with minimal differentiation impact (e.g., those with s1 row values starting at 23) and removed the attributes that were previously dominant in the initial evaluation. After attribute exclusion, we ran the COCO Y0 engine a second time. This gave us a clearer and more detailed ranking of the students, which is demonstrated in (Figure 7). Figure 8 visualizes the performance distribution of students, highlighting the variability in student performance across the group. A symmetry effect analysis was conducted to validate these results. The original ranking was reversed (Reversed Rank = 24 - Original Rank + 1), and the COCO Y0 engine re-evaluated the inverted data. The product of the original and inverted delta values was calculated for each student, with all values satisfying the validity condition (product ≤ 0), as shown in Figure 9. This confirmed the consistency and reliability of the rankings. To further assess the COCO Y0 engine's reliability sensitivity tests were tested. Incorrect inputs, such as excessively high values (Figures 10 and 11) or constant high values for one student (Figures 12 and 13), were given. In Figure 11, the output was erratic and inconsistent due to the exaggerated inputs, while in Figure 13, all scores converged to identical values, reflecting the model's response to uniform distortion. These tests demonstrated that the engine appropriately handles poor-quality data, supporting confidence in its performance with accurate inputs. Once all validation processes are complete, we want to see how objective and subjective evaluations differ. To get the subjective evaluations we asked students to rate their peers' performance and engagement on a scale of 0 to 9. In Figure 14, the results of the pure subjective evaluation can be seen, and the correlation between Figure 14 and 7 is -0.25—a negative value that tells us these two approaches don't align. This lack of correlation underscores how subjective perceptions can differ sharply from objective data-driven results~~.~~ (c.f. Figure-14):

Figure 14. Subjective evaluation (Result of the vote within the classroom) (source: own presentation)

userid	username	total_score	rank
00001	Student_1	128	9
00002	Student_2	129	5
00003	Student_3	123	10
00004	Student_4	123	11
00005	Student_5	25	21
00006	Student_6	123	16
00007	Student_7	21	20
00008	Student_8	18	23
00009	Student_9	135	3
00010	Student_10	119	15
00011	Student_11	135	2
00012	Student_12	127	7
00013	Student_13	123	12
00014	Student_14	17	24
00015	Student_15	22	22
00016	Student_16	27	18
00017	Student_17	137	4
00018	Student_18	122	13
00019	Student_19	104	19
00020	Student_20	44	17
00021	Student_21	129	6
00022	Student_22	122	8
00023	Student_23	138	1
00024	Student_24	121	14

Even though the model tries to be objective, it has some limitations we need to consider. The researchers had to choose which features to focus on and decide whether higher or lower values were better (represented as 0 or 1). This introduces some subjectivity. For instance, if the model values how many replies a student makes more than how quickly they reply, it might not give a fair picture of their performance. Also, the accuracy of the text analysis (NLP) depends on how clear and consistent the writing is in the Moodle discussions. If students use informal language or give incomplete answers, the results could be inaccurate. The sensitivity tests (Figures 10–13) show that if the data isn't good, the model's results will be unreliable. There are also ethical concerns about using AI to detect AI use. We need to be careful not to unfairly punish students who are using AI tools in acceptable ways. Clear rules are needed. Finally, the model was only tested on 24 students, which is a small group. This means we can't be sure if it will work well for all students in different educational settings. We need to test it on a larger and more diverse group to see if it's truly reliable. Despite these challenges, the model advances educational assessment by offering a data-driven, anti-discriminative tool that balances quantitative and qualitative dimensions.

CONCLUSION (reactions concerning the discussed layers)

This model offers a more equitable and precise approach to evaluating online learning, though further research across diverse settings and ethical considerations are essential for its widespread and responsible application.

Summary!!!

REFERENCES

Each URL should be listed here: e.g.

Author1, year1, title1, url1, apropo(journal,conference,etc.)1, page1, ...
Author2, year2, title2, url2, apropo(journal,conference,etc.)2, page2, ...
ABC-ordered

[i] https://www.statista.com/statistics/1130331/e-learning-market-size-segment-worldwide/
[ii] https://miau.my-x.hu/myx-free/index_e.php3?x=e01