Francisco Jose De Caldas District University
Faculty of Engineering

Steven Navarro
20221020048

Juan David Amaya Patiño
20221020057

David Santiago Garcia Galeano
20231020158

Eng. Carlos Andrés Sierra
SYSTEMS ANALYSIS AND DESIGN
Group: 020-82

Bogotá
2025

Competition link:
https://www.kaggle.com/competitions/eedi-mining-misconceptions-in-mathematics/overview

Competition Overview: Predicting the affinity between misconceptions and incorrect answers (distractors) in multiple-choice questions.

Summary: This analysis addressed the system that comprises the Kaggle competition Eedi - Mining Misconceptions in Mathematics from a systems theory perspective. This competition seeks to develop machine learning models capable of identifying misconceptions in student responses to multiple-choice mathematical questions. Through the study of the system's elements—such as data, evaluation metrics, rules, actors involved, and incentives—a complex but coherent structure was revealed, aimed at improving teaching-learning processes through artificial intelligence. One of the main findings is the efficient automation of misconceptions labeling, which addresses an important pedagogical need and how metrics such as MAP@25 and computational constraints not only guide the development of effective solutions, but also applicable in real-world contexts. Together, this system represents an effective synergy between technology, education, and data science to address a high-impact educational problem.

Systems Analysis Report

1. **Key components of this competence:**
   1.1. **Problem description:**
      1.1.1. **Intent of the competition:** The competition seeks to improve the labeling of misconceptions in student responses using a Natural Language Processing (NLP) model.

      1.1.2. **What it's about:** A Machine Learning (ML) model must be developed that predicts the relationship between distractors (incorrect answers) and underlying misconceptions in mathematical problems.

      1.1.3. **Why the problem needs to be solved:**
      - Manual labeling is slow and inconsistent across different people.
      - Student errors can be very varied and difficult to classify.
      - Existing models have not performed well due to the complexity of the mathematical content.
      - A more effective and scalable system is needed to improve the process.

      1.1.4. **Aim:** Develop an NLP model that automates and optimizes the labeling process, allowing for accurate classification of known misconceptions and generalization to new, emerging concepts. This will improve mathematics teaching and learning.

   1.2. **Dataset:**
      1.2.1. **Description:** In Eedi, students answer Diagnostic Questions (DQs), which are multiple-choice questions with one correct answer and three incorrect answers (distractors). Each question is associated with a specific construct (skill or concept being assessed). Distractors represent common errors students may make, that is to say, misconceptions.

1.2.2. Example Diagnostic Question: Each answer choice is labeled with a misconception (except the correct answer):

- A: Error in incorrect division in a proportion.
- B: Confusion about the order of the terms in the proportion.
- C: Incomplete calculation of one part of the proportion.
- D: Correct answer.

1.2.3. Structure of Data Files:

1.2.3.1. train.csv and test.csv: They contain the questions and answers, along with truth labels in train.csv to train the model.

- QuestionId: Unique identifier of the question.
- ConstructId: Unique ID of the construct being assessed.
- ConstructName: Name of the concept assessed in the question.
- CorrectAnswer: Indicates the correct answer (A, B, C, or D).
- SubjectId: Unique identifier of the general topic.
- SubjectName: Broader context of the construct (e.g., Mathematics).
- QuestionText: Question text, extracted using OCR.
- Answer[A/B/C/D]Text: Text of each answer option.
- Misconception[A/B/C/D]Id: ID of the misconception associated with each distractor.

1.2.3.2. misconception_mapping.csv:

- Relates MisconceptionId to MisconceptionName, providing a description of the associated error.

1.2.3.3. sample_submission.csv:

- It shows the expected submission format, where each incorrect answer must be associated with up to 25 predicted misconceptions, delimited by spaces.

1.3. Target variable:

1.3.1. Description: The target variable in this competition is the association between distractors and misconceptions, represented by the column Misconception[A/B/C/D]Id in the training set (train.csv). For each diagnostic question, incorrect answers (A, B, C) are labeled with a MisconceptionId, which indicates the associated misconception.

1.3.2. Objective of the Model: The model must predict the correct MisconceptionId for each distractor in the test set (test.csv). Since distractors can be related to multiple misconceptions, the prediction can include up to 25 possible MisconceptionIds, separated by spaces.

1.3.3. Example:

QuestionId	Answer	MisconceptionId
1001	A	5
1001	B	12 23
1001	C	8 19 21

1.4. Evaluation metric:

1.4.1. Evaluation: Submissions are evaluated according to the Mean Average Precision @ 25 (MAP@25):

$$\text{MAP@25} = \frac{1}{U} \sum_{u=1}^{U} \sum_{k=1}^{min(n,25)} P(k) \times rel(k)$$

where ***U*** is the number of observations, ***P(k)*** is the precision at cutoff ***k***, ***n*** is the number predictions submitted per observation, and ***rel(k)*** is an indicator function equaling 1 if the item at rank ***k*** is a relevant (correct) label, zero otherwise.
Once a correct label has been scored for an observation, that label is no longer considered relevant for that observation, and additional predictions of that label are skipped in the calculation. For example, if the correct label is A for an observation, the following predictions all score an average precision of 1.0.

1.4.2. **Submission File:** For each QuestionId_Answer row in the test set, you must predict the corresponding MisconceptionId. You can predict up to 25 MisconceptionId values per row, and these should be space-delimited. The file should contain a header and have the following format:

```
QuestionId_Answer,MisconceptionId
1869_A,1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
1869_B,1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
1869_C,1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
1870_B,1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
...
```

1.5. Rules and restrictions

1.5.1. This competition is a Code Competition, which means submissions must be made through Notebooks on Kaggle.

1.5.2. Maximum execution time:

- If the notebook uses only the CPU, the execution time should not exceed 9 hours.
- If the notebook uses a GPU, the execution time should also not exceed 9 hours.

- This means that the code must be efficient and execute within this time limit.

1.5.3. **Internet access disabled:**
- Data cannot be downloaded at runtime.
- All datasets or pre-trained models must be uploaded first or come from sources approved by Kaggle.

1.5.4. **Use of external data:**
- The use of external data is permitted as long as it is free and publicly available.
- Pre-trained models that meet this requirement may also be used.

1.5.5. **Submission file format:**
- The submission file must be named submission.csv.
- If the file has a different name, the platform will reject it.

1.6. **Awards and incentives:**

1.6.1. **Prizes based on leaderboard ranking:** These prizes are awarded to the best-performing participants on the final leaderboard.

- 🥇 1er lugar: $12,000
- 🥈 2do lugar: $8,000
- 🥉 3er lugar: $5,000
- 4to lugar: $5,000

1.6.2. **Efficiency awards:** These incentives are awarded to the most efficient models, which may include factors such as speed, resource consumption, or model generalization.

- 🏆 1er lugar: $12,000
- 🏆 2do lugar: $8,000
- 🏆 3er lugar: $5,000

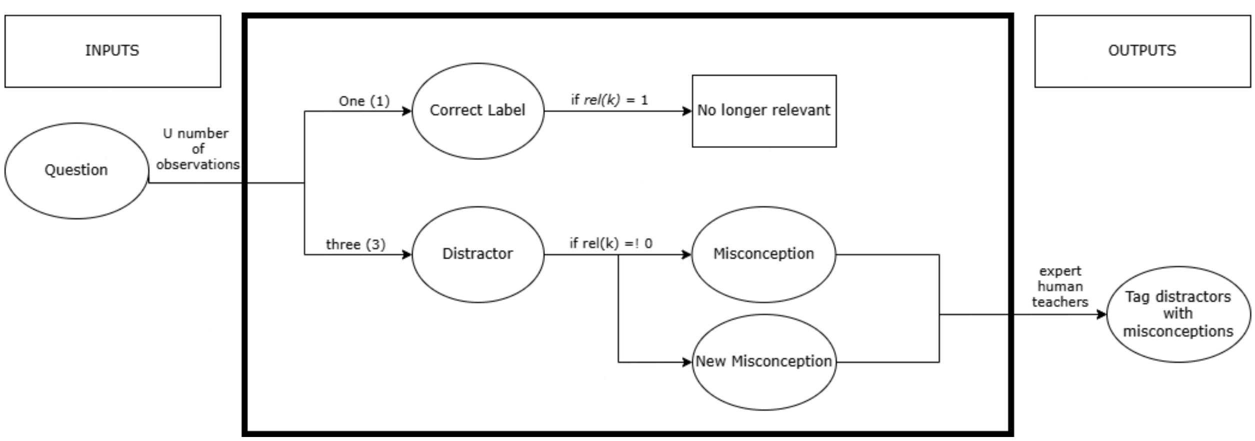
2. **System relations:**

2.1. **Seen relationships:** in the system market in *italics* and **bold** which connect the elements in the system

- Questions → ***U (number of observations)*** → Correct Label or Distractors
- Correct Label → ***if rel(k) = 1***→ No longer considered relevant (skip)
- Distractors → ***if rel(k) =! 1***→ Misconception or New Misconceptions
- Misconception → ***expert human teachers*** → Tag distractors with misconceptions
- New Misconceptions → ***expert human teachers*** → Tag distractors with misconceptions

The box boundaries for this system are all the operations or calculus which are not part of the Mean Average Precision @ 25 (MAP@25):

The diagram is a representation of the system relationships, adding which are the inputs and outputs for the system.



System Relationships

3. **Principles of systems engineering applied to the system:**

3.1. **Stakeholders:**

- Expert human teachers need an efficient way to tag misconceptions.

- Possible Students who answer the questions are indirect beneficiaries, better misconception detection will improve their learning.

3.2. Requirements and Requirements Management:

- Model should predict affinity between misconceptions and distractors.
- It should generalize to new misconceptions.
- Should run efficiently under computational constraints (CPU-only for efficiency prize).
- For each QuestionId_Answer row in the test set, it must predict the corresponding MisconceptionId.
- The execution time should not exceed 9 hours of only use of the CPU or GPU
- Data cannot be downloaded at runtime.
- The use of external data is permitted as long as it is free and publicly available.
- Pre-trained models may be used
- The submission file must be named submission.csv.

3.3. Life Cycle Approach

- The approach could be extended to many subjects beyond around the educational context
- Potential integration with adaptive learning platforms (educational apps) to improve or adapt the feedback.

3.4. Holistic Thinking

- Aims to improve misconception tagging in education, considering both machine learning accuracy and computational efficiency.
- The involvement of multiple components like NLP models, human labeling, and evaluation metrics to work all together for the same objective.

4. Sensitivity analysis

4.1. Identification of Critical Variables

- Text Preprocessing Strategy: Tokenization, stopword removal, and handling of casing or special characters.
- Vectorization Method: Whether TF-IDF, word embeddings, or transformer-based embeddings are used.
- Classifier Parameters: Parameters such as regularization strength in Logistic Regression, or depth and number of estimators in tree-based models.
- Data Distribution: Imbalanced labels, especially when some misconceptions are underrepresented.

4.2. Definition of Variation Scenarios:

- Vary preprocessing strategies: compare raw text with lowercase and cleaned versions.
- Change embedding strategy: compare performance using TF-IDF versus transformer embeddings.
- Adjust model hyperparameters: experiment with Logistic Regression's regularization (C), or Decision Tree's max_depth.
- Modify the train/test data split ratio.

4.3. Measuring the Impact:

- Train and validate the model under each variation scenario.
- Record MAP@25 results for each setup.
- Example finding: using transformer embeddings instead of TF-IDF improved MAP@25 by 8%, while removing stopwords reduced performance by 4%.

4.4. Interpretation and Conclusions:

- The model is highly sensitive to the choice of embedding method.
- Removing certain preprocessing steps negatively impacts performance, showing they add meaningful information.
- Hyperparameter tuning significantly changes results, stressing the need for systematic search.
- Recommendation: standardize preprocessing and embedding choices, and use cross validation with hyperparameter optimization to ensure robust model performance.

- Propose recommendations to strengthen the system, such as implementing cross validation techniques or using more robust optimization methods.

5. Chaos Theory and Complexity:

5.1. Identification of Nonlinear Behaviors:

- The NLP component introduces non-linearities due to the contextual nature of language.
- Minor variations in student answers may trigger entirely different embeddings and lead to different misconception predictions.
- The model sometimes assigns multiple misconceptions to similar answers, suggesting hidden interactions among error types.

5.2. Sensitivity to Initial Conditions:

- Minor changes in the dataset, like removing one answer or rewording a question, can cascade into different prediction outcomes.
- Models trained on slightly different data subsets can produce divergent results due to the class imbalance and label subjectivity.

5.3. System Complexity:

- The system combines human labeling, NLP encoding, multi-label classification, and evaluation using MAP@25.
- Complexity arises from the human in the loop feedback cycle: misconceptions are defined by teachers, but modeled computationally.
- The dynamic nature of language and knowledge introduces emergent behaviors where patterns are not explicitly programmed but arise from data interactions.

5.4. Mitigation and Control Strategies:

- Use ensemble models to average out erratic behaviors from individual classifiers.
- Apply label smoothing or robust loss functions to account for label noise.

- Regularly retrain the model with updated labeled data to adjust to evolving misconceptions.
- Perform perturbation testing: introduce small changes to input samples and measure prediction stability.

6. **Conclusions:** The analysis of the Eedi - Mining Misconceptions in Mathematics competition system reveals a complex interaction between its components. The data provided (questions, answers, constructs, and errors) are articulated with machine learning models that must meet specific evaluation metrics and operate under technical constraints. Each element fulfills a key function within the system, contributing to its main objective: identifying common error patterns in mathematical learning.

   One of the most relevant findings is the need to efficiently automate the misconceptions labeling, which has traditionally been a manual and costly process in terms of time and consistency. The competition proposes a robust computational solution that, by relying on natural language processing, seeks to accelerate this process and offer more accurate and scalable results. From a systemic perspective, the competition design is aligned with fundamental principles of systems engineering, such as the identification of key stakeholders (students and teachers), the clear definition of constraints (use of computational resources, submission format, participation rules), and a comprehensive approach that seeks to enhance learning through technology.

   Furthermore, it is observed that this system promotes adaptive learning, as the results obtained can be integrated into educational platforms to provide personalized feedback. By accurately identifying students' misconceptions, opportunities are opened for early and targeted interventions in their learning process.

   The use of the MAP@25 metric as an evaluation criterion not only incentivizes accurate models, but also practically useful ones, focusing on the quality of the most relevant predictions. This ensures that the proposed solutions have a real impact in educational settings.

Furthermore, the prizes offered not only recognize the accuracy of the models but also their efficiency, thus encouraging the development of solutions that can be applied in real-life contexts where resources are limited. This incentive structure contributes to greater applicability of the results.

Finally, the system is designed to allow for continuous improvement. By accepting public data and pre-trained models, the competition remains open to innovation and collaborative learning, fostering an active community that can continue to enrich the system even after its formal closure.

7. References:

- An Effective Requirement Engineering Process Model for Software Development and Requirements Management. IEEE Conference Publication | IEEE Xplore. Published 1 de octubre de 2010. https://ieeexplore.ieee.org/abstract/document/5656776/
- Wolfram, S. *Chaos Theory – An Introduction*. Wolfram MathWorld. Published online. https://mathworld.wolfram.com/Chaos.html
- Molnar, C. *Interpretable Machine Learning Book*. 2022. https://christophm.github.io/interpretable-ml-book/