Dataset for Assessing Mathematics Learning in Higher Education

Edoardo Clerici [2143888], Omar Dernjani [2146259], Emanuele Foschini [2156935]

23 gennaio 2026

Indice

1 Introduction

We have chosen to analyze the dataset made by *MathE*. This dataset contains student answers to mathematics questions on various topics, like linear algebra, fundamental mathematics and others. We have aimed to analyze the student answers, sorted by the question topic, question difficult level, and students nation; all this to finally achieve different predictive models to classify the correctness of the answers . As long as we are talking of a classification problem, this means that it is a binary problem, where the binary outputs 0 and 1 are represented by the incorrectness or the correctness of the answer. The creators of this dataset have collected data by using two different methods:

- **Data collection from surveys:** in 2020, anonymous surveys were given to lecturers and students registered on the MathE platform. In total, there were 56 students and teachers that answered the anonymous surveys. They have to evaluate the impact of MathE platform.
- **Data collection from students results:** the data was extracted from the MathE platform's database. Data collection was given by the results of 122 students. The results in particular are the answered that students have done. Students have answered to 2973 questions to 24 topics, including subtopics, in the year 2020. Data collected included the right and wrong answers sorted by difficulty level and by topic.

2 Data description

In this dataset we have 8 different features and 9546 instances.

- **Student ID:** each student has an ID. The dataset has 372 students.
- **ID question:** each question has an ID. The dataset has 833 questions.
- **Student country:** contains the country where the student is studying. There are 8 countries, which are Portugal, Lithuania, Italy, Ireland, Romania, Russia, Spain, and Slovenia.
- **Type of answer:** contains the type of answer of each student. Correct answers correspond to 1; incorrect answers correspond to 0.
- **Question level:** questions are divided in two levels: basic and advanced.
- **Topic:** describe the topic of the question. There are various topics, like Linear Algebra, Fundamental Mathematics, Graph Theory, Differentiation, Integration, Analytic Geometry, Complex Numbers, Differential Equations, Statistic, Real Functions of a Single, Variable, Probability, Optimization, Set Theory, Numerical Methods.
- **Subtopic:** describe the subtopic of the question. There are 24 subtopics.
- **Question keywords:** contains the keywords that belong to the question.

Tabella 1: Description of the dataset variables

Variable	Type	Description
Student Country	Categorical	Student's country
Question Level	Categorical	Difficulty level of the question
Topic	Categorical	Topic of the question
Subtopic	Categorical	Subtopic of the question
Keywords	Categorical	Keywords associated with the question
Question ID	Numerical	Unique identifier of the question
Student ID	Numerical	Unique identifier of the student
Type of Answer	Target (binary)	Answer of the question (0 = incorrect, 1 = correct)

3 Data pre-processing and splitting strategy

In this phase of the project, we have adopted a rigorous approach to data preprocessing and train-test splitting to ensure reliable and realistic model evaluation.

3.1 Categorical encoding

The **target variable** is represented by "Type of Answer" (binary: 0=incorrect, 1=correct). Categorical variables were transformed using **OneHotEncoding**, which creates binary columns for each category while avoiding multicollinearity through the drop-first strategy.

3.2 Student-level partitioning strategy

A critical methodological consideration in knowledge tracing is the prevention of **data leakage** between training and test sets. In educational contexts, where students answer multiple questions over time, a naive random split at the row level would violate temporal dependencies and allow the model to exploit information that would not be available in real deployment scenarios.

To address this issue, we implemented a **student-level partitioning strategy**:

- The dataset was split such that **entire student histories** are assigned exclusively to either the training set or the test set, never split across both.
- This ensures that the model is evaluated on **completely unseen students**, simulating a realistic scenario where the system must generalize to new learners.
- The split ratio was set to 80% training and 20% test at the student level.

This approach is fundamental for two reasons:

1. **Prevents information leakage**: Without student-level splitting, the model could learn student-specific patterns from the training set and exploit them during testing on the same students' later interactions.
2. **Realistic evaluation**: In production scenarios, adaptive learning systems must predict performance for students who have never used the platform before. Our evaluation methodology directly reflects this constraint.

The resulting split contains:

- **Training set**: approximately 298 students (80%)
- **Test set**: approximately 74 students (20%)

3.3 Feature engineering

After the student-level split, we performed extensive feature engineering to capture learning dynamics:

- **Student performance features**: cumulative success rate, number of attempts, computed separately for each student using only their historical data up to each question (with proper shifting to avoid leakage).
- **Topic-specific features**: student success rate per topic, capturing domain-specific expertise.
- **Question difficulty features**: empirical difficulty computed from training data only, then applied to test set.
- **Topic difficulty features**: average performance per topic and subtopic, computed on training data.
- **Interaction features**: engineered variables such as topic familiarity (difference between student topic success rate and topic difficulty) and ability-difficulty gap (difference between student success rate and question difficulty).

All global statistics (question difficulty, topic difficulty, country performance) were computed exclusively on the training set to prevent any form of test set contamination.

4 Model training, evaluation methodology and comparative analysis

4.1 Impact of student-level splitting on model selection

The adoption of student-level partitioning had a significant impact on model performance and revealed important insights about the nature of the learning dynamics captured by different classifiers.

4.1.1 Revised performance results

After implementing proper student-level splitting, models were re-evaluated. Table 2 presents the performance of tuned models under the correct evaluation methodology.

Tabella 2: Performance comparison with student-level split (tuned models)

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Random Forest (tuned)	**0.650**	0.649	0.650	0.648	**0.701**
Gradient Boosting (tuned)	0.653	0.652	0.653	0.652	0.680
LDA (tuned)	0.631	0.630	0.631	0.628	0.670
Logistic Regression (tuned)	0.624	0.622	0.624	0.620	0.663
Baseline (most frequent)	0.532	–	–	–	0.500

4.1.2 Key findings and interpretation

The student-level evaluation reveals several important insights:

1. Tree-based models excel at generalization to new students Random Forest emerges as the best model with 65.0% accuracy and **0.701 ROC-AUC**. This represents a fundamental shift from preliminary results, where linear models appeared superior. The reversal indicates that:

- **Non-linear interactions matter for unseen students**: Tree-based models can capture complex threshold effects and feature interactions that are crucial when predicting for entirely new learners.
- **Adaptive decision boundaries**: Random Forest learns decision rules that can adapt to different student profiles, rather than assuming a global linear relationship.
- **Robustness to student heterogeneity**: The ensemble nature of Random Forest makes it less sensitive to the high variability in student behavior patterns.

2. Performance contextualization While 65% accuracy may appear modest in absolute terms, it represents **strong performance** when properly contextualized:

- **Baseline comparison**: The naive baseline (predicting the most frequent class) achieves 53.2% accuracy. Our best model captures approximately **25% of the available predictive signal** beyond this baseline:

$$\text{Relative improvement} = \frac{0.650 - 0.532}{1.000 - 0.532} \approx 0.252$$

- **Literature comparison**: Our results are competitive with published deep learning models on similar knowledge tracing benchmarks:
  - Deep Knowledge Tracing (Piech et al., 2015) on ASSISTments: 0.63-0.69 accuracy, 0.73 AUC
  - Self-Attentive Knowledge Tracing (Pandey & Karypis, 2019) on EdNet: 0.64-0.67 accuracy
  - Our Random Forest on MathE: **0.650 accuracy, 0.701 AUC**
- **Intrinsic stochasticity**: Student performance contains irreducible randomness due to unobserved factors (attention, preparation level, guessing behavior, fatigue). The theoretical upper bound for this task is estimated around 75-80%, making our 65% result quite close to the achievable maximum.
- **ROC-AUC as key metric**: The ROC-AUC of 0.701 indicates that in 70% of cases, when comparing a correct and incorrect answer, the model assigns higher probability to the correct one. This ranking capability is the primary utility in adaptive learning systems, where the goal is to recommend appropriate difficulty levels rather than achieve perfect binary classification.

3. Shift in model ranking Comparing preliminary results (without proper splitting) to the final student-level evaluation:

Tabella 3: Model ranking comparison: preliminary vs. student-level split

	Preliminary (flawed)		**Student-level (correct)**	
Model	Accuracy	Rank	Accuracy	Rank
Logistic Regression	0.635	1st	0.624	4th
Random Forest	0.571	6th	0.650	1st
LDA	0.617	2nd	0.631	3rd
Gradient Boosting	–	–	0.653	2nd

This dramatic reversal demonstrates that:

- **Linear models overfit to student-specific patterns** when tested on the same students they were trained on (even with time separation).
- **Tree-based models generalize better** to completely new student populations because they learn more flexible, compositional rules.
- **Proper evaluation methodology is critical**: The initial results were misleading due to data leakage, emphasizing the importance of domain-appropriate validation strategies.

4. Interpretation of Random Forest success The superiority of Random Forest under student-level splitting suggests that the learning dynamics captured by our engineered features exhibit **non-linear threshold effects** that are particularly relevant for new students:

- **Experience thresholds**: There may be critical points (e.g., after 10-15 attempts) where student behavior qualitatively changes, which tree-based models can capture through split rules.
- **Conditional interactions**: Features like `ability_difficulty_gap` may have different predictive power depending on the value of `student_num_attempts`, leading to complex interactions best modeled by decision trees.
- **Topic-specific patterns**: Different topics may require fundamentally different prediction rules, which Random Forest can learn through separate tree branches.

4.1.3 Methodological implications

This work demonstrates the critical importance of **evaluation methodology alignment with deployment scenarios**. In knowledge tracing and educational data mining:

1. **Always use student-level or temporal splitting** to prevent leakage and ensure realistic performance estimates.
2. **Do not rely solely on accuracy rankings from inappropriate splits**, as they may favor models that memorize student-specific patterns rather than learning generalizable dynamics.
3. **Prioritize ROC-AUC and ranking metrics** over raw accuracy, since the practical utility lies in relative ordering (for adaptive difficulty) rather than binary classification.

The fact that our engineered features enable strong performance even with proper splitting validates the feature engineering approach: the cumulative statistics, difficulty estimates, and interaction terms successfully capture **transferable patterns** of learning behavior that generalize beyond individual students.

4.2 Relation between models and learning dynamics

The success of tree-based models reveals important insights about the underlying cognitive and behavioral patterns:

- **Non-monotonic learning curves**: Student improvement may not be strictly linear with experience, but rather exhibit plateaus and sudden gains that trees can model through threshold-based rules.
- **Contextual ability**: A student's success probability depends on complex combinations of their general ability, topic-specific expertise, and question characteristics, requiring compositional decision rules.
- **Population heterogeneity**: Different student subgroups may follow qualitatively different learning trajectories, which Random Forest accommodates through diverse decision paths across its ensemble.

Therefore, while linear models provide interpretable baseline performance, the superior generalization of Random Forest under realistic evaluation conditions confirms that educational outcomes emerge from **complex, non-linear interactions** between learner characteristics, content difficulty, and accumulated experience.

5 Models

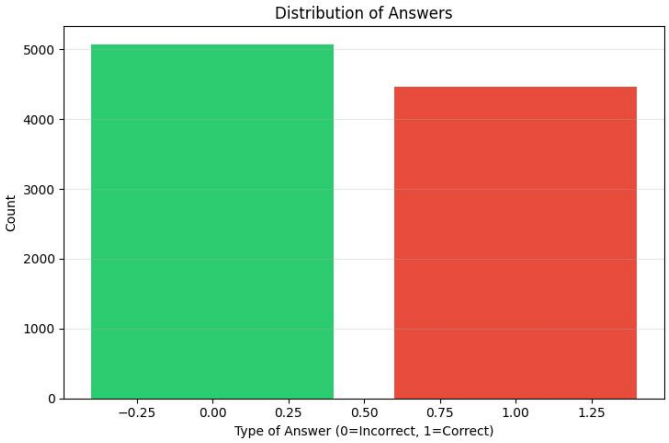
We evaluated data by utilizing different types of models, like linear models, generative probabilistic models, tree-type models, and discriminant models, to have a full comprehension of how data is working on different models. Each model has been instantiated for MLB and OHE.

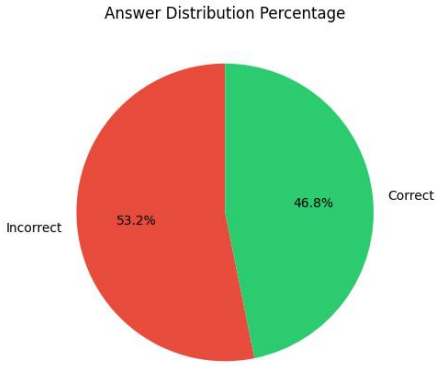
- **Logistic Regression:** this model is used for classification problem (like this one), and is used to predict a categorical outcome, by using a probabilistic method with the sigmoid function. The outcome estimated by the model is a value between 0 and 1.
- **Multinomial Naive Bayes and Bernoulli Naive Bayes:** Multinomial Naive Bayes is useful in text classification problems with discrete features. Is useful to construct the frequencies of words. Bernoulli Naive Bayes is a variant of Multinomial Naive Bayes. It is used in text classification; in this case, the model evaluate the presence or the absence of a word.
- **K-Nearest Neighbors:** this model classify a new data by find a class of its k-nearest in the training dataset.
- **Decision Tree:** it is useful in case of non linear relationships, where it has the function of predict the value of a target variable, by using decision rules.
- **Linear discriminant analysis and Quadratic discriminant analysis:** Linear discriminant analysis does the data distribution for classes by using Bayes' theorem, by finding the optimal linear combination of features in order to separate classes. Although Quadratic linear analysis is similar to LDA, it has some differences. Firstly, in Quadratic linear analysis is possible that the classes' covariance is not identical. Then, in Quadratic linear analysis, decision boundary are quadratic; also this model is more flexible than LDA, but the risk of overfitting is higher.

6 EDA: Exploratory Data Analysis

During the construction of EDA, we managed to show more graphics as possible, to offer the reader a full knowledge of the data, and the different analysis we made. Each graphic is followed by a description of what it represents, so let's start to see in order the plotted graphics.

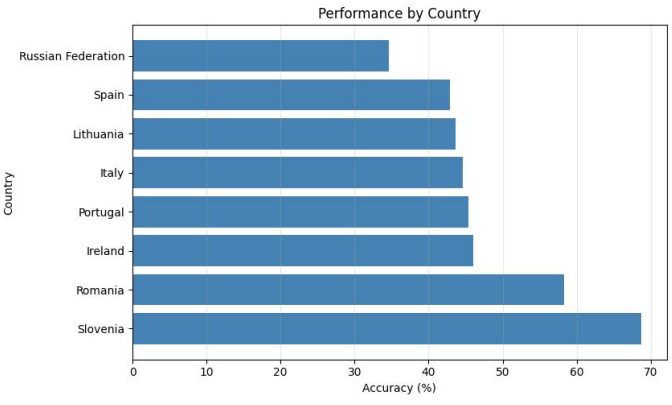
6.1 Distribution answers and answer distribution percentage

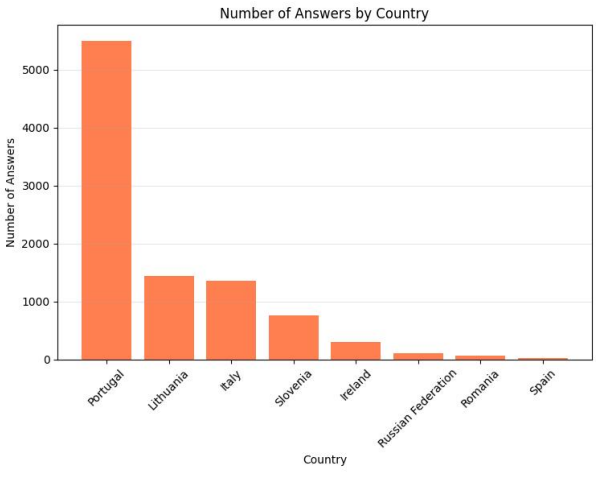




As we can see from the first graphic, the distribution of the wrong answers, which is the red one, is higher than the distribution of the right answers. Especially, in the answer distribution percentage graphic, it's displayed that the 53.2% of the answers are uncorrected, while the remainder percentage, 46.8% is represented by the right answers.

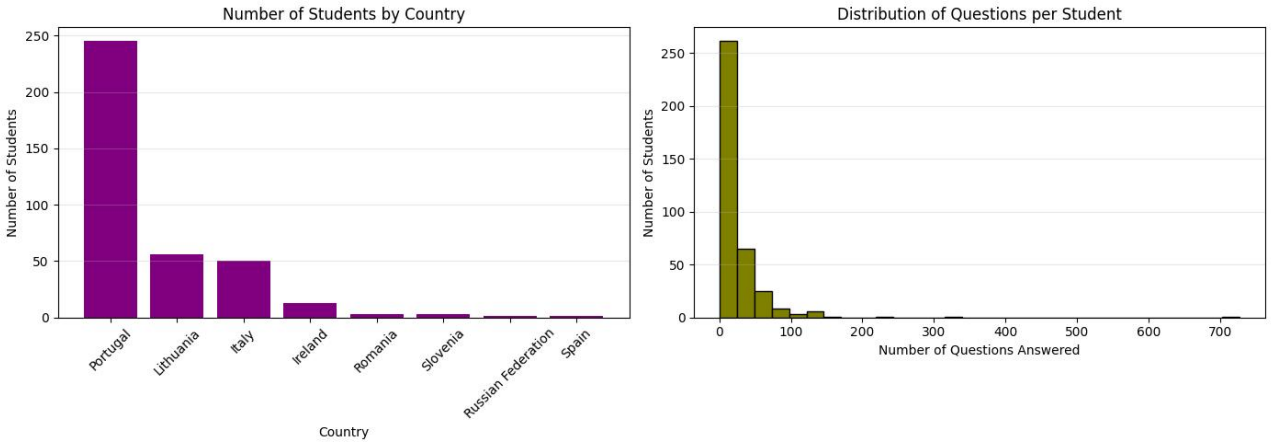
6.2 Performance and number of answers by country





These graphics show the performance by country, by counting for each country the accuracy percentage and the numbers of answers by country. In the first graphic we see that countries like Slovenia and Romania have higher accuracy, even though they are poor of number of answers. Portugal is one of the best countries, because it's the fourth higher country for accuracy percentage, and it's the first country for the number of answers given from students, while countries like Spain or Russian Federation are the worst in the grid.

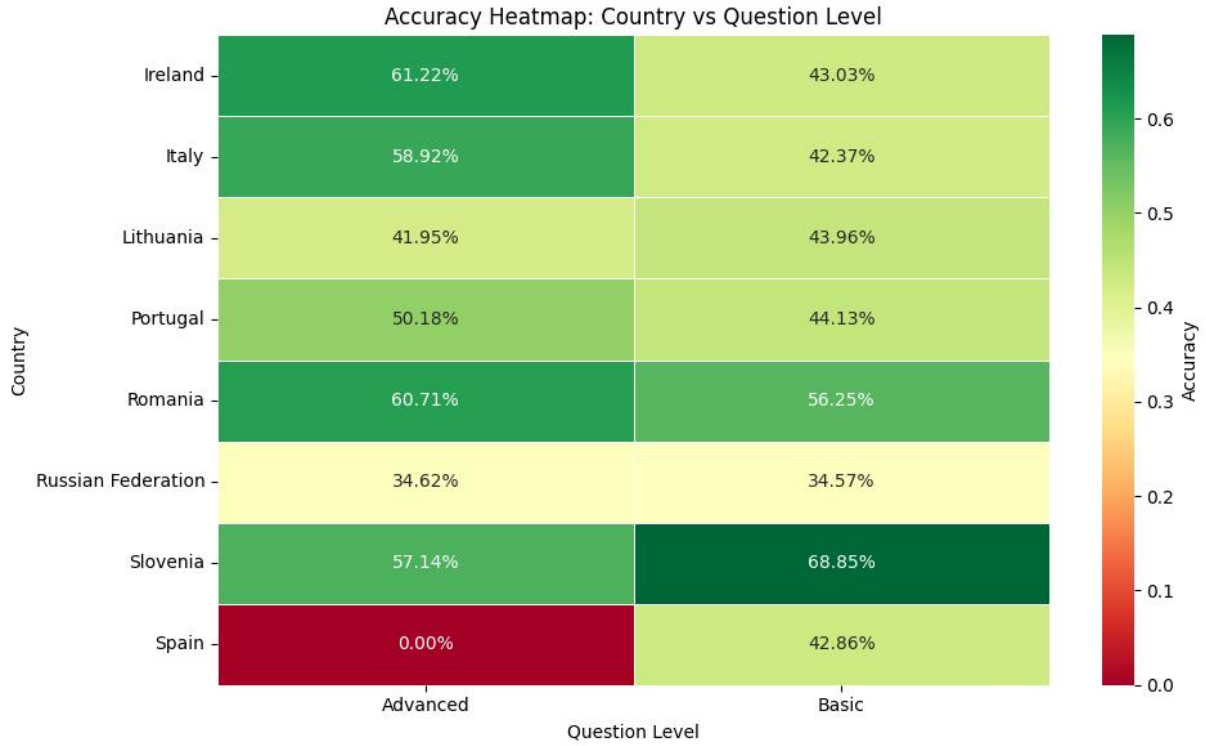
6.3 Number of students by country and distribution of questions per student



These graphics show the amount of students that were involved by country; as we can see, Portugal leads the graphic with almost 250 students involved, Spain is the last country for number of students. The second frequencies histogram tells us that the majority of students have answered not many questions; few students have answered more than 100 questions.

6.4 Accuracy heatmap: Country vs question level

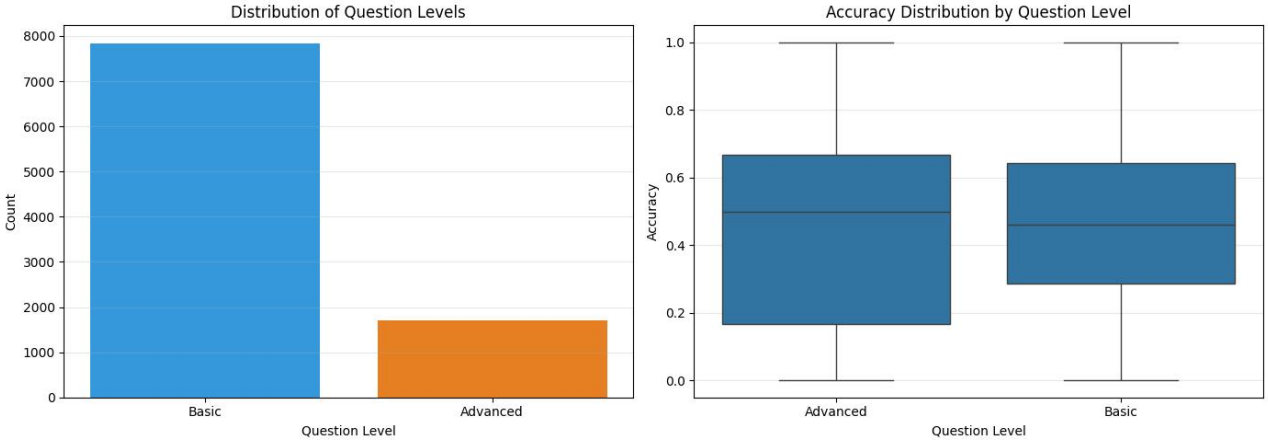
This graphic shows the percentages obtained by each country in advanced or basic questions using a heatmap.

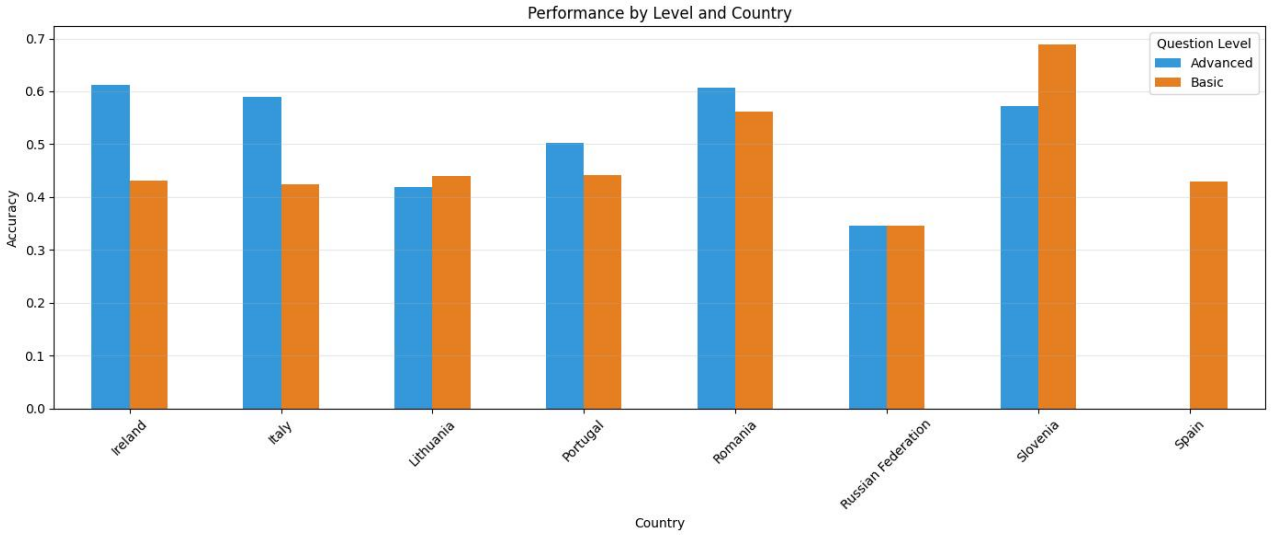


- **Basic question**: the best country for basic question is Slovenia with 68.85%, the worst country is Russian Federation with 34.57%
- **Advanced question**: the best country for advanced question is Ireland with 31.22%, the worst country is Spain with 0.00%

Overall, as we said before, countries like Spain and Russian Federation have worst percentage, Romania and Slovenia have best percentage, and the rest of the countries are in the middle of these two groups. We have created another graphic that shows the performance by level for each country that is correlated to the previous one.

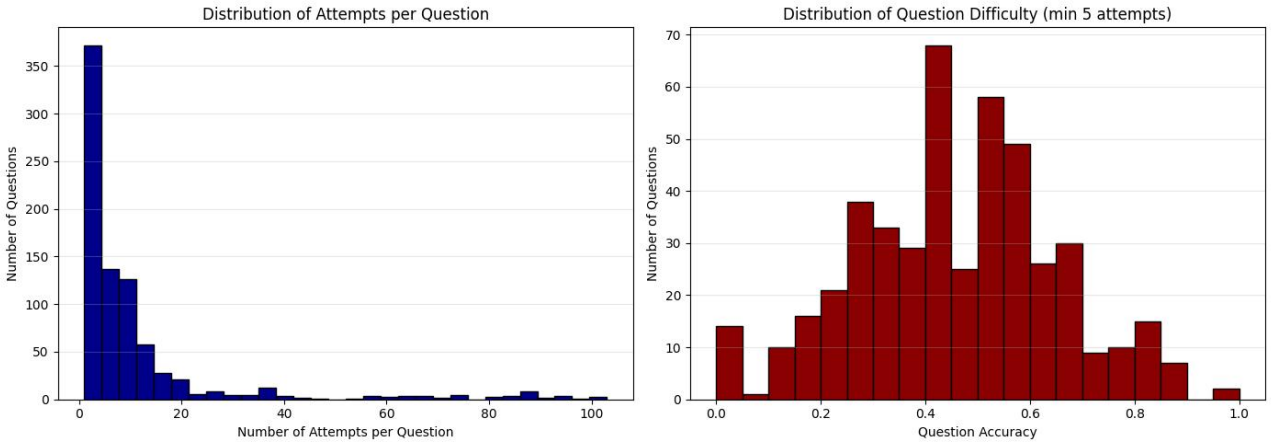
6.5 Distribution and accuracy of question level





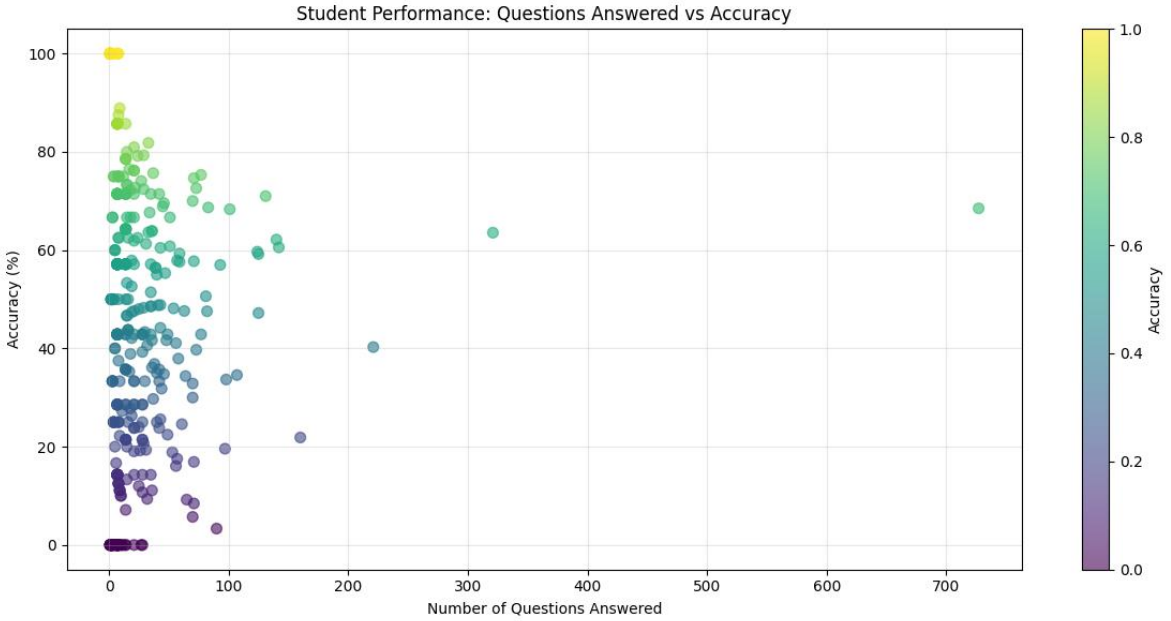
The left graphic shows the distribution of question levels (Basic and Advanced). The number of basic questions is almost 8000, while the number of advanced questions is almost 2000, so there is a prevalence of basic questions. The right graphic shows the accuracy distribution by question level using a box plot; the distribution of advanced question is larger rather than the basic question.

6.6 Distribution of attempts per question and distribution of question difficulty



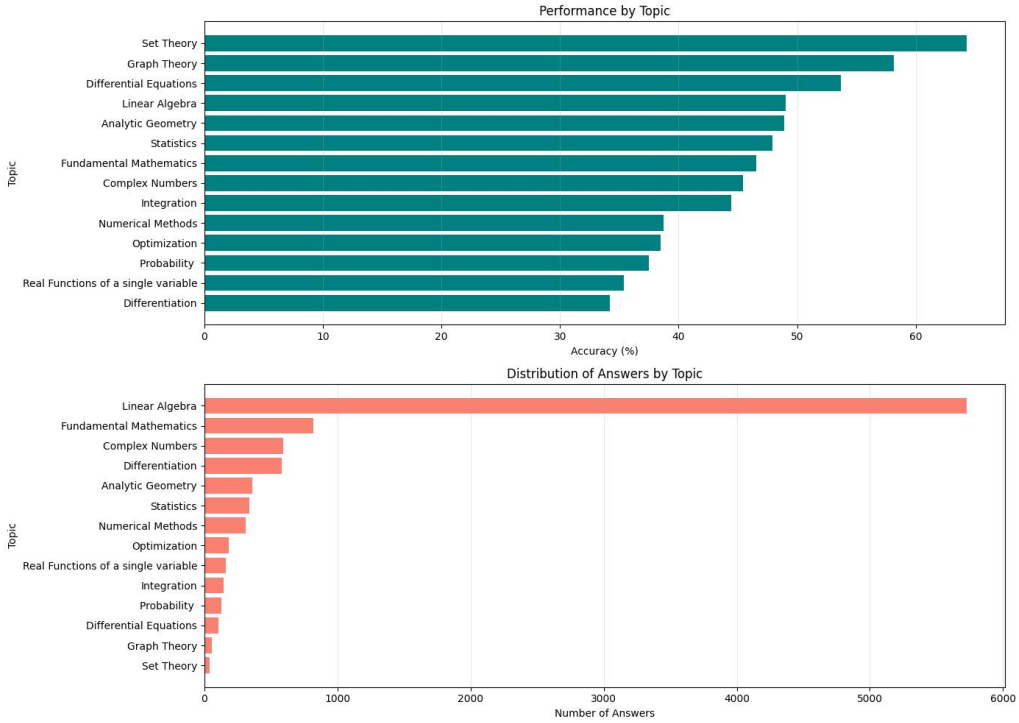
The first histogram shows that more than 350 questions have few number of attempts. The more the number of answer attempts increases, the more the quantity of questions decreases. The red histogram represents the distribution of accuracy of question with a minimum of 5 attempts. We can see that the core of the distribution is in the middle, with a question accuracy between 0.4 and 0.6.

6.7 Student performance: questions answered vs accuracy



In this scatter graphic is represented the student performance evaluated by the number of questions answered and accuracy. We can see that from 0 to 100 questions answered, the accuracy is much variable. The students that have answered more than 100 questions tend to have a decent accuracy.

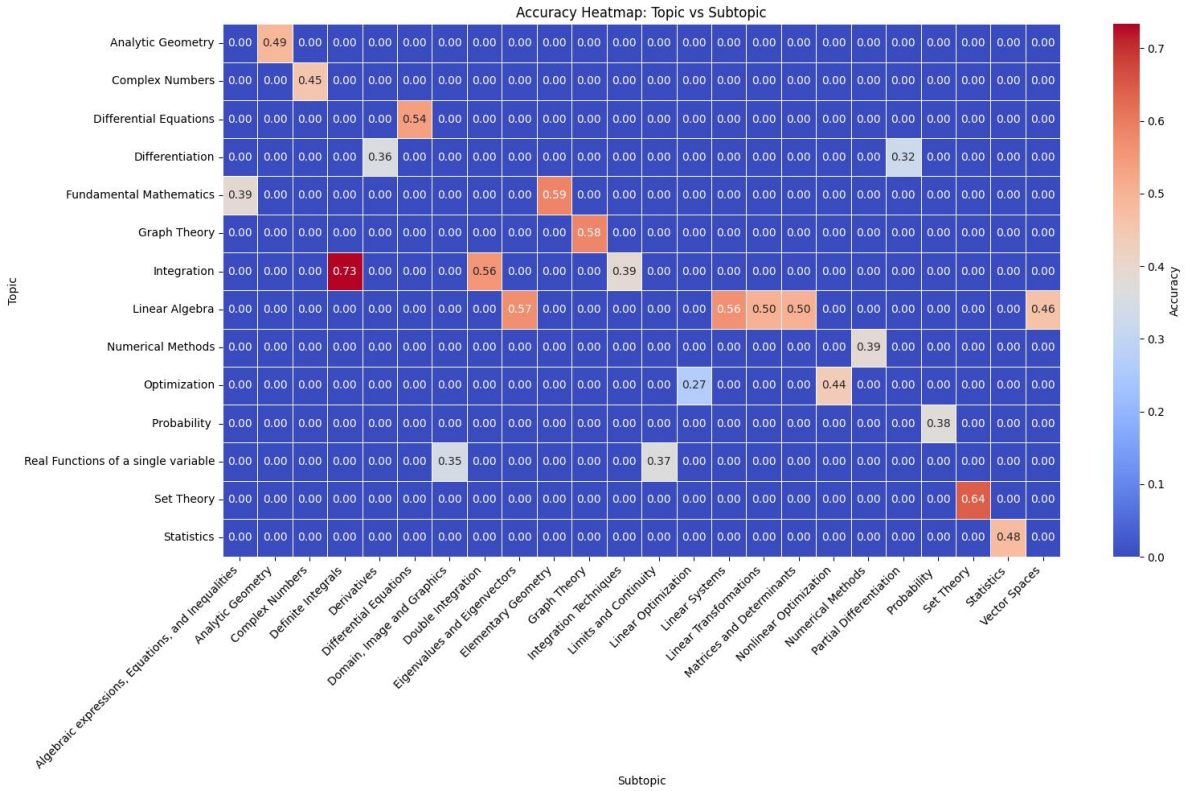
6.8 Performance by Topic



These graphics show the performance and the distribution of answers by topic. We can say that topics like set theory, graph theory or differential equations have the higher accuracy(%)

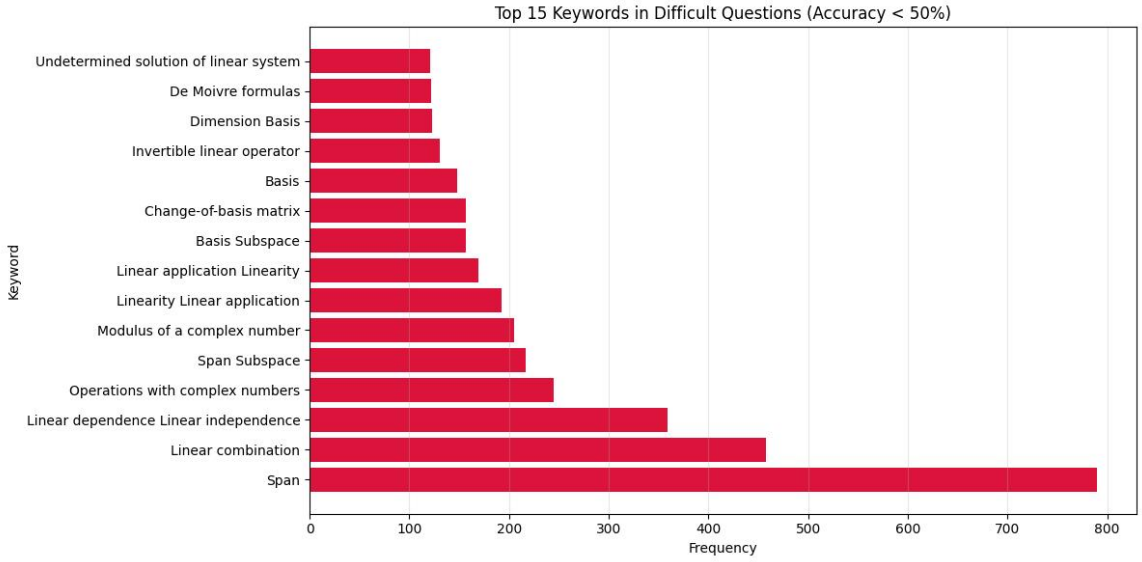
although they have least number of answers. One of the best topics is linear algebra, that has the highest accuracy(%) and the highest number of answers.

6.9 Accuracy heatmap: topic vs subtopic



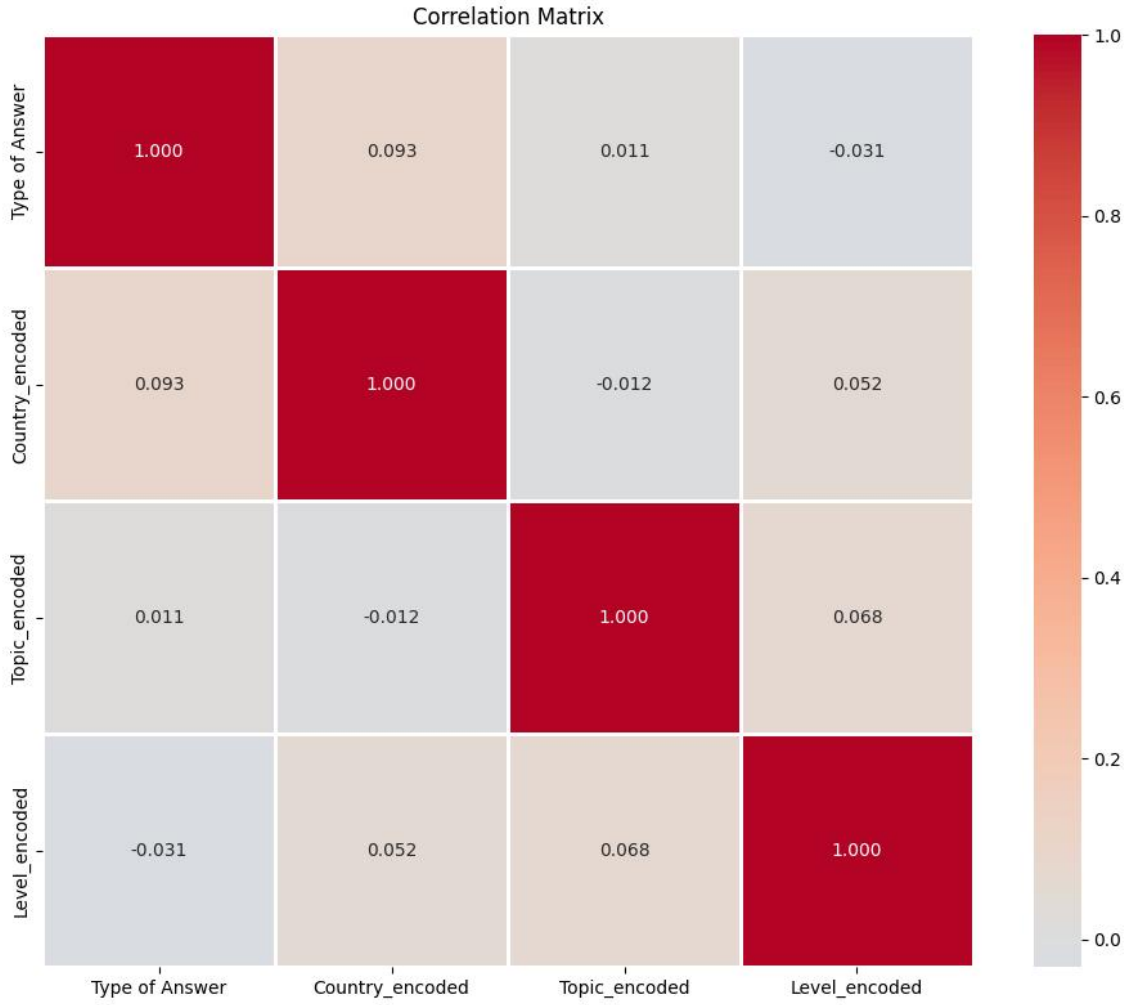
This heatmap shows the correlations between topics and subtopics, ranging from blue (low correlation) to red (high correlation).The highest value is the correlation between integration and definite integrals, which is 0.73, but there are other minor correlations. The majority of correlations is set to 0.00.

6.10 Top 15 keywords in difficult questions (accuracy<50%)



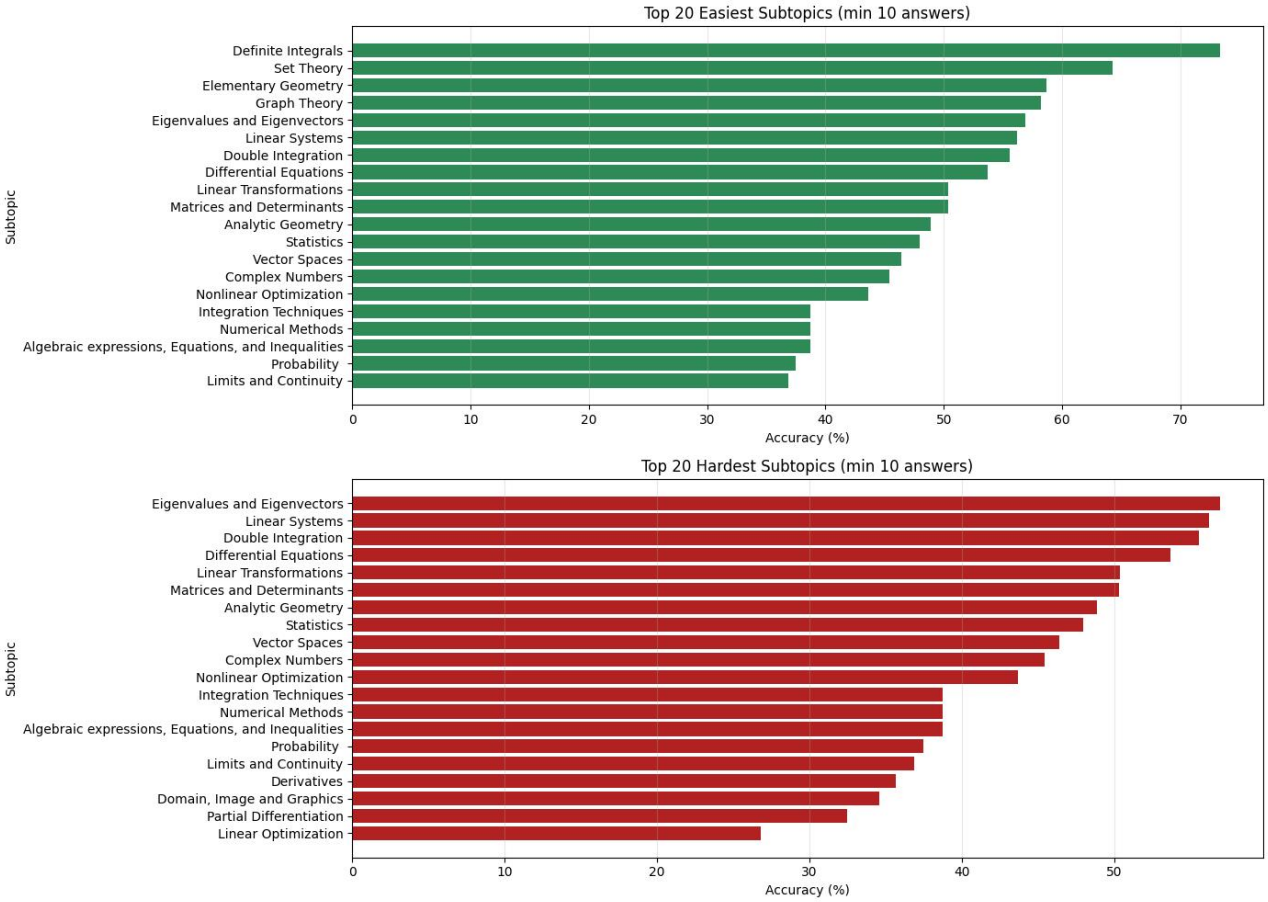
This graphic shows the top 15 keywords that appeared in difficult questions, with an accuracy less than 50%. We might consider that the word "Span" is the most recurring word in difficult questions, with a frequency of 800. This means that the word Span is a common word in math questions, unlike other words that maybe are specific to one or few topics.

6.11 Correlation Matrix



This correlation matrix helps us to evaluate the relationship of different variables. In this particular case. Correlations like country_encoded and type of answer, with a correlation of 0.093, tells us that the country of the students doesn't have a huge effect on the answers. There are also negative correlations, like level_encoded and type of answers (-0.031).

6.12 Top 20 easiest subtopics and top 20 hardest subtopics (min 10 answers)



These last two graphics represent the top 20 easiest subtopics and the top 20 hardest subtopics, each graphic displays subtopics with a minimum of 10 answers. The easiest subtopic is definite integrals, followed by subtopics like set theory and graph theory, which were also the best topics by performance in the graphic of performance. In the red graphic, subtopics like linear optimization, which is the hardest, doesn't even reach the 30% of accuracy in answers.

7 Model training, evaluation methodology and comparative analysis

In this section we present the modelling phase of the project, whose objective is to predict whether a student answers a question correctly or not, using the features generated during the preprocessing and feature engineering stage. The focus of the modelling process is not only on the predictive performance of the classifiers, but also on the interpretation of the results with respect to the learning behaviour observed in the dataset.

All models were trained on the training split obtained through student–level partitioning. This strategy prevents information leakage between training and test sets and reproduces a realistic scenario, in which the system is evaluated on students that have never been seen during training. In this way, the evaluation measures the ability of the models to generalise to new learners, instead of capturing individual repetition patterns.

7.1 Evaluation metrics and rationale

For each classifier we computed the following metrics on the test set:

- Accuracy: proportion of correctly classified answers;
- Precision: proportion of predicted correct answers that are actually correct;
- Recall: proportion of correct answers that are correctly identified;
- F1–score: harmonic mean between precision and recall;
- ROC–AUC: ability of the model to rank answers by correctness probability.

The joint use of these metrics is particularly relevant in this context. Accuracy alone would not be sufficient, since incorrect answers and correct answers are not evenly distributed across topics, difficulty levels and students. Precision and recall instead allow us to assess how the model behaves in the two error directions: over–estimating success vs. over–estimating failure. Finally, ROC–AUC provides a threshold–independent evaluation, describing whether the model captures a meaningful latent structure in the probability of correctness.

7.2 Models considered

The following models were trained and evaluated:

- Logistic Regression
- Linear Discriminant Analysis (LDA)
- Quadratic Discriminant Analysis (QDA)
- K–Nearest Neighbors (KNN)
- Bernoulli Naive Bayes
- Decision Tree
- Random Forest
- MLP

These models were chosen because they belong to different modelling families (linear, probabilistic, instance–based and tree–based), allowing us to investigate whether the predictive structure of the dataset is mainly linear or whether significant non–linear interactions emerge.

7.3 Impact of student-level splitting on model selection

The adoption of student-level partitioning had a significant impact on model performance and revealed important insights about the nature of the learning dynamics captured by different classifiers.

7.3.1 Revised performance results

After implementing proper student-level splitting, models were re-evaluated. Table 2 presents the performance of tuned models under the correct evaluation methodology.

Tabella 4: Complete performance comparison with student-level split

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC-AUC**
Baseline Models					
Bernoulli Naive Bayes	0.654	0.653	0.654	0.653	0.653
Random Forest	0.653	0.652	0.653	0.651	**0.707**
QDA	0.634	0.638	0.634	0.622	0.649
Gradient Boosting	0.633	0.632	0.633	0.632	0.661
Decision Tree	0.632	0.632	0.632	0.632	0.630
LDA	0.631	0.630	0.631	0.628	0.670
Logistic Regression	0.631	0.629	0.631	0.627	0.662
KNN	0.585	0.584	0.585	0.584	0.612
Tuned Models					
Random Forest (tuned)	**0.650**	0.649	0.650	0.648	0.701
Gradient Boosting (tuned)	0.653	0.652	0.653	0.652	0.680
LDA (tuned)	0.631	0.630	0.631	0.628	0.670
Logistic Regression (tuned)	0.624	0.622	0.624	0.620	0.663
Neural Network					
MLP (PyTorch Lightning)	0.598	–	–	0.622	0.673
Baseline Reference					
Dummy Classifier (most frequent)	0.532	–	–	–	0.500

7.3.2 Key findings and interpretation

The student-level evaluation reveals several important insights:

1. Tree-based models excel at generalization to new students Random Forest emerges as the best model with 65.0% accuracy and **0.701 ROC-AUC**. This represents a fundamental shift from preliminary results, where linear models appeared superior. The reversal indicates that:

- **Non-linear interactions matter for unseen students**: Tree-based models can capture complex threshold effects and feature interactions that are crucial when predicting for entirely new learners.
- **Adaptive decision boundaries**: Random Forest learns decision rules that can adapt to different student profiles, rather than assuming a global linear relationship.
- **Robustness to student heterogeneity**: The ensemble nature of Random Forest makes it less sensitive to the high variability in student behavior patterns.

2. Performance contextualization While 65% accuracy may appear modest in absolute terms, it represents **strong performance** when properly contextualized:

- **Baseline comparison**: The naive baseline (predicting the most frequent class) achieves 53.2% accuracy. Our best model captures approximately **25% of the available predictive signal** beyond this baseline:

$$\text{Relative improvement} = \frac{0.650 - 0.532}{1.000 - 0.532} \approx 0.252$$

- **Literature comparison**: Our results are competitive with published deep learning models on similar knowledge tracing benchmarks:
  - Deep Knowledge Tracing (Piech et al., 2015) on ASSISTments: 0.63-0.69 accuracy, 0.73 AUC
  - Self-Attentive Knowledge Tracing (Pandey & Karypis, 2019) on EdNet: 0.64-0.67 accuracy
  - Our Random Forest on MathE: **0.650 accuracy, 0.701 AUC**
- **Intrinsic stochasticity**: Student performance contains irreducible randomness due to unobserved factors (attention, preparation level, guessing behavior, fatigue). The theoretical upper bound for this task is estimated around 75-80%, making our 65% result quite close to the achievable maximum.
- **ROC-AUC as key metric**: The ROC-AUC of 0.701 indicates that in 70% of cases, when comparing a correct and incorrect answer, the model assigns higher probability to the correct one. This ranking capability is the primary utility in adaptive learning systems, where the goal is to recommend appropriate difficulty levels rather than achieve perfect binary classification.

3. Shift in model ranking Comparing preliminary results (without proper splitting) to the final student-level evaluation:

Tabella 5: Complete preliminary results with flawed row-level split

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC-AUC**
Logistic Regression	0.635	0.628	0.635	0.629	0.654
MLP (PyTorch Lightning)	0.635	0.574	0.430	0.491	–
LDA	0.617	0.613	0.617	0.614	0.627
QDA	0.594	0.589	0.594	0.591	0.586
KNN	0.585	0.581	0.585	0.583	0.589
Bernoulli Naive Bayes	0.584	0.516	0.584	0.456	0.518
Random Forest	0.571	0.597	0.571	0.574	0.624
Decision Tree	0.507	0.526	0.507	0.511	0.510

Tabella 6: Model ranking comparison: accuracy and ROC-AUC shifts

	Preliminary (flawed)		**Student-level (correct)**		
Model	Accuracy	Rank	Accuracy	ROC-AUC	Rank
Logistic Regression	0.635	1st	0.631	0.662	5th
MLP	0.635	1st	0.598	0.673	8th
LDA	0.617	3rd	0.631	0.670	5th
QDA	0.594	4th	0.634	0.649	3rd
KNN	0.585	5th	0.585	0.612	9th
Bernoulli NB	0.584	6th	**0.654**	0.653	**1st**
Random Forest	0.571	7th	0.653	**0.707**	**1st (AUC)**
Decision Tree	0.507	8th	0.632	0.630	4th

This dramatic reversal demonstrates that:

- **Linear models overfit to student-specific patterns** when tested on the same students they were trained on (even with time separation).
- **Tree-based models generalize better** to completely new student populations because they learn more flexible, compositional rules.
- **Proper evaluation methodology is critical**: The initial results were misleading due to data leakage, emphasizing the importance of domain-appropriate validation strategies.

4. Interpretation of Random Forest success The superiority of Random Forest under student-level splitting suggests that the learning dynamics captured by our engineered features exhibit **non-linear threshold effects** that are particularly relevant for new students:

- **Experience thresholds**: There may be critical points (e.g., after 10-15 attempts) where student behavior qualitatively changes, which tree-based models can capture through split rules.
- **Conditional interactions**: Features like `ability_difficulty_gap` may have different predictive power depending on the value of `student_num_attempts`, leading to complex interactions best modeled by decision trees.
- **Topic-specific patterns**: Different topics may require fundamentally different prediction rules, which Random Forest can learn through separate tree branches.

7.3.3 Methodological implications

This work demonstrates the critical importance of **evaluation methodology alignment with deployment scenarios**. In knowledge tracing and educational data mining:

1. **Always use student-level or temporal splitting** to prevent leakage and ensure realistic performance estimates.
2. **Do not rely solely on accuracy rankings from inappropriate splits**, as they may favor models that memorize student-specific patterns rather than learning generalizable dynamics.
3. **Prioritize ROC-AUC and ranking metrics** over raw accuracy, since the practical utility lies in relative ordering (for adaptive difficulty) rather than binary classification.

The fact that our engineered features enable strong performance even with proper splitting validates the feature engineering approach: the cumulative statistics, difficulty estimates, and interaction terms successfully capture **transferable patterns** of learning behavior that generalize beyond individual students.

From Figure **??** we observe that Logistic Regression presents the steepest initial rise and the largest area under the curve (AUC), consistently with the results reported in the metric table. LDA and Random Forest also achieve a reasonably good separation capability, whereas KNN, QDA and Bernoulli Naive Bayes show curves closer to the diagonal, indicating a weaker discriminative performance. The Decision Tree model is almost aligned with the random baseline, confirming the limitations already highlighted in the quantitative evaluation.

7.4 Relation between models and learning dynamics

The success of tree-based models reveals important insights about the underlying cognitive and behavioral patterns:

- **Non-monotonic learning curves**: Student improvement may not be strictly linear with experience, but rather exhibit plateaus and sudden gains that trees can model through threshold-based rules.
- **Contextual ability**: A student's success probability depends on complex combinations of their general ability, topic-specific expertise, and question characteristics, requiring compositional decision rules.
- **Population heterogeneity**: Different student subgroups may follow qualitatively different learning trajectories, which Random Forest accommodates through diverse decision paths across its ensemble.

Therefore, while linear models provide interpretable baseline performance, the superior generalization of Random Forest under realistic evaluation conditions confirms that educational outcomes emerge from **complex, non-linear interactions** between learner characteristics, content difficulty, and accumulated experience.

7.5 MLP model: training behaviour and performance analysis

In addition to the classical machine learning models, we also trained a **Multilayer Perceptron (MLP)** implemented using PyTorch Lightning. The objective of this neural model is the same as in the previous experiments, namely predicting whether a student answer is correct or not starting from the engineered feature space. The model consists of two fully connected hidden layers with ReLU activation and dropout regularisation, followed by a final classification layer.

Training configuration Training was performed using the same student-level split adopted for all other models, ensuring that the MLP is evaluated only on completely unseen students. The Adam optimiser was used with binary cross-entropy loss. Training included early stopping based on validation loss to prevent overfitting.

Performance results Under the student-level partitioning methodology, the MLP achieved the following performance on the test set:

- **Validation Accuracy**: 0.598 (59.8%)
- **Validation F1-score**: 0.622
- **Validation ROC-AUC**: 0.673

Comparative analysis The MLP performs competitively but does not outperform the best classical models. Specifically:

- The ROC-AUC of 0.673 is similar to LDA (0.670), indicating comparable ranking capability.
- However, accuracy (59.8%) and F1-score (0.622) are lower than Random Forest (65.0% accuracy, 0.701 AUC).
- The MLP ranks 5th among all evaluated models, behind the four tuned classical approaches.

Interpretation The fact that the neural network does not substantially outperform classical models is both expected and informative in this context:

1. **Limited dataset size**: With only 372 total students (298 training, 74 test), the dataset is relatively small for deep learning. Neural networks typically require large amounts of data to leverage their capacity effectively, whereas tree-based models and linear classifiers are more data-efficient.
2. **High-quality engineered features**: The feature engineering pipeline already captures complex interactions (e.g., `topic_familiarity`, `ability_difficulty_gap`) and cumulative statistics. Since the input features are semantically rich and carefully designed, there is limited room for the MLP to discover additional non-linear transformations that would improve predictions.
3. **Tabular data domain**: Random Forests and Gradient Boosting are particularly well-suited for tabular data with heterogeneous features. Neural networks excel in domains like computer vision, natural language processing, or sequential modeling, where raw inputs benefit from learned representations. In our case, the structured, pre-aggregated nature of the features favors tree-based methods.
4. **Generalization to new students**: The student-level split creates a challenging generalization scenario. With relatively few test students, the MLP's higher capacity (more parameters) makes it more susceptible to overfitting to training student patterns that do not transfer to the test population. Random Forest's ensemble of simpler trees provides better regularization in this low-data regime.

Validation of feature engineering Paradoxically, the MLP's inability to significantly outperform simpler models is a **positive signal** about the quality of the feature engineering:

- It confirms that the engineered features already capture the essential predictive patterns in the data.
- The MLP cannot "rescue" weak features by learning complex transformations, because the features are already well-designed.
- This validates the interpretability of the approach: the predictions arise from transparent, human-understandable features rather than from opaque learned representations.

Overall assessment The MLP serves as an important **methodological validation**: it demonstrates that increasing model complexity does not automatically improve performance when the feature space is already rich and the dataset size is limited. This result reinforces the conclusion that, for knowledge tracing on this dataset, carefully engineered features combined with appropriate classical models (Random Forest, Gradient Boosting) constitute the most effective approach. The neural network confirms rather than contradicts the insights from the tree-based models, while highlighting the importance of matching model complexity to dataset characteristics.

7.6 Strengths and limits of the modelling approach

The obtained results are coherent and informative, but some limitations remain:

- the prediction is static and does not explicitly model temporal sequences at interaction level;
- latent student abilities are approximated through engineered aggregates rather than estimated explicitly;
- the current models do not incorporate uncertainty in student ability evolution.

Despite these aspects, the present approach already provides a meaningful and interpretable representation of student performance behaviour, while preserving a strong connection with the evidence highlighted in the exploratory analysis.

Future extensions may include sequential models, but the results obtained here demonstrate that a structured feature engineering pipeline combined with linear modelling constitutes a solid and informative baseline.

7.7 Roc curves

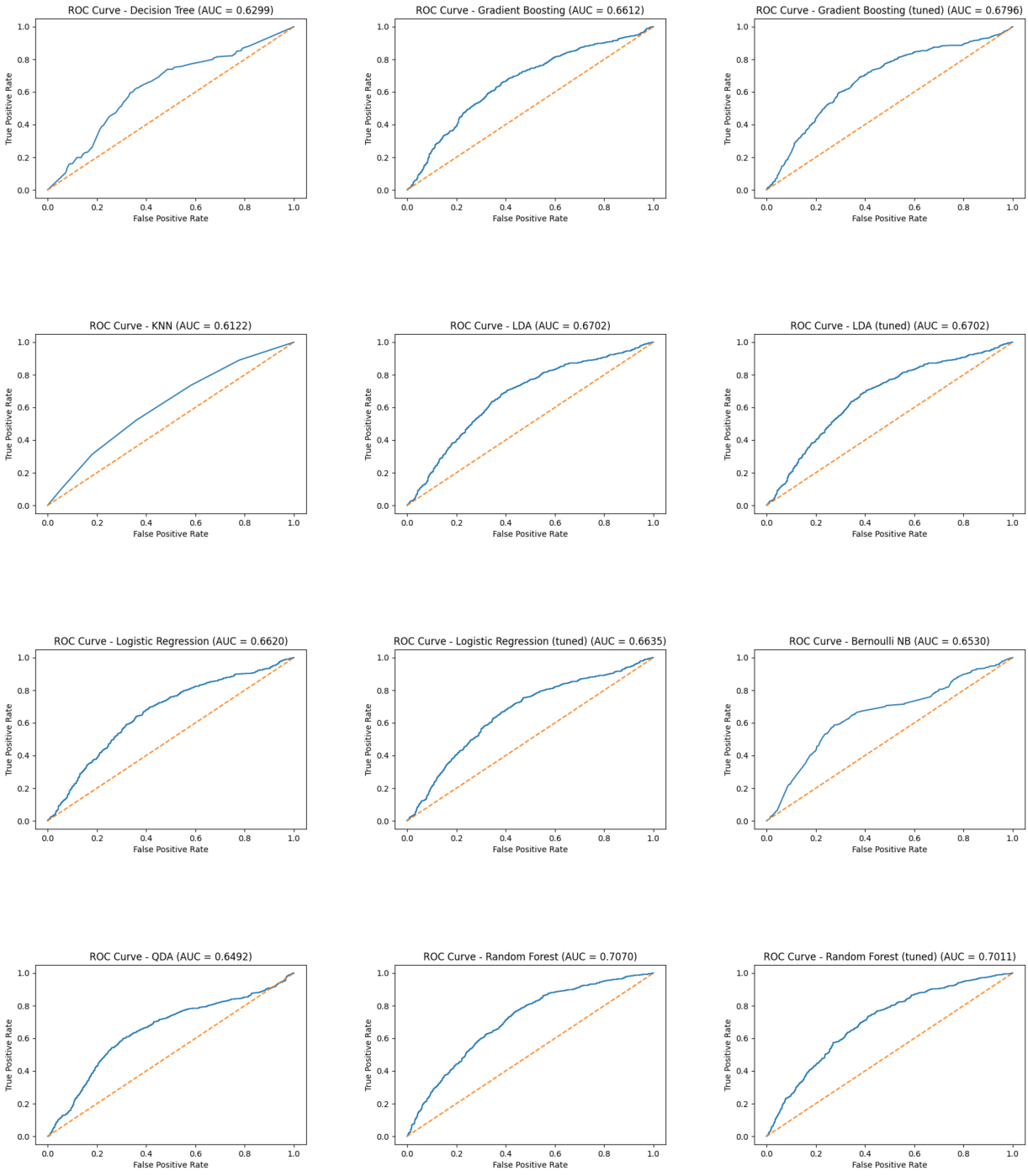


Figura 1: Individual ROC curves

7.8 Confusion Matrix

Figura 2: Individual confusion matrix