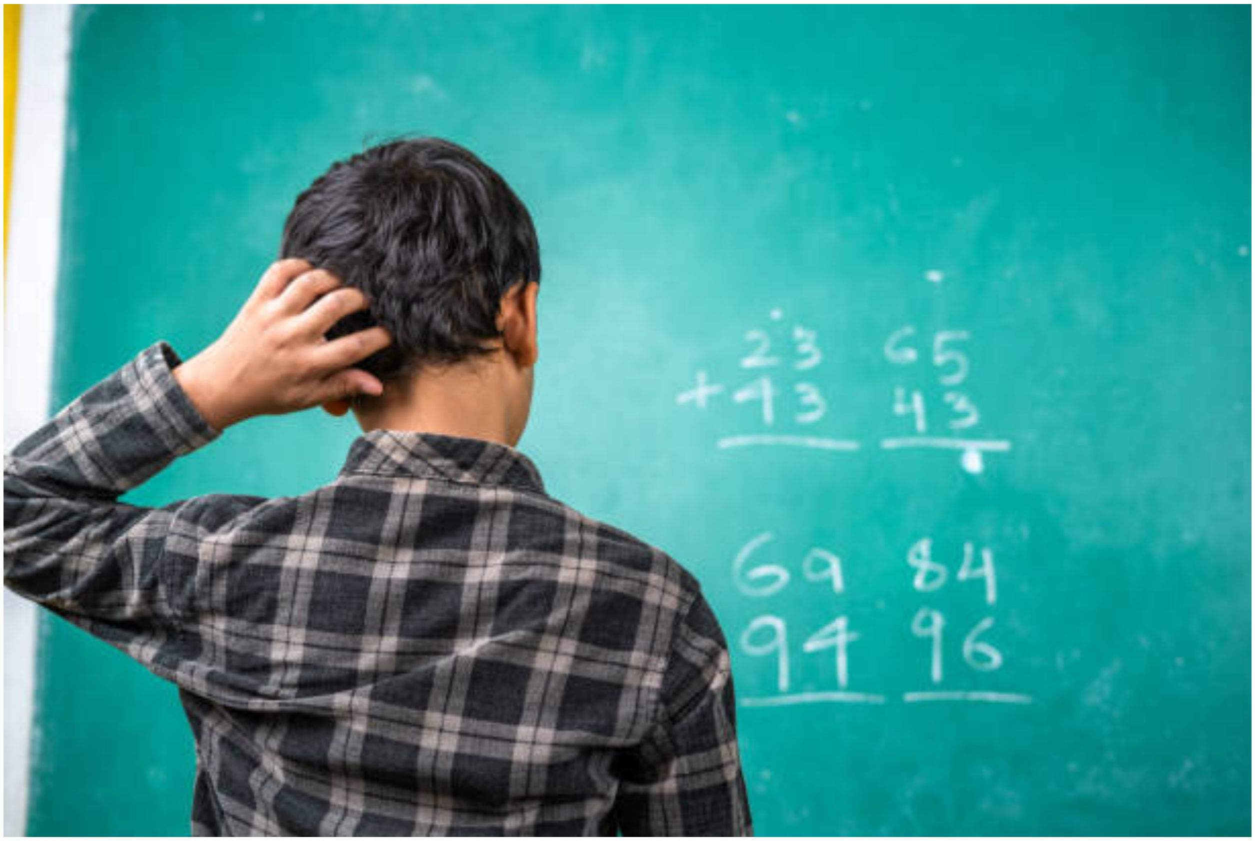
Capstone Final Report

NLP Model to Identify Math Student Misconceptions



Georgia Jenkins

October 2025

Background

Students' written explanations of their mathematical reasoning provide valuable insight into their thinking and can reveal systematic misconceptions, such as applying whole-number logic to decimals (e.g., reasoning that 0.355 is greater than 0.8 because 355 > 8). These misconceptions arise when students misapply prior knowledge or misunderstand new concepts, and while tagging such responses is useful for diagnostic feedback, the process is labor-intensive and difficult to scale. The Misconception Annotation Project (MAP), a Kaggle competition hosted by Vanderbilt University and The Learning Agency, was designed to address this challenge by encouraging the development of Natural Language Processing (NLP) models capable of identifying student misconceptions across diverse math problems. By leveraging machine learning to automate this process, the project aims to predict when a student's answer contains a misconception, and what type of misconception they have, thus improving feedback and supporting more effective learning experiences for students.

Dataset Profile

Eedi is a London-based online learning platform where students practice math problems, and tutors identify student misconceptions. Students answer multiple choice questions with four possible options, only one of which is correct. After selecting their multiple choice, they are prompted to provide a written explanation justifying their answer. The multiple choice questions, answers, and explanations are included in the MAP dataset.The MAP dataset has 36,696 rows and 6 variable columns.

Variable Name	**Description**
QuestionId	Categorical – 15 5-digit IDs corresponding to each QuestionText, no NA values
QuestionText	Categorical – 15 math questions, no NA values
MC_Answer	4 possible responses per Question, no NA values
StudentExplanation	Free text, no NA values
Category	Categorical – 6 labels describing the MC_Answer and StudentExplanation, no NA values • True_Correct: MC_Answer is true, Explanation is correct • True_Misconception: MC_Answer is true, Explanation contains a misconception • True_Neither: MC_Answer is true, Explanation is neither correct nor incorrect • False_Correct: MC_Answer is false, Explanation is correct • False_Misconception: MC_Answer is false, Explanation contains a misconception • False_Neither: MC_Answer is false, Explanation is neither correct nor incorrect
Misconception	Categorical - 36 misconception types (e.g., wrong_fraction, whole_number_bigger) Only used if Category is True_Misconception or False_Misconception, otherwise NA

Table 1: Description of variables in the MAP dataset.

Exploratory Data Analysis and Preprocessing

Categorical Variables

Starting with the categorical variables (QuestionID will be used synonymously with QuestionText throughout the analysis), one observes class imbalance in all three variables. Some questions appear more often than others, and 'True_Correct' is the most common Category. Unsurprisingly, 'True_Misconception' and 'False_Correct' are the least common categories, since it is rare for a student to answer correctly without understanding the concept, or to answer incorrectly with no misconceptions. The most extreme class imbalance, however, came from the type of misconception, with 'incomplete' being the most common.

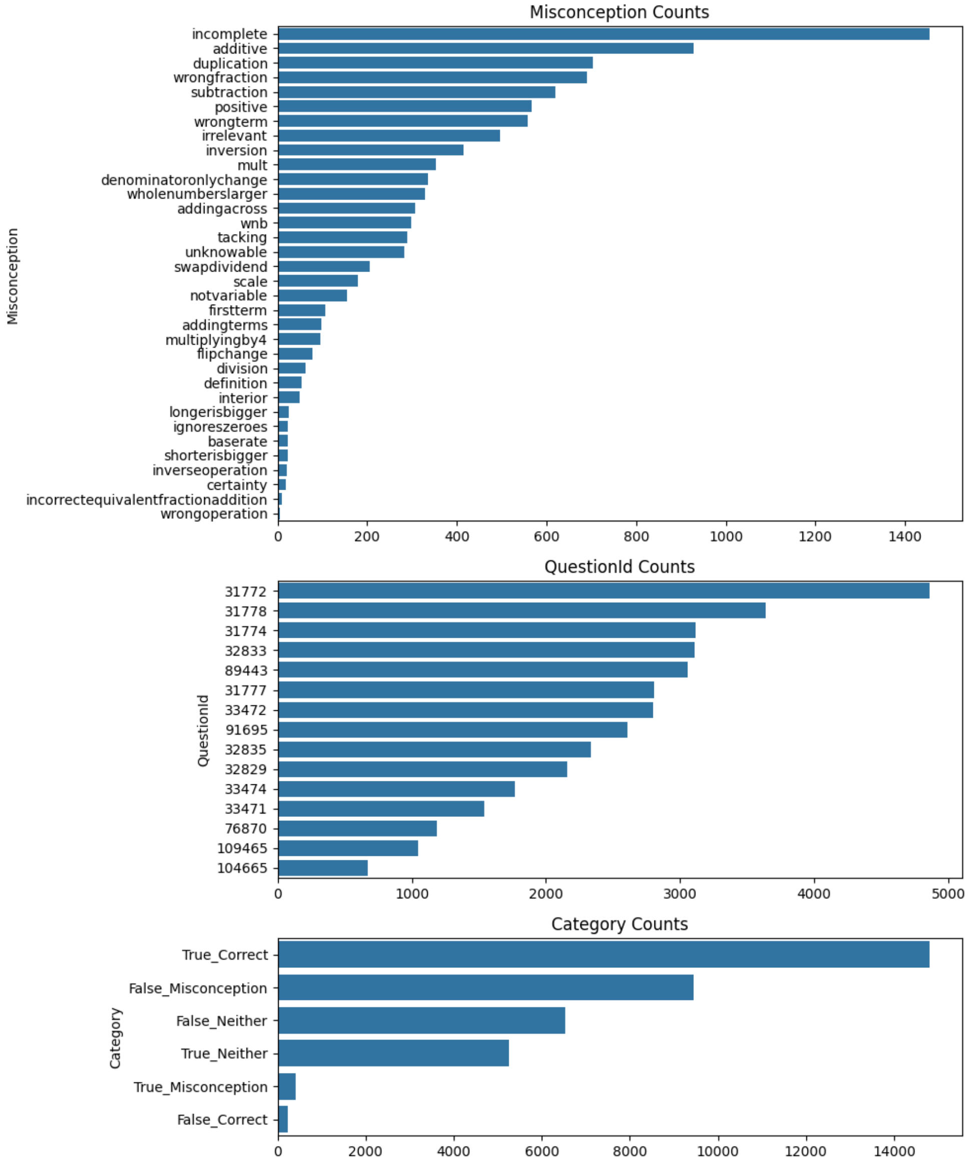


Figure 1: class imbalance can be observed in all categorical variables of the dataset.

After cleaning the Misconception variable (removed spaces, special characters, and set all to lower), there were 36 classes in this feature, with several making up less than 1% of records in the MAP dataset. Considering the challenges this would pose during modeling, the misconceptions were combined logically into 11 classes.

New Class	**Contents from Old Classes**
incomplete	incomplete
addition	additive, addingterms, addingacross
procedure	duplication, irrelevant, flipchange, ignoreszeroes
fraction	wrongfraction, denominatoronlychange, incorrectequivalentfractionaddition,swapdividend
variable	notvariable, wrongterm, firstterm, tacking
wholenumber	wholenumberbias, wnb, shorterisbigger, wholenumberslarger, longerisbigger
subtraction	subtraction
definition	definition, baserate, scale, certainty, interior, unknowable
positive	positive
operation	wrongoperation, inverseoperation, inversion, division
multiplication	multiplyingby4, mult,

Table 2: mapping 36 old Misconception classes to the 11 new classes.

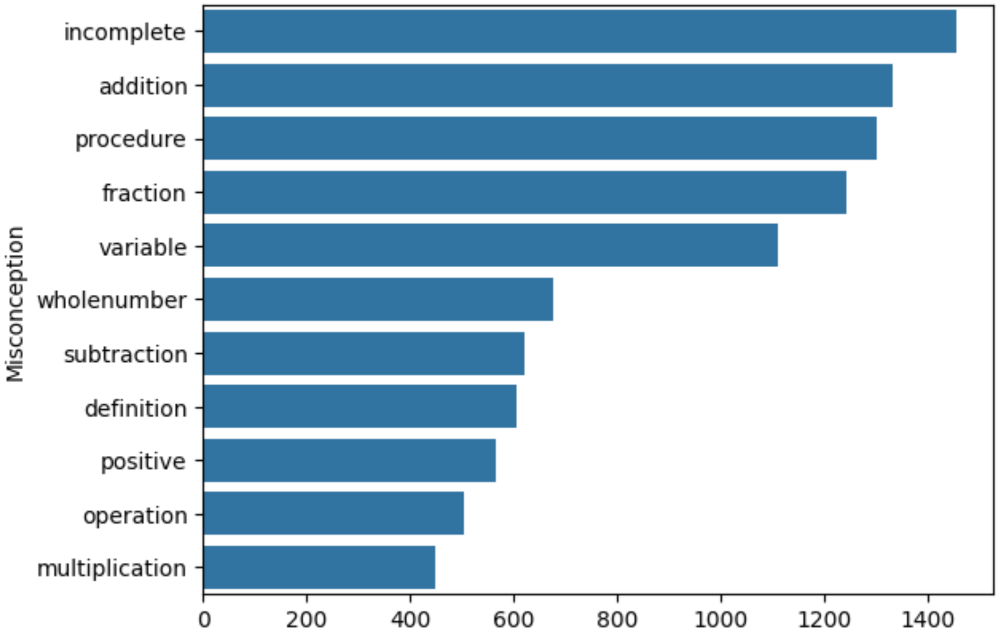


Figure 2: new distribution of Misconception classes.

The next step was to the Category target variable from six classes into three. Machine learning algorithms are not necessary to tell us if a multiple choice question is correct. In educational environments, it is safe to assume that an answer key exists. So a new binary feature, is_correct, was added to tell us whether the student answered the multiple choice question correctly, and Category was simplified into NewCategory, containing the classes, 'Correct,' 'Misconception', and 'Neither.'

StudentExplanation Free Text Variable

The free text variable, StudentExplanation, was characterized by responses with fewer than 25 words or 100 characters.

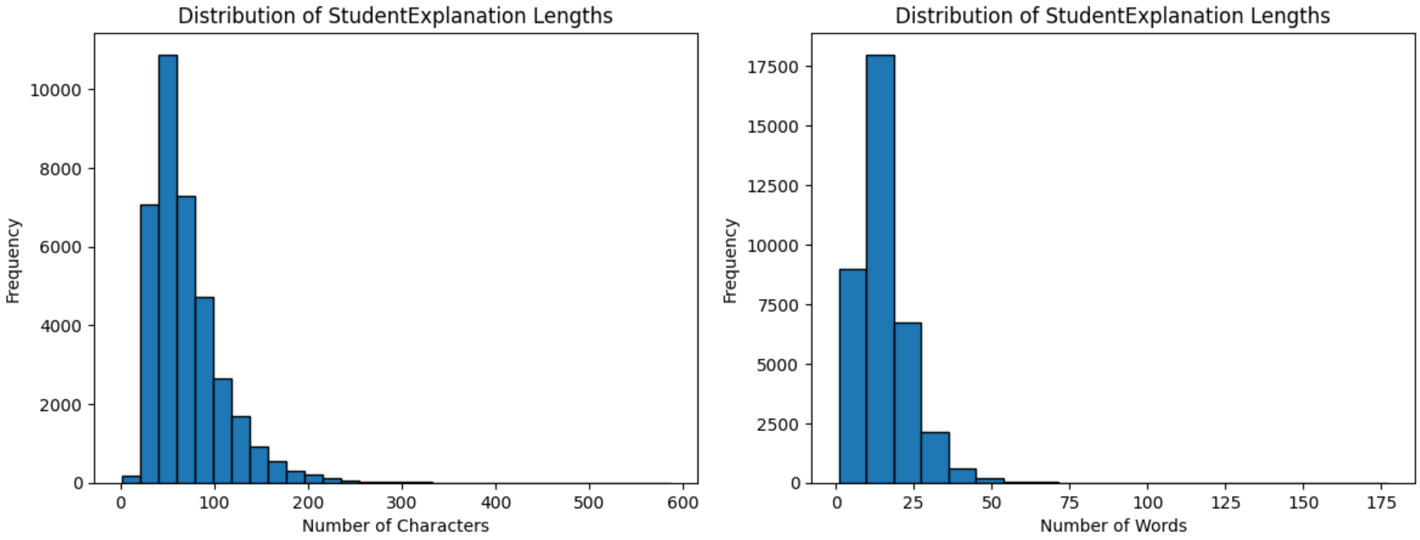


Figure 3: distributions of StudentExplanation lengths.

The language used in StudentExplanation was informal, containing many contractions. These first steps were taken to clean the text data:

- Set all to lower
- Remove accents
- Expand contractions
- Remove repeated words
- Remove English stop words

StudentExplanation also contained many spelling errors and casual speech. Thus, TokTokTokenzier was used for it's advantage with user-generated text data. After tokenizing, it was clear that there many special characters and numbers pertaining to the math problems. At first, one might consider keeping the special characters and numbers because of their relevance to the topic. However, these characters dominated the most common tokens and were not very insightful when it came to the sentiment of the explanation. Thus, special characters and numbers were removed, and the top 20 tokens were revealed.

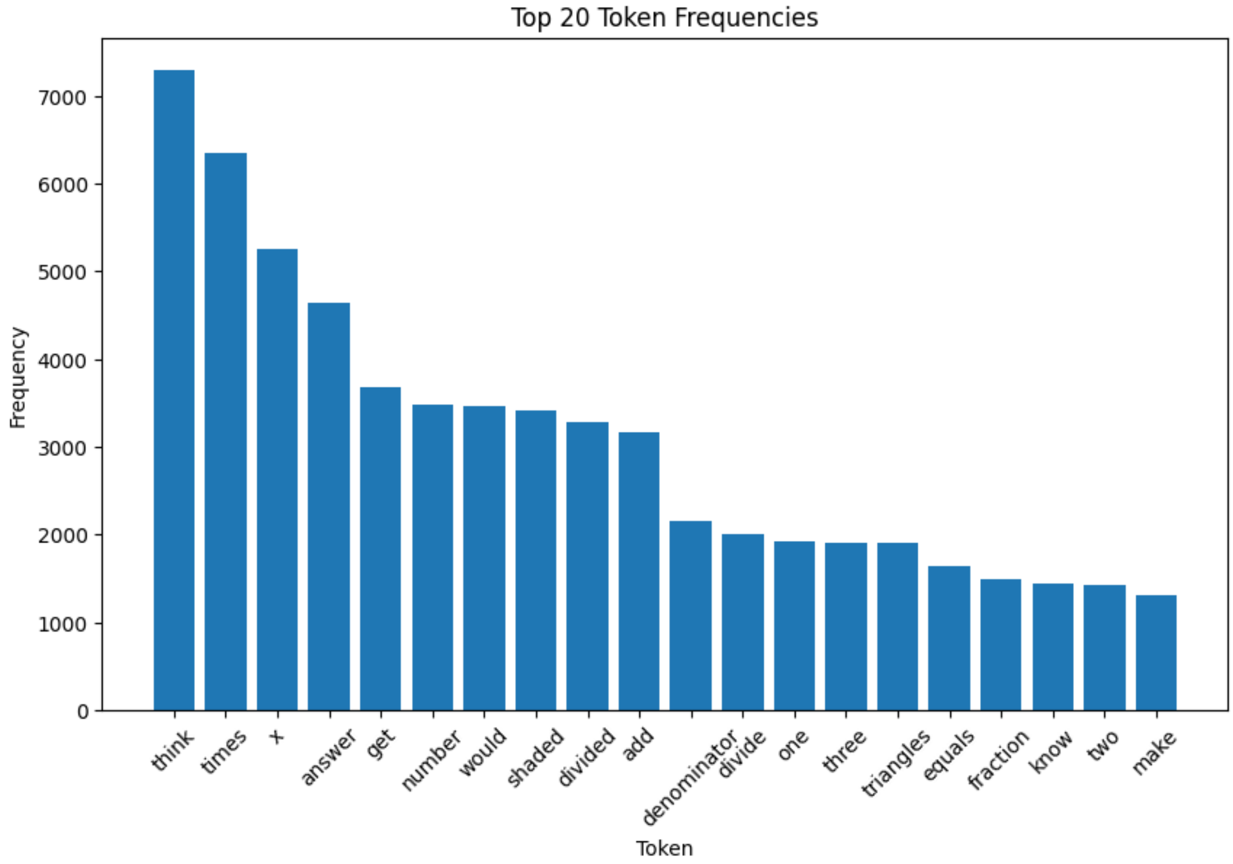


Figure 4: Top 20 tokens from the preprocessed StudentExplanation. The tokenized text was also passed through a WordNetLemmatizer, but the difference between these two outputs was minimal.

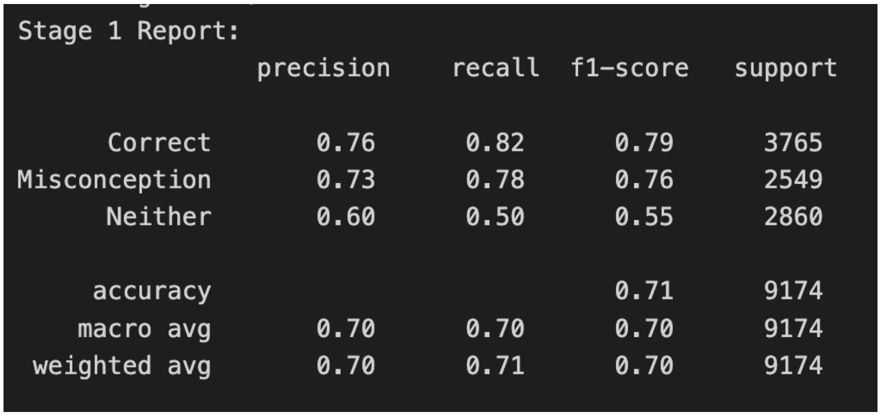
Modeling

Two-Step Categorization with Multinomial Naive Bayes

This model is trying to predict two targets:

1. Whether the StudentExplanation was correct, had a misconception, or neither
2. If there is a misconception, what type of misconception is it?

The baseline model used was a Multinomial Naive Bayes with a TF-IDF Vectorizer and TokTokTokenizer for the free text explanation. The other independent features were the binary is_correct variable and the QuestionID. The baseline model was only focused on predicting the first half of the problem: whether the StudentExplanation had a misconception, was correct, or neither. The model performance was fair, slightly outperforming a similar model that was predicting the original 6 categories (‘True_Correct,’ ‘True_Misconception,’ etc.). Still class imbalance is the target variable was holding the model back, so a RandomOverSampler was introduced. Then a GridSearchCV revealed the best alpha value to be 0.01, and tuning the TF-IDF Vectorizer to use ngram_range = (1,2), max_features=50000, and sublinear_tf=True led to these baseline results.



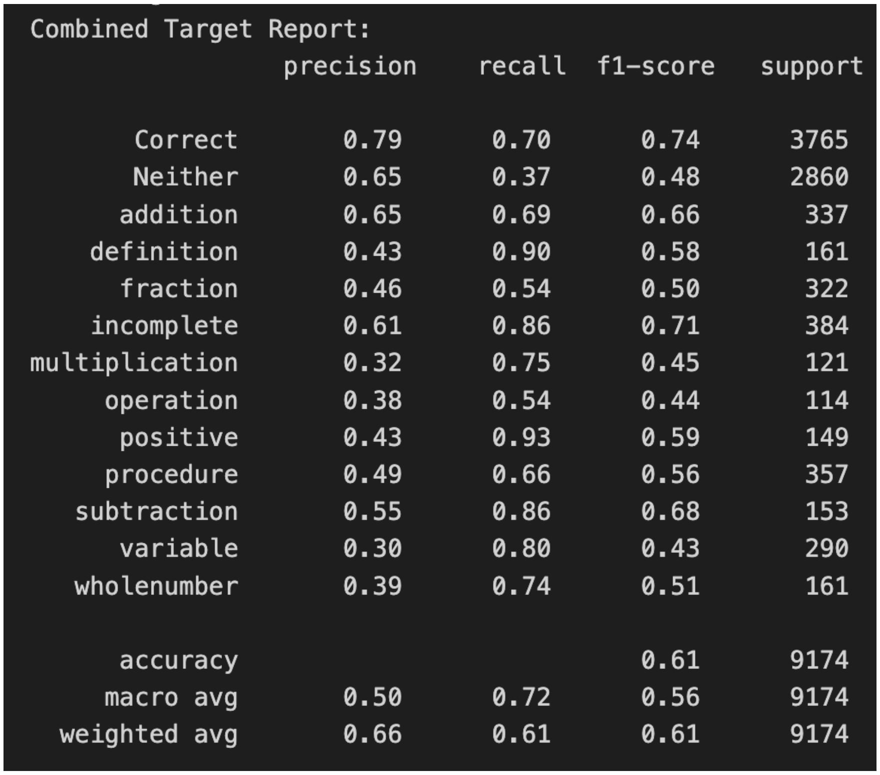
```
Stage 1 Report:
                precision    recall  f1-score   support

      Correct       0.76      0.82      0.79      3765
Misconception       0.73      0.78      0.76      2549
      Neither       0.60      0.50      0.55      2860

     accuracy                           0.71      9174
    macro avg       0.70      0.70      0.70      9174
 weighted avg       0.70      0.71      0.70      9174
```

Figure 5: results of baseline Multinomial Naives Bayes model on stage 1 of the two-step categorization problem.

For stage 2, predicting the type of misconception, only the rows with a misconception predicted in the test set were selected. However, given the small size of this sub dataset and the 11 classes of misconceptions, results were very poor (F1 = 0.00). Predicting the misconception type from the full dataset resulted in strong results (F1 = 0.80), but this was not a realistic representation of the problem that needed to be solved. Using the full dataset allowed the model to “cheat” because it received data that was properly labelled as having a misconception, something that would not be available in practice.

Combined Target Categorization with Multinomial Naive Bayes

To address this problem, a new target variable was created called CombinedTarget. CombinedTarget contained the ‘Correct’ and ‘Neither’ classes from NewCategory, but replaced ‘Misconception’ with the type of misconception that was identified, resulting in 13 classes and only one target to predict. Using a Multinomial Naive Bayes with the same parameters as the previous models yielded promising results.



```
Combined Target Report:
                 precision    recall  f1-score   support

       Correct       0.79      0.70      0.74      3765
       Neither       0.65      0.37      0.48      2860
      addition       0.65      0.69      0.66       337
    definition       0.43      0.90      0.58       161
      fraction       0.46      0.54      0.50       322
    incomplete       0.61      0.86      0.71       384
multiplication       0.32      0.75      0.45       121
     operation       0.38      0.54      0.44       114
      positive       0.43      0.93      0.59       149
     procedure       0.49      0.66      0.56       357
   subtraction       0.55      0.86      0.68       153
      variable       0.30      0.80      0.43       290
   wholenumber       0.39      0.74      0.51       161

      accuracy                           0.61      9174
     macro avg       0.50      0.72      0.56      9174
  weighted avg       0.66      0.61      0.61      9174
```

Figure 6: results of Multinomial Naives Bayes model using CombinedTarget.

Combined Target Categorization with Light Gradient-Boosting Machine

To predict many classes, several other models were tested. RandomForests, transformers, and neural networks all struggled to outperform the Multinomial Naive Bayes. The Light Gradient Boosting Machine (LightGBM) proved the most successful, finally besting the Multinomial Naive Bayes. After running an Optuna hyperparameter optimization framework, the final LightGBM model setup differed slightly from the Naive Bayes:

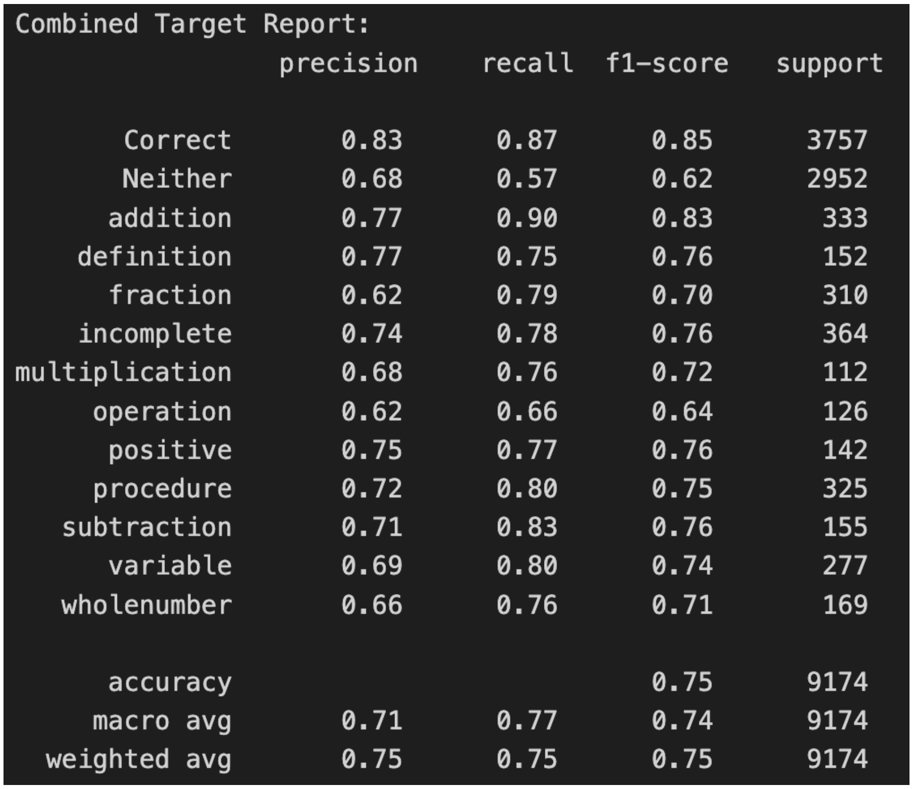
- TF-IDF ngram_range set to (1,3) instead of (1,2)
- Encoded QuestionID and is_correct and converted to a Compressed Sparse Row (CSR) matrix

The final model resulted in an overall F1 score = 0.75, weighted average recall = 0.77, and weighted average precision = 0.75. The lowest performing class was ‘Neither’ with an F1 score = 0.62, and the best performing class was ‘Correct’ with F1 = 0.85.

Final Model Summary and Results

1. Train test split with 75/25 ratio and stratified target variable
2. Text preprocessing with TF-IDF and ToktokTokenizer on PreprocessedExplanation
3. Encode categorical QuestionId with LabelEncoder
4. CSR matrix on encoded QuestionId and binary is_correct
5. Combine preprocessed features into final X_train and X_test

6. Handle class imbalance with RandomOverSampler
7. Light Gradient-Boosting Machine with boosting_type="gbdt", objective="multiclass", class_weight="balanced", n_estimators=500, learning_rate=0.05, max_depth=-1, n_jobs=-1
8. For reproducibility, random_state=42 was used throughout this report.



```
Combined Target Report:
                 precision    recall  f1-score   support

       Correct       0.83      0.87      0.85      3757
       Neither       0.68      0.57      0.62      2952
      addition       0.77      0.90      0.83       333
    definition       0.77      0.75      0.76       152
      fraction       0.62      0.79      0.70       310
    incomplete       0.74      0.78      0.76       364
multiplication       0.68      0.76      0.72       112
     operation       0.62      0.66      0.64       126
      positive       0.75      0.77      0.76       142
     procedure       0.72      0.80      0.75       325
   subtraction       0.71      0.83      0.76       155
      variable       0.69      0.80      0.74       277
   wholenumber       0.66      0.76      0.71       169

      accuracy                           0.75      9174
     macro avg       0.71      0.77      0.74      9174
  weighted avg       0.75      0.75      0.75      9174
```

Figure 7: results of final LightGBM model.