Multilabel Classification in Large-Scale Text Datasets: A Data-Driven and Model-Driven Approach

Julius Maliwat[a]

[a]*University of Milano-Bicocca, Data Science*

Abstract

This study evaluates two distinct objectives in multilabel classification on a large-scale text dataset. The first objective is to compare the predictive performance and computational efficiency of a data-driven approach using lightweight models, such as Logistic Regression trained on the full training set, with a model-driven approach employing transformer-based architectures, like DistilBERT, trained on a reduced version of the training set. The second objective is to examine the impact of label grouping on the performance of lightweight models. Logistic Regression with a threshold of 0.2 achieved the best results, with a macro F1 score of 0.59 and a weighted F1 score of 0.68 on the test set, outperforming DistilBERT, which was limited by higher computational costs and the need for a reduced dataset. Additionally, applying label grouping to lightweight models further improved performance, with Logistic Regression achieving a macro F1 score of 0.76 and a weighted F1 score of 0.79 on the test set.

Contents

1. Introduction

Effectively managing and classifying large-scale textual data is crucial across various industries, from content recommendation systems to automated document categorization. This project focuses on addressing the challenge of multilabel classification in a large dataset where each document can cover multiple topics (1). The ability to accurately assign multiple labels to documents has wide-ranging applications, such as improving search engine relevance, enhancing content personalization, or streamlining information retrieval in complex datasets.

The study aims to explore and compare two main approaches to tackle this problem. The first approach, data-driven, uses lightweight machine learning models such as Logistic Regression and Naive Bayes, applied to the full training set. These models are designed to quickly process large volumes of data while maintaining a balance between accuracy and computational efficiency. The second approach, model-driven, employs more advanced deep learning architectures, such as transformer-based models like DistilBERT, trained on a reduced version of the training set due to computational limitations. This approach aims to capture deeper, more complex relationships within the data, albeit with higher computational costs.

The goal of this experiment is to determine which approach performs better for our specific classification task, considering both predictive performance and the computational efficiency required to handle large datasets. Additionally, we explore whether simplifying the classification problem by grouping similar labels can improve overall model performance. By reducing the complexity of the label space, we aim to evaluate whether this strategy enhances model outcomes without sacrificing the essential detail of each category.

2. Dataset

2.1. Overview

The dataset used in this study is a comprehensive collection of 200,005 articles and documents, each annotated with multiple topics. The dataset consists of 5 key columns, including:

- **id**: A unique numerical identifier for each document.
- **title**: The title of the document, providing a brief description of its content.

- **topic**: A general categorization of the document, marked as "Mixed" for articles covering multiple themes. There are 39 distinct topics excluding "Mixed".
- **topics_with_percentages**: A serialized dictionary showing the percentage distribution of different topics covered in the article. This column is particularly important for our multilabel classification task as it provides a detailed breakdown of how much of each topic is covered in the document.
- **text**: The full textual content of the document, which serves as the primary input for our classification models.

2.2. Topic Distribution

The dataset analysis reveals that 80% of the documents are labeled as "Mixed" in the **topic** column, indicating that they contain multiple topics, while the remaining 20% are labeled with a specific single topic (e.g., "Politics," "Military," "Sports"). For this analysis, all documents categorized under a single topic were grouped together under the label "Single." This significant prevalence of mixed labels, as shown in Figure 1, highlights the necessity for multilabel classification techniques capable of effectively managing documents with diverse topics.

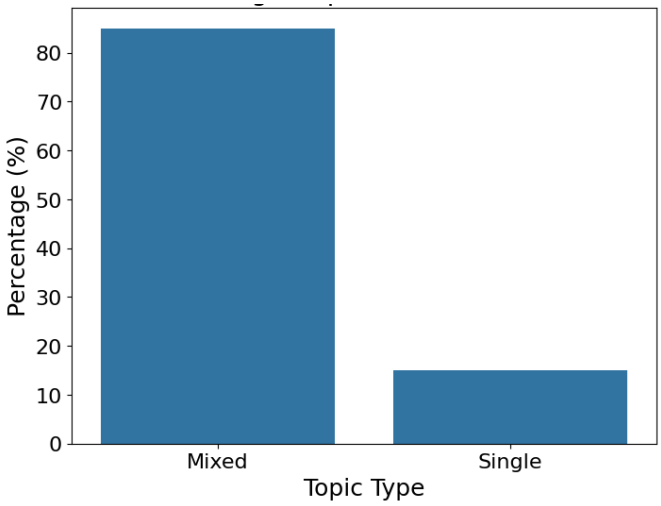

Figure 1: Distribution of documents by topic type. Mixed topics dominate with 80%, indicating the complexity and multilabel nature of the dataset, compared to 20% for single topics.

Since most documents are associated with more than one topic, we rely on the **topics_with_percentages** column, which provides the specific breakdown of topics within each document. This column is crucial for accurately representing and classifying the complex content of these documents. The detailed methodology for processing and utilizing this column in our classification models will be further elaborated in Section 3

3. Methodologies

In this section, we explore the various methodologies utilized to tackle the multilabel classification problem. This includes a detailed examination of the preprocessing steps, the metrics used for evaluation, the validation methods applied, and the models implemented.

We provide a brief overview of the theoretical foundations of each method, discussing their relevance and appropriateness for the task at hand. The selection of these specific methods is driven by the nature of the dataset, the need for computational efficiency, and the objective of optimizing predictive performance in classifying the multilabel data. The rationale behind each choice is elaborated to give a clear understanding of how these methodologies contribute to the overall effectiveness of the model.

3.1. Preprocessing

In this subsection, we describe the primary methods employed to prepare and structure the dataset for the training phase of our multilabel classification task. Preprocessing is a critical step in any machine learning pipeline, particularly in the context of multilabel classification, where the quality of input data significantly impacts the performance of the models. Here, we outline the procedures for defining labels, removing outliers, grouping labels, preprocessing text, and applying downsampling techniques. Each of these steps is designed to optimize the dataset for effective and efficient model training.

3.1.1. Labels Definition

In this step, we focused on extracting and defining the labels from the `topics_with_percentages` column. This column is structured as a JSON-like object where each key corresponds to a topic and the value represents the percentage of the document that covers that topic.

To prepare the dataset for multilabel classification, we transformed each key (topic) into a separate column, simplifying the process of analysis and modeling. Next, we applied a thresholding technique to convert the percentage values into binary labels. Specifically, if a topic covered more than 10% of a document, we marked the corresponding column with a `1`, indicating the presence of that topic. If the topic coverage was less than or equal to 10%, we marked it with a `0`, indicating its absence.

The choice of the 10% threshold was informed by an analysis of the topic distribution in documents labeled as "Mixed." We observed that these documents typically had at least two topics with a coverage greater than 10%, reinforcing our decision to set the threshold at this level.

Distribution of Topics Per Document. To better understand the complexity of the dataset and the nature of the documents, we analyzed the number of topics assigned to each document after processing the `topics_with_percentages` column.

As illustrated in Figure 2, the majority of documents are associated with multiple topics, with 33% containing two topics, followed by 26% with three topics. Documents with a single topic account for 18%, while the remaining instances contain a higher number of topics, with decreasing frequency.

This distribution highlights the inherent complexity of the dataset, where a significant portion of the documents contains

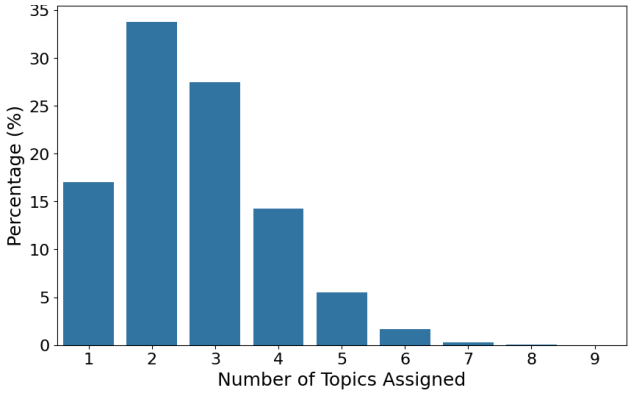

Figure 2: Relative Distribution of the Number of Topics per Document in the Dataset. This bar chart illustrates that documents with 1 to 4 topics cover the majority, with a peak at 2 topics (over 30%). The frequency declines as the number of topics increases, indicating a lower prevalence of documents with more than 5 topics.

multiple topics, reinforcing the need for multilabel classification. The model must be able to manage documents with various degrees of topic diversity, ranging from single-topic articles to those with several intertwined themes.

Topic Distribution Analysis. Following the binarization process, we conducted an analysis of the topic distribution to identify any imbalances and assess their potential impact on the multilabel classification task.

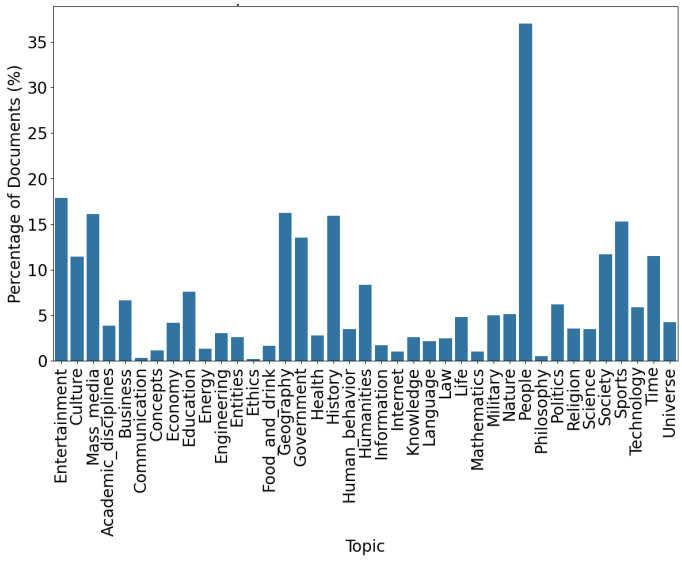

Figure 3: Relative Distribution of Topics Across Documents. The 'People' topic dominates the dataset, accounting for approximately 35% of all documents, followed by 'Entertainment' at 18%. Other topics, such as 'Ethics' and 'Communication', are significantly underrepresented, each appearing in less than 5% of the documents. This imbalance highlights potential challenges in training models that generalize well across both frequent and rare topics.

Figure 3 illustrates the relative frequency of topics across the dataset. A clear imbalance is observed, with the *People* topic dominating at approximately 35%, followed by *Entertainment* (about 18%). In contrast, topics like *Ethics* and *Communication* are significantly underrepresented, each appearing in fewer than 5% of the documents.

Such imbalances can introduce challenges for model performance, as the model may become biased towards the more prevalent topics, which could affect its ability to generalize well across the less frequent labels.

3.1.2. Label Grouping

To address the complexity inherent in the multilabel classification task, we grouped the original topics into broader macro-topics. This strategy aimed to reduce the dimensionality of the problem and evaluate whether working with more abstract categories could enhance model performance.

The macro-topics were defined by clustering related topics from the original dataset into larger, more general groups, as outlined below:

- **Culture and Society:** Culture, Society, Human Behavior, Religion, Philosophy, Ethics, People.
- **Academic Disciplines:** Academic Disciplines, Humanities, History, Education, Language, Law, Concepts, Time.
- **STEM:** Mathematics, Technology, Science, Engineering, Internet, Information, Knowledge.
- **Economics:** Economy, Business, Communication.
- **Health:** Health, Food and Drink, Life.
- **Entertainment:** Entertainment, Sports, Mass Media.
- **Environment:** Geography, Nature, Energy, Universe.
- **Government:** Government, Politics, Military, Entities.

As illustrated in Figure 4, the dataset exhibits a clear imbalance in the distribution of these grouped topics, although the disparity is less pronounced compared to the original, more granular topics.

The most prominent macro-topic is **Culture and Society**, which accounts for nearly 60% of the dataset. This is followed by **Academic Disciplines**, representing roughly 35%, and **Entertainment**, which encompasses about 35% of the documents. Other macro-topics, such as **Health**, **STEM**, and **Economics**, are underrepresented, each accounting for less than 15% of the total dataset.

This uneven distribution suggests that despite reducing the complexity of the classification problem by grouping related topics, a significant imbalance persists. The dominance of the **Culture and Society** category may still introduce biases in the model's performance, as models tend to favor more frequent classes during training. However, this approach has nonetheless mitigated some of the extreme disparities present in the original distribution, making the task more manageable and reducing the risk of overfitting to minority classes.

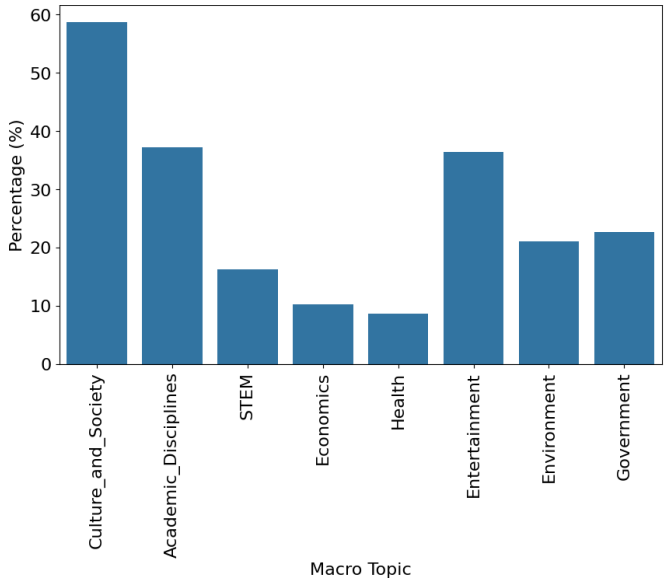

Figure 4: Refined Grouped Topic Distribution (Relative Frequency). This bar chart showcases the distribution of aggregated topics within the dataset. 'Culture and Society' emerges as the most dominant category, comprising over 50% of the documents. Conversely, 'Economics' and 'Health' are markedly underrepresented, each accounting for less than 10% of the documents.

3.1.3. Text Preprocessing

Effective text preprocessing is crucial for improving the performance of machine learning models in natural language processing (NLP) tasks (2). In this section, we outline the preprocessing pipeline applied to the textual data in our dataset, focusing on stopword removal, text cleaning, and lemmatization. These steps were designed to enhance the quality of the input data, reduce noise, and ensure that the models are able to focus on the most informative features of the text.

Stopword Removal. Stopwords, such as *"the," "and,"* and *"in,"* are frequently occurring words in any language that do not contribute significantly to the meaning of the text. In our preprocessing, we utilized a predefined list of stopwords to filter out these common but insignificant words, thereby improving the quality of the dataset.

Text Cleaning. Raw text data often contains noise in the form of HTML tags, URLs, special characters, and other irrelevant symbols. This noise can negatively affect model performance. To address this, we applied a series of text cleaning steps. These included removing HTML tags, URLs, special characters, and converting all text to lowercase.

Lemmatization. Lemmatization involves converting words to their base or dictionary form, known as the lemma. This step helps reduce inflected forms of a word, such as "running," "ran," and "runs," to a single form, "run." By lemmatizing the text, we reduced the vocabulary size and ensured that the model captures the core meaning of the content without being distracted by word variations.

3.1.4. Outliers Removal

Outlier removal is a key step in preparing textual data for machine learning models, particularly in cases where text length varies significantly across documents. In this dataset, outliers refer to documents with exceptionally long texts that could distort the learning process. Figure 5 illustrates the distribution of text lengths before outlier removal. The majority of documents have short text lengths, but a few outliers exceed 20,000 words.

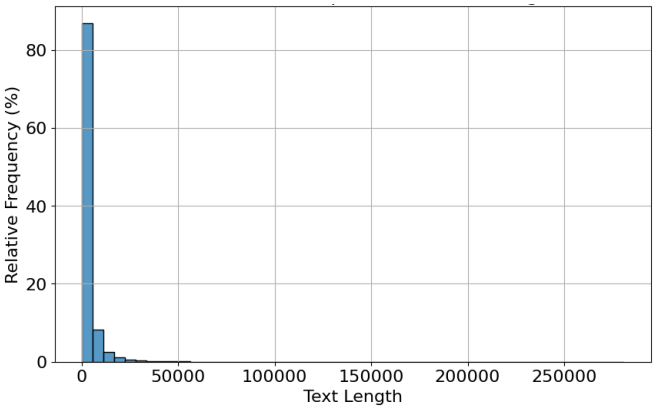

Figure 5: Distribution of Text Lengths Before Outlier Removal. The distribution shows that 80% of the documents have a text length of 4,002 characters or less, with the 95th percentile reaching 11,203 characters. A small portion of documents exceed this threshold, highlighting the presence of outliers that can introduce bias and negatively impact model training.

These long documents introduce several challenges. First, they disproportionately increase the computational load, leading to longer training times and higher memory consumption. Additionally, such outliers may dominate the learning process, causing the model to focus disproportionately on these few lengthy texts. This introduces bias, as the model may prioritize the specific patterns of the long documents instead of learning generalized representations from the overall dataset. Furthermore, the presence of these outliers can result in an imbalance, with longer documents contributing more features than shorter ones, skewing the training data.

To mitigate these issues, we removed the documents with text lengths exceeding the 90th percentile. At this threshold, the maximum text length is 6,950 words, which is more manageable compared to outlier documents that had lengths exceeding 20,000 words. This threshold was selected to retain the majority of the dataset while excluding extreme cases that could negatively impact the training process.

The effect of this removal is shown in Figure 6, where the distribution is now more concentrated, and the extreme text lengths have been eliminated.

By excluding these outliers, we improved the efficiency and performance of the training process, ensuring that the model was trained on a more representative subset of the dataset, free from the distortions introduced by excessively long documents.

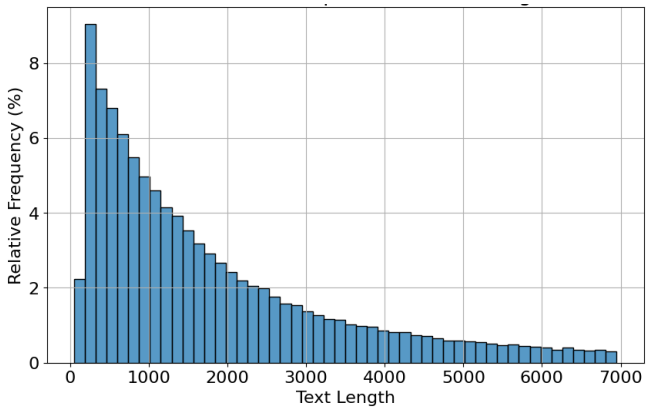

Figure 6: *Distribution of Preprocessed Text Lengths After Outlier Removal.* This histogram illustrates the refined distribution of text lengths, with a more uniform spread across lengths up to 7,000 characters, emphasizing a significant reduction in extreme values.

3.1.5. Downsampling

To address the class imbalance and improve computational efficiency, we applied downsampling on the training set, reducing the number of instances for overrepresented classes (3). The original training set contained 108,001 observations, which was reduced to 8,346 observations after downsampling. This technique allowed us to create a more balanced dataset, as illustrated in Figure 7.

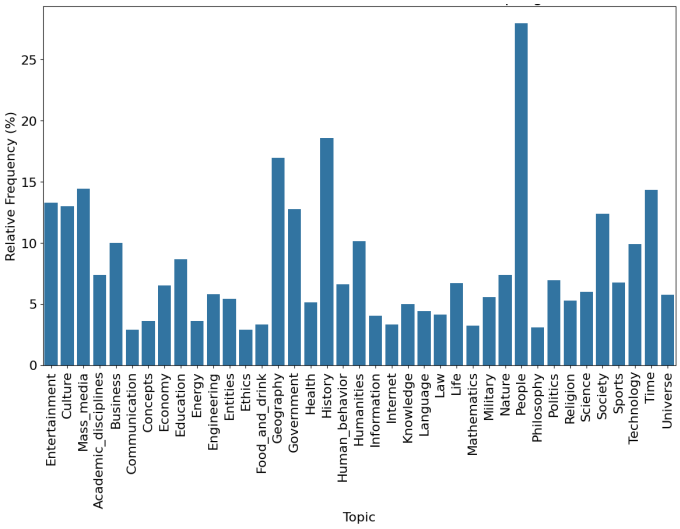

Figure 7: Relative Label Distribution After Downsampling. The downsampling process has created a more balanced label distribution, reducing the prevalence of overrepresented classes. Notably, the 'Ethics' class, previously one of the least represented, has seen a relative increase in its representation, contributing to a more equitable training scenario for the models.

The downsampling process involved reducing the number of instances in each class to match the size of the smallest class, which was **Ethics**, the rarest class in the dataset. By doing so, we ensured that all classes were equally represented in the downsampled training set, while maintaining the multilabel nature of the problem.

However, due to the complexity of multilabel classification, achieving a perfectly balanced distribution is more challenging compared to multiclass classification. Since documents can be associated with multiple labels, this interdependence between topics makes it difficult to achieve uniformity. Nonetheless, the downsampling process successfully mitigated the extreme imbalances, as shown in Figure 7, resulting in a more equitable dataset for training. Notably, while "People" remains a dominant label even after downsampling, minority topics such as "Ethics" and "Philosophy" have increased in representation, reducing the model's bias towards more frequent topics.

In addition to improving balance, downsampling also addressed the computational limitations associated with training complex models, such as DistilBERT, which require substantial resources. By reducing the size of the dataset, we made the training process more manageable, allowing us to experiment with advanced models that would have been impractical to use with the original dataset size.

3.2. Metrics

Evaluating performance in multilabel classification requires metrics that account for the complexity of predicting multiple labels for each instance. Unlike single-label classification, where only one label is assigned per instance, multilabel classification involves predicting a set of labels, making traditional metrics insufficient. This section presents the key metrics used to assess our models, focusing on how well they handle the multilabel nature of the task while considering potential label imbalances.

3.2.1. F1 Score

The F1 Score provides a useful balance between precision and recall, making it an important metric for evaluating multilabel classification models. In this context, we compute the F1 Score for each individual class to understand how well the model predicts each label. Additionally, we aggregate these scores using both macro and weighted averages to capture different aspects of model performance (4).

The F1 Score is defined as follows:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:

- **Precision** measures the proportion of correct positive predictions out of all positive predictions.
- **Recall** measures the proportion of actual positives that were correctly predicted.

Our evaluation includes three key methods for calculating the F1 Score:

- **Class-Level F1 Score**: The F1 Score is computed for each individual class, allowing us to assess the model's performance on specific labels. This helps us identify which classes the model handles well and which ones may require further attention.

- **Macro F1 Score**: This is the unweighted average of the F1 Scores across all classes. By treating each class equally, the macro F1 Score highlights how well the model performs across both frequent and infrequent labels, making it particularly useful for assessing performance in datasets with class imbalance.
- **Weighted F1 Score**: In this variant, the F1 Scores are averaged based on the support (i.e., the number of instances) of each class. The weighted F1 Score provides a more global view of the model's performance, accounting for the relative frequency of each label in the dataset. This gives us insight into how well the model performs overall, while still considering the impact of class imbalance.

By using both macro and weighted F1 scores, we aim to ensure that our model performs well across all labels (as indicated by the macro F1 score) while also maintaining strong general performance that reflects the distribution of the dataset (as indicated by the weighted F1 score).

3.2.2. Threshold Optimization

Threshold optimization is a critical step in improving model performance in multilabel classification, especially when working with artificially constructed labels (see Section 3.1.1) (5). Since the labels were created based on a predefined 10% threshold for topic coverage, the default decision threshold of 0.5 used by models like Logistic Regression and DistilBERT may not be optimal.

To address this, we performed threshold optimization to find the value that maximizes the F1 Score across all labels during validation. By fine-tuning the decision threshold, we can better balance precision and recall, ensuring that both metrics are optimized for each label. This is particularly important in our scenario, where a consistent decision boundary across all labels is necessary for accurate multilabel predictions.

The goal of this optimization is to select a uniform threshold that enhances the model's ability to classify each label accurately, aligning the model's probabilistic outputs with the specific characteristics of the dataset and improving overall predictive performance.

3.3. Validation Method

To validate the performance of our multilabel classification models, we applied the holdout method, splitting the dataset into training, validation, and test sets. After applying outlier removal, the dataset contains a total of 180,003 documents (as opposed to the original 200,000), with 108,001 allocated to the training set, 36,001 to the validation set, and 36,001 to the test set.

We opted for the holdout method instead of cross-validation due to the size of the dataset. While cross-validation is a reliable technique, it can be computationally expensive and time-consuming when applied to large datasets. A 3-fold cross-validation, for instance, would require substantial resources and longer training times, which could slow down the model development process.

In the holdout method, the training set is used to fit the model, the validation set is employed for hyperparameter tuning and threshold optimization, and the test set is reserved for final model evaluation. By maintaining distinct sets, we ensured that the model's ability to generalize to unseen data could be properly evaluated while also keeping the computational overhead manageable.

3.4. Models

In this section, we present the models employed for the multilabel classification task, with a focus on balancing computational efficiency and performance. We used both lightweight and complex models. The lightweight models—Naive Bayes, Logistic Regression, and SVM with a linear kernel—were selected for their ability to process large datasets quickly (6). For more complex patterns, we trained DistilBERT, a transformer-based model known for its superior performance in NLP tasks (7). Due to its higher computational demands, DistilBERT was trained on a downsampled version of the training set.. The following subsections provide a detailed explanation of each model and its role in this classification task.

3.4.1. Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' Theorem, assuming conditional independence among features given the class label. Despite this often unrealistic assumption, the model's computational efficiency makes it ideal for large-scale text classification tasks.

Its main advantage is speed in both training and prediction, even on large datasets like ours. It handles high-dimensional, sparse data effectively, making it suitable for text features like word counts or TF-IDF vectors. However, the independence assumption can limit its performance, especially when feature correlations exist. Additionally, Naive Bayes may struggle to capture complex patterns in multilabel classification, where label interdependencies are significant.

3.4.2. Logistic Regression

Logistic Regression is a linear model used for binary classification and extended to multilabel classification by applying it independently to each label. In this multilabel setting, each label is treated as a separate binary classification problem, which allows for fast and efficient training, especially with large datasets like ours.

A key advantage of Logistic Regression is its ability to provide probabilistic outputs. This makes it ideal for threshold optimization, where adjusting the decision threshold can significantly improve performance by balancing precision and recall across labels. The model is also computationally efficient and handles high-dimensional feature spaces well, which is beneficial when dealing with text data, where features often come from word counts or TF-IDF vectors.

However, its main limitation is the assumption of linearity between features and the log-odds of the labels. This can restrict the model's ability to capture non-linear interactions between features and labels, which may limit its effectiveness in scenarios where complex relationships exist.

3.4.3. Support Vector Machine

Support Vector Machine (SVM) with a linear kernel is a classification algorithm designed to find the optimal hyperplane that best separates the data points in the feature space. In our multilabel classification task, we employed the one-vs-rest approach, where the model independently classifies each label.

We opted for the linear kernel primarily to handle the high-dimensional nature of text data efficiently. Text datasets often involve numerous features, such as word counts or embeddings, leading to a high-dimensional feature space. The linear SVM is well-suited for this scenario as it performs well in such spaces without the computational cost associated with non-linear kernels, like RBF, which would require significantly more resources given our dataset size.

The linear SVM balances performance and computational efficiency, allowing us to process large datasets while still providing robust classification. However, its key limitation lies in its inability to model complex non-linear interactions between features, which can be critical in capturing dependencies between labels in multilabel classification.

3.4.4. DistilBERT

DistilBERT is a transformer-based model, designed as a smaller and more efficient variant of BERT (Bidirectional Encoder Representations from Transformers). While it is computationally lighter, DistilBERT retains much of BERT's capability to model complex relationships in text, making it particularly suited for tasks involving nuanced language understanding, such as multilabel classification.

DistilBERT's main strength lies in its ability to capture contextual dependencies between words, offering an advantage in multilabel classification where labels can have intricate relationships. This ability allows the model to handle label interdependence more effectively than traditional models. However, despite being more efficient than BERT, DistilBERT still demands significant computational resources, particularly for large datasets.

Given the computational constraints, we trained DistilBERT on a downsampled version of the dataset, reducing the dataset size to make training feasible. This trade-off between dataset size and model complexity allowed us to harness DistilBERT's language modeling power while keeping the training process manageable. While the downsampling limits the model's exposure to the full diversity of the data, the ability to capture deeper linguistic patterns justifies this approach for our multilabel classification task.

4. Experiments

In this section, we describe the experimental approaches used to evaluate model performance in the multilabel classification task. We conducted two approaches: training lightweight models on the full dataset and training a more complex model on a downsampled dataset. These approaches help assess the balance between model complexity and computational efficiency.

4.1. Data Driven Multi-label Classification

In this approach, we trained three computationally efficient models—Naive Bayes, Logistic Regression, and Linear SVM—using the entire training dataset. The focus was on leveraging the full data to explore how these lightweight models perform in a multilabel classification context. For text representation, we employed vectorization techniques to transform the textual data into numerical features suitable for model input.

We applied this method to both the original label set and a grouped version, where related topics were consolidated into broader categories. For the grouped labels, we focused on Linear SVM and Logistic Regression, as Naive Bayes had already shown suboptimal performance with the original label set (see Section 5.1.1 for details). By maintaining consistent preprocessing and training methods across both label sets, we aimed to compare model performance between the original and grouped label structures, while avoiding models that had previously underperformed.

4.1.1. Vectorizer

For the vectorization of the textual data, we used the `TfidfVectorizer`, a method that balances the influence of both common and rare words by applying Term Frequency-Inverse Document Frequency (TF-IDF) weighting (8). This choice helps highlight terms that are more informative for classification while down-weighting those that are too common and contribute less to distinguishing between classes.

We carefully selected the following hyperparameters for the `TfidfVectorizer`, based on the nature of our multilabel classification task:

Max Features: We set `max_features` to 30,000. This limits the number of unique terms included as features, reducing the dimensionality of the data and making the models more computationally efficient. This cap allows us to strike a balance between capturing a sufficient range of terms and avoiding overfitting due to overly complex feature sets.

N-gram Range: We defined `ngram_range` as (1, 2), meaning that both unigrams (individual words) and bigrams (pairs of consecutive words) were used as features. Including bigrams helps capture contextual relationships between words, which is particularly useful in a multilabel classification task where word pairs can provide more information than single terms alone.

Minimum Document Frequency: We set `min_df` to 0.001, meaning that a term must appear in at least 0.1% of the documents to be considered as a feature. This helps filter out extremely rare terms that are unlikely to generalize well, reducing the noise in the feature set.

Maximum Document Frequency: A `max_df` of 0.8 was applied to exclude very common terms that appear in more than 80% of the documents. These terms are usually not informative for classification and could distort the model's ability to focus on more discriminative features.

4.1.2. Model Training

In training the Naive Bayes, Logistic Regression, and Linear SVM models, we encountered computational limitations

that restricted the scope of hyperparameter tuning. Given the dataset's size, we used the default settings provided by scikit-learn for all models to balance efficiency and model performance.

Each model training session required approximately 15 minutes, even with these relatively simple models, due to the scale of the dataset. This made hyperparameter tuning computationally expensive and time-consuming, so we prioritized establishing strong baseline results.

For Logistic Regression, we applied threshold optimization during the validation phase. Since Logistic Regression produces probabilistic outputs, we adjusted the decision threshold to maximize the F1 Score, improving the balance between precision and recall. This was an essential step in fine-tuning the model's predictions for multilabel classification.

For the Linear SVM model, threshold optimization was not applied due to the computational complexity of generating probability estimates. SVMs do not natively output probabilities, but scikit-learn offers a method called Platt scaling to approximate them. Platt scaling involves fitting a logistic regression model on the decision values of the SVM to convert them into probability estimates. This process significantly increases computational cost, as it requires recalculating probabilities for each class across the dataset. Given the already considerable training time and the size of our dataset, applying Platt scaling would have introduced an impractical computational burden. (9)

As for Naive Bayes, we did not explore further optimization or threshold adjustments, as its baseline performance was notably poor in comparison to the other models (see Section 5.1.1). This underperformance made additional tuning efforts unnecessary.

4.2. Model Driven Multi-label Classification

In this approach, we trained DistilBERT, a transformer-based model, on the downsampled version of the dataset to balance computational constraints with model performance. We used the DistilBERT tokenizer to preprocess the text, ensuring that the data was tokenized effectively for input into the model.

We limited the training to 5 epochs, both to prevent overfitting and to manage computational resources effectively. Training for more epochs would have extended the training time significantly, especially considering the 90-minute GPU time limit in Colab, which we were nearing during our experiments. Additionally, we increased the dropout probability to 0.3, applying dropout during training to reduce ovefitting and improve generalization.

This configuration allowed us to leverage DistilBERT's powerful language modeling capabilities while staying within the practical limits of available computational resources.

5. Results

This section presents the results of our experiments, structured into two parts: the full multilabel classification and the simplified classification with grouped labels. Each subsection analyzes the models' performance during the validation phase, focusing on the F1 Score.

We report both macro and weighted F1 Scores to evaluate the models from different perspectives. The macro F1 Score averages the performance across all labels, giving equal importance to each, regardless of frequency. This provides insight into how well the model handles all labels, including the rare ones. The weighted F1 Score, in contrast, takes label frequency into account, offering a more generalized view of the model's performance.

After identifying the best-performing model during validation, we evaluate it on the test set to measure generalization. A detailed analysis of the test results follows, discussing the performance for both the complete and simplified label structures.

5.1. All Labels

5.1.1. Validation Results and Best Model

The validation results for all models are reported in Table 1. Among the models, Logistic Regression, with an optimized decision threshold of 0.2, achieved the highest macro F1 score of 0.597 and a weighted F1 score of 0.682. This indicates its ability to balance performance across both frequent and infrequent labels, making it the best overall performer in our multilabel classification task.

Model	Threshold	Macro Avg F1	Weighted Avg F1
Logistic Regression	0.2	0.597	0.682
DistilBERT	0.4	0.578	0.677
SVM	0.5	0.570	0.663
Multinomial Naive Bayes	0.5	0.055	0.173

Table 1: Validation results for all labels: A comparison of different models evaluated on macro and weighted F1 scores. Logistic Regression with a threshold of 0.2 achieved the best balance across frequent and infrequent labels, with a macro F1 score of 0.597 and a weighted F1 score of 0.682, outperforming DistilBERT and SVM. The underperformance of Naive Bayes (macro F1: 0.055, weighted F1: 0.173) reflects its struggle with the complexity of mixed-topic documents.

The DistilBERT model, with a threshold of 0.4, followed closely with a macro F1 score of 0.578 and a weighted F1 score of 0.677. Although slightly lower than Logistic Regression, its strong performance suggests that DistilBERT effectively captures the nuances of the multi-topic documents in the dataset, despite the computational challenges associated with training on a downsampled set.

The SVM model, which used a threshold of 0.5, achieved a macro F1 score of 0.570 and a weighted F1 score of 0.663. Although competitive, its slightly lower scores likely result from the absence of threshold optimization, which could have enhanced the model's ability to manage the imbalance between relevant and irrelevant labels.

Finally, the Multinomial Naive Bayes model significantly underperformed, with a macro F1 score of 0.055 and a weighted F1 score of 0.173. These results indicate that Naive Bayes struggled with the dataset's complexity, particularly due to the mixed-topic nature of the documents.

Based on these results, Logistic Regression is identified as the best model for the validation phase, balancing both macro

and weighted F1 metrics. This model will be further tested on the test set to evaluate its generalization capabilities and better understand its performance across different label distributions.

5.1.2. Test Results

The Logistic Regression model with a threshold of 0.2 was tested on the test set, yielding a macro average F1 score of 0.59 and a weighted average F1 score of 0.68 (Table 2). These results are consistent with the validation phase, where the model achieved a macro F1 of 0.597 and a weighted F1 of 0.682, indicating strong generalization to unseen data. A macro F1 score of 0.59 demonstrates the model's balanced performance across all labels, independent of label frequency, and shows that the model is capable of managing both frequent and infrequent labels. The weighted F1 score of 0.68 reflects slightly stronger performance on more frequent labels, as it accounts for label frequency in the dataset. This suggests the model is handling common labels well, but there may still be challenges with less frequent labels, as indicated by the gap between the two scores.

Model	Threshold	Macro Avg F1	Weighted Avg F1
Logistic Regression	0.2	0.59	0.68

Table 2: Test results for the Logistic Regression model on the test set. The Logistic Regression model achieved a macro F1 score of 0.59 and a weighted F1 score of 0.68, closely aligning with the validation results (macro F1: 0.597, weighted F1: 0.682). These results demonstrate the model's strong generalization ability and its consistent performance across both frequent and infrequent labels, confirming the robustness observed during validation.

Overall, while the test results confirm that Logistic Regression is robust in a multilabel classification context, generalizing well across different label frequencies, there remains room for improvement, particularly with rare labels. Enhancing the model's ability to accurately classify less frequent labels could further boost its performance and provide better overall results in highly imbalanced datasets.

Label-wise F1 Score Analysis. The Logistic Regression model's F1 scores for each label in the test set are depicted in Figure 8. Labels are ordered from highest to lowest F1 score, and the bar color intensity represents the mean TF-IDF weight of the top three most influential words associated with each label.

Labels with the highest F1 scores, such as *Sports*, *Life*, and *People*, exhibit relatively lower mean TF-IDF weights, as shown by the lighter bar colors. This suggests that the key terms for these categories are more frequent in the dataset, aiding the model in classifying documents under these labels with greater accuracy.

Conversely, labels with the lowest F1 scores, such as *Communication*, *Ethics*, *Entities*, and *Information*, exhibit higher mean TF-IDF weights, represented by darker colors. This indicates that the key terms for these labels are less frequent and more specific, which challenges the model's classification accuracy.

However, this inverse relationship between TF-IDF weight and F1 score is not strictly linear. For labels with mid-range F1 scores, such as *Government*, *Health*, and *Education*, the variation in TF-IDF weights is less pronounced, suggesting that other factors beyond term frequency may influence classification performance. While frequent terms generally help improve accuracy, labels in this middle range do not follow a clear pattern, indicating a more complex interaction between term specificity, label diversity, and model performance.

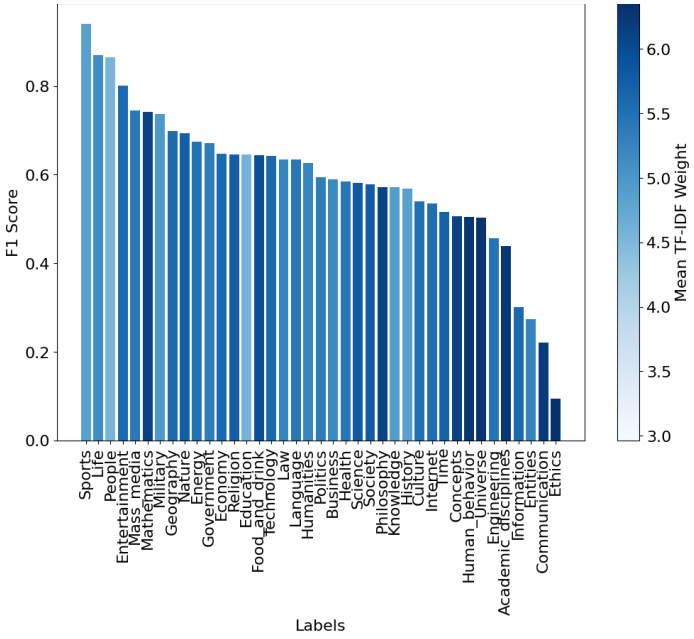

Figure 8: F1 scores by label on the test set, ordered from highest to lowest. Bar color intensity represents the mean TF-IDF weight of the top three words for each label. An inverse relationship between F1 score and TF-IDF weight is evident for the highest and lowest scores, but less clear for mid-range labels.

Further analysis based on the support for each label, as illustrated in Figure 9, shows that the relationship between support and F1 score is not straightforward. While *People* has the highest support, it only ranks third in terms of F1 score, trailing behind *Sports* and *Life*. For labels with mid-range F1 scores, there is no clear pattern that links higher support to better performance. This suggests that factors beyond support, such as term specificity and distribution within the dataset, may play a more significant role in influencing the model's classification accuracy.

Analysis of Influential Words in Highest and Lowest Performing Labels. The performance analysis of the Logistic Regression model on the test set reveals that the label *Sports* achieved the highest F1 score of 0.94, while the label *Ethics* had the lowest F1 score of 0.09. Table 3 presents the top three most influential words for both labels, along with their respective coefficients and TF-IDF weights.

As shown in Table 3, the words associated with *Sports* have high coefficients and relatively low TF-IDF weights, indicating their strong relevance and frequent occurrence in the dataset. This combination enables the model to classify documents under *Sports* with high accuracy. Additionally, the high support

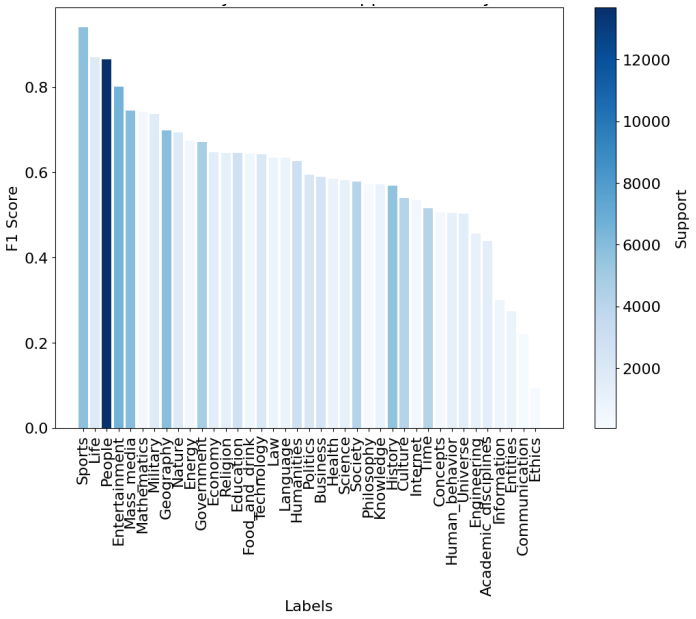

Figure 9: F1 scores by label with support intensity on the test set. Labels are ordered by descending F1 score, with darker colors representing higher support. The relationship between support and F1 score is not consistent. For instance, *People* has the highest support but ranks third in F1 score. In mid-range F1 scores, no clear pattern emerges, indicating that other factors, like term specificity and distribution, may play a larger role.

Label	Word	Coefficient	TF-IDF Weight
Sports	**League**	7.8147	3.6339
	Football	8.6421	3.7661
	Played	8.6606	3.1811
Ethics	**Utility**	4.2568	6.3380
	Moral	5.1725	6.2373
	Ethic	5.4987	6.4941

Table 3: Top 3 Most Influential Words for the Best-Performing Label (*Sports*) and the Worst-Performing Label (*Ethics*). *Sports* benefits from frequent and influential terms, along with high support, leading to the highest F1 score. *Ethics*, in contrast, is characterized by rare and specific terms, with low support, contributing to its significantly lower F1 score.

of 5725 instances further improves the model's performance, leading to the highest F1 score.

In contrast, the words associated with *Ethics* have lower coefficients and higher TF-IDF weights, suggesting these terms are more specific and less frequent, making it harder for the model to generalize. The minimal support of just 59 instances further hinders the model's ability to learn, contributing to the significantly lower F1 score.

However, when analyzing the mid-range labels, the relationship between TF-IDF, support, and F1 score is less clear. The strong influence of TF-IDF weights and support is most evident at the extremes, such as with *Sports* and *Ethics*, but does not consistently explain the performance of labels with mid-range F1 scores.

5.2. Grouped Labels

5.2.1. Validation Results and Best Model

Table 4 presents the validation results for the grouped labels approach. Logistic Regression with a threshold of 0.3 outperformed other models, achieving a weighted average F1 score of 0.784 and a macro average F1 score of 0.759. These scores indicate that Logistic Regression strikes a strong balance between precision and recall across various grouped labels, proving effective in managing the inherent complexity of multilabel classification in this context.

Model	Threshold	Macro Avg F1	Weighted Avg F1
Logistic Regression	0.3	0.759	0.784
SVM Grouped	0.5	0.742	0.773

Table 4: Validation results for grouped labels comparing Logistic Regression and SVM. Logistic Regression achieves the highest Macro and Weighted F1 scores with a threshold of 0.3, indicating better performance in handling both frequent and infrequent labels.

The SVM model also demonstrated competitive performance, with a weighted average F1 score of 0.773 and a macro average F1 score of 0.742. However, the slightly lower results compared to Logistic Regression suggest that SVM may not optimize as well for grouped labels, likely due to the absence of threshold optimization. The ability of Logistic Regression to fine-tune the decision threshold allows it to manage imbalanced classes and variations in label distributions more effectively.

Given these results, we select Logistic Regression with a threshold of 0.3 to proceed to the test phase.

5.2.2. Test Results

Logistic Regression with a threshold of 0.3 was applied to the test set for grouped labels. The model achieved a macro average F1 score of 0.76 and a weighted average F1 score of 0.79, as shown in Table 5. These scores indicate that the model performs well across both frequent and less frequent labels, maintaining balanced classification performance. The high weighted F1 score of 0.79 suggests strong performance on labels with higher support, while the macro F1 score of 0.76 shows that the model also generalizes well across all labels, irrespective of their frequency.

Model	Threshold	Macro Avg F1	Weighted Avg F1
Logistic Regression	0.3	0.76	0.79

Table 5: Test results for the Logistic Regression model on grouped labels. The model achieves a Macro F1 score of 0.76 and a Weighted F1 score of 0.79, indicating strong generalization across both frequent and infrequent labels in the test set.

This result reflects the model's ability to handle the complexity of grouped labels effectively, achieving robust performance across a range of categories. The relatively high scores indicate that Logistic Regression successfully captures the relationships within the grouped label structure, providing consistent classification accuracy.

When compared to the results on all non-grouped labels (Table 2), where the macro F1 score was 0.59 and the weighted

F1 score was 0.68, it is clear that grouping the labels improved overall performance. By simplifying the label space, the model was able to achieve higher F1 scores, particularly benefiting from better generalization across both frequent and less frequent labels.

Label-wise F1 Score Analysis. The analysis of grouped labels within the Logistic Regression model shows varied relationships between F1 scores, TF-IDF intensities, and support intensities, as demonstrated in Figures 10 and 11. Each label achieves an F1 score above 0.6, marking a significant improvement over the non-grouped labels, which exhibited greater variability in performance. This demonstrates the effectiveness of simplifying the classification task by grouping similar labels, which helps in reducing the complexity and enhances the model's ability to generalize across diverse topics.

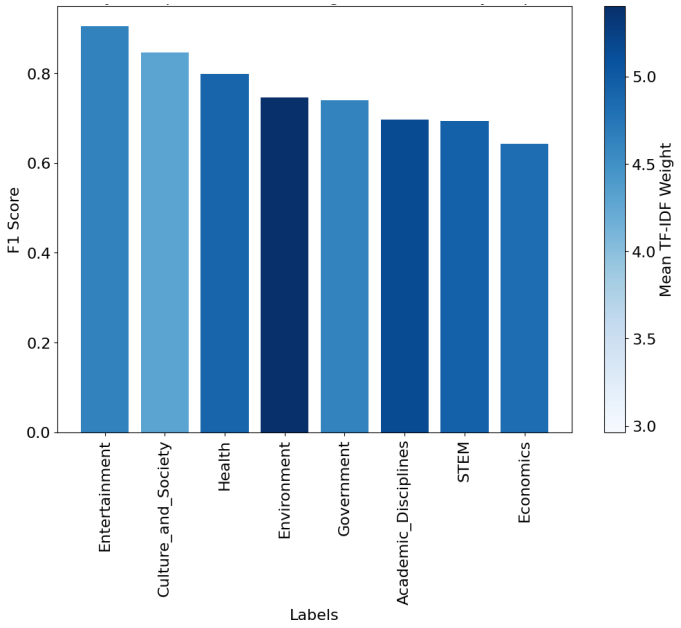

Figure 10: F1 scores and mean TF-IDF weights for grouped labels on the test set. The chart presents F1 scores alongside the average TF-IDF weight of the top three words for each label category. While *Entertainment* achieves the highest F1 score, there is no consistent pattern indicating that lower TF-IDF weights correspond to higher F1 scores across categories.

Regarding TF-IDF intensity, the data do not exhibit a consistent trend linking lower TF-IDF weights to higher F1 scores, as illustrated in Figure 10. While *Entertainment* achieves the highest F1 score of 0.90 with relatively lower TF-IDF weights, suggesting that its influential words are common across the dataset, this pattern does not hold uniformly across all labels. For instance, other categories such as *Academic Disciplines* demonstrate lower F1 scores despite having higher TF-IDF weights, but similar patterns are not consistently observed across other label groups. This variability indicates that factors other than the commonality of terms might influence the classification performance.

In our analysis of F1 scores by grouped label with support intensity, it becomes evident that the relationship between the amount of support and classification performance is not straightforward. Higher support does not uniformly equate to higher performance, as demonstrated by the *Health* category. Despite having the lowest support among the categories at only 3,016 instances, *Health* achieves a high F1 score of 0.80, ranking third overall. In contrast, *Academic Disciplines*, which benefits from much higher support with 13,068 instances, only manages an F1 score of 0.70. This indicates that factors beyond the sheer volume of data, such as the internal diversity and complexity of topics, or the distinctiveness of the features within a group, significantly impact the efficacy of the classification. The example of *Culture and Society*, which has the highest support yet does not achieve the highest F1 score, further underscores that mere data quantity cannot guarantee superior model performance.

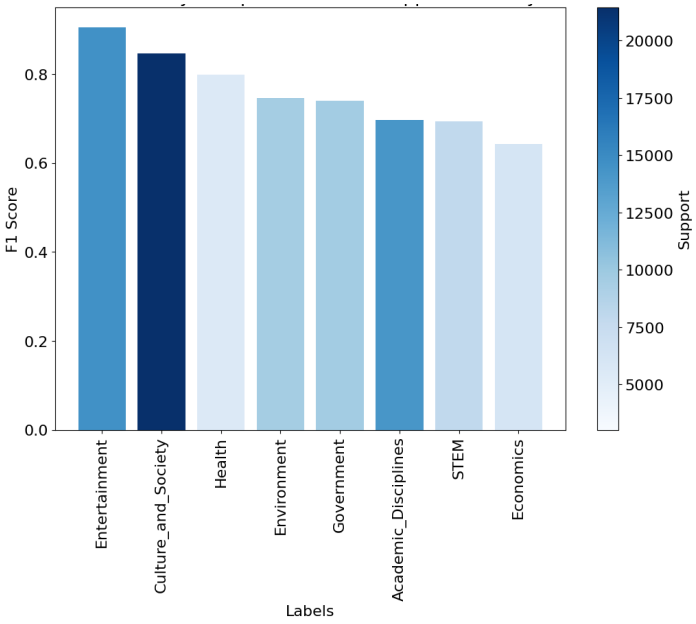

Figure 11: F1 scores by grouped label with support intensity. The relationship between support and performance is not linear: *Health*, with the lowest support (3,016 instances), achieves a high F1 score (0.80), while *Academic Disciplines*, with much higher support (13,068 instances), scores only 0.70. This highlights how factors like label diversity and feature distinctiveness can impact classification performance, as seen with *Culture and Society*, which has the highest support but not the highest F1 score.

Analysis of Influential Words in Grouped Labels: Entertainment vs. Economics. In the analysis of the Logistic Regression model's performance on the grouped labels test set, the label *Entertainment* achieved the highest F1 score (0.90), while the label *Economics* had one of the lowest F1 scores (0.64). To better understand these results, we analyze the top three most influential words (based on their coefficients in the Logistic Regression model) and their associated TF-IDF weights for both labels, as shown in Table 6 .

For the *Entertainment* label, the most influential words, such as **Championship**, **Cast**, and **Played**, are associated with both high coefficients (6.2658, 7.5623, and 7.7720, respectively) and moderate TF-IDF weights. This combination of high influence and commonality (moderate TF-IDF) suggests that the words

Label	Word	Coefficient	TF-IDF Weight
Entertainment	**Championship**	6.2658	3.8467
	Cast	7.5623	4.0950
	Played	7.7720	3.1811
Economics	**Company**	5.7460	3.2153
	Businessman	6.4291	5.4083
	Business	6.5310	3.7729

Table 6: Top 3 Most Influential Words for the Best-Performing Label (*Entertainment*) and the Worst-Performing Label (*Economics*). *Entertainment* benefits from frequent and influential terms, contributing to its high F1 score. In contrast, *Economics* includes more specific and less common terms like *Businessman*, leading to a lower F1 score and increased classification difficulty.

are both significant and frequent across the dataset. The relative frequency of these terms simplifies the model's classification task, allowing it to achieve the highest F1 score for this label.

In contrast, the *Economics* label features words like **Company**, **Businessman**, and **Business**, which have lower coefficients compared to *Entertainment* and more variable TF-IDF weights. Specifically, **Businessman** has a notably higher TF-IDF weight (5.4083), indicating it is less common and more specific within the dataset. This higher specificity likely challenges the model, reducing its ability to generalize effectively for this label, which contributes to its lower F1 score. Although the coefficients for these terms are moderate, the greater term uniqueness (higher TF-IDF) likely increases classification difficulty, leading to the observed performance gap between *Entertainment* and *Economics*.

This analysis highlights that the model's performance is not solely determined by the magnitude of word influence (coefficients) but is also strongly impacted by term frequency and specificity (TF-IDF weights). The model performs best when influential words are also common across the dataset, as seen with *Entertainment*. In contrast, labels defined by more unique terms, such as *Economics*, introduce greater challenges, leading to lower classification performance.

6. Conclusions

This study had two primary objectives. The first objective was to compare the performance of a data-driven approach, using lightweight models such as Logistic Regression trained on the full dataset, against a model-driven approach, employing more complex architectures like DistilBERT, which was trained on a reduced subset of the dataset due to computational limitations, in the task of multilabel classification across all labels. The second objective was to assess the impact of label grouping on model performance, focusing on whether this strategy could improve the classification results, particularly for lightweight models.

In addressing the first objective, the results clearly demonstrated that the data-driven approach, represented by Logistic Regression with a threshold of 0.2 and trained on the full training set, provided the best overall performance. Logistic Regression achieved a macro F1 score of 0.59 and a weighted F1 score of 0.68 on the test set, showing strong generalization across both frequent and infrequent labels. In contrast, the model-driven approach using DistilBERT, despite its ability to capture deeper relationships in the data, was constrained by the reduced training set and computational overhead, resulting in slightly lower performance during validation.

For the second objective, the impact of label grouping on performance was explored by applying this strategy to the lightweight models trained on the full dataset. Grouping similar labels into broader categories led to a significant improvement in model performance. Logistic Regression, when applied with label grouping and a threshold of 0.3, achieved a macro F1 score of 0.76 and a weighted F1 score of 0.79 on the test set.

References

[1] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.

[2] S. Vijayarani, M. J. Ilamathi, M. Nithya, *et al.*, "Preprocessing techniques for text mining-an overview," *International Journal of Computer Science & Communication Networks*, vol. 5, no. 1, pp. 7–16, 2015.

[3] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, pp. 1263–1284, 2009.

[4] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[5] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "Addressing imbalance in multilabel classification: Measures and random resampling algorithms," *Neurocomputing*, vol. 163, pp. 3–16, 2015.

[6] S. Brindha, K. Prabha, and S. Sukumaran, "A survey on classification techniques for text mining," in *2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, pp. 1–5, IEEE, 2016.

[7] V. Sanh, "Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[8] S. Qaiser and R. Ali, "Text mining: use of tf-idf to examine the relevance of words to documents," *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25–29, 2018.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Support vector machines - scores and probabilities," 2011. Accessed: 2024-09-10.