The 17 UN Sustainable Development Goals: Classification of Research Topics Using BERT and Logistic Regression

Eunike Endariahna Surbakti[1,*], Fenina Adeline Twince Tobing[1], Charlie Frederico[1], Sagita Sasmita Wijaya[1], Felix Frederico[1]

[1]Informatics, Faculty of Informatics and Engineering, Multimedia Nusantara University, 15810, Tangerang, Indonesia

*Author to whom correspondence should be addressed:
E-mail: eunike.endariahna@umn.ac.id

(Received June 16, 2025; Revised January 19, 2026; Accepted February 17, 2026)

Abstract This study addresses the challenge of classifying over 3,000 research articles produced by a university between 2018 and 2023 according to the 17 United Nations (UN) Sustainable Development Goals (SDGs), a task that is traditionally inefficient when performed manually. To automate this process, two machine learning systems were developed and evaluated using a dataset of 76,958 records. The first approach implements a Bidirectional Encoder Representations from Transformers (BERT) model, preprocessed with NLTK and fine-tuned over 4 epochs with a learning rate of 2e-5 and a batch size of 32 using a 70/30 train-test split; this model achieved an accuracy of 90.68%, precision of 0.99, recall of 0.82, and an F1-score of 0.87. The second approach utilizes a Logistic Regression model with TF-IDF vectorization, an L1 penalty, and the Saga solver, trained on an 80/20 split without additional data cleaning, reaching an accuracy of 90.01%, precision of 0.86, recall of 0.82, and an F1-score of 0.84. Ultimately, while both models demonstrated strong performance, the BERT-based model provided superior precision and overall classification quality, offering the university a consistent and efficient method to align institutional research with global sustainability targets and support faculty accreditation

Keywords: 17 UN SDG; BERT; Logistic Regression; Research Topic; Text Classification

1. Introduction

Technological advancements have transformed the way we access information, particularly with the advent of the internet and electronic books in the digital age. Digital publishing exemplifies this shift, enabling quicker, broader, and more efficient distribution and access to information. Through digital publishing, information can be presented in multiple formats such as text, images, and multimedia providing users with a richer understanding[1)]. Consequently, digital publishing not only showcases technological progress but also redefines our perception of publishing, ushering in a new era of information dissemination and consumption[2)].

The advent of technology in information retrieval has fundamentally transformed how we manage and store data. As the volume of available information continues to grow, the need for effective information filters has become increasingly critical. Information filtering is essential for addressing this challenge, especially given the multitude of information sources both online and offline[3)]. The primary difficulty lies in distinguishing between valid and invalid information and ensuring that the information obtained is reliable. Effective information filtering also demands the skills to access accurate and relevant knowledge[4)].

At the Research and Community Service Institute of Multimedia Nusantara University (LPPM UMN), the objective is to enhance work efficiency by categorizing research article data from UMN lecturers. This categorization aims to streamline the tasks of LPPM UMN employees by organizing the existing article data into relevant categories, thereby facilitating quicker data analysis. Previously, LPPM UMN employees manually grouped these categories. This process involves analyzing the titles of articles published by UMN lecturers, which is time consuming due to the large volume of published articles.

At UMN, there are approximately 200 lecturers who have collectively published over 3000 articles. This research focuses on categorizing these articles according to the 17 United Nations Sustainable Development Goals (UN SDG) issued by the United Nations (UN)[5)]. The UN established these 17 SDG categories to address a range of critical global issues, including poverty, hunger, education, health, gender equality, clean water, affordable energy,

Cite: E. Surbakti et al., "The 17 UN Sustainable Development Goals: Classification of Research Topics Using BERT and Logistic Regression". Evergreen, 13 (01) 417-431 (2026). https://doi.org/10.5109/7411078.

decent work, and environmental protection[6].

Search algorithms provide a solution to help filter and locate information tailored to specific needs. In a study by Jingzhou Liu, the Convolutional Neural Networks (CNN) algorithm was utilized for multilabel text classification to enhance CNN's performance for optimal classification results. This involved extensive testing and training with various datasets to improve the quality of the CNN algorithm[7]. Another study by Samuel Rodriguez Medina compared multiple algorithms, using the AUROC value as a benchmark. This study evaluated various machine learning and deep learning algorithms to determine the one with the best AUROC value[8]. Additionally, Santiago Gonzales conducted a study comparing the accuracy of text classification algorithms, including the BERT algorithm and machine learning algorithms like multinomial naive bayes, linear support vector classifier, and logistic regression[9]. Hence, another similar study by eunike using LSTM have gained low error[10] also with the similar study by Eunike Endariahna have use Analytic Hierarchy Process[11]. Therefore, there are also several similar study by Ravikant Kholwal using Logistic Regression, random forest, and K-Nearest Neighbor[12]. Another similar study by Satya Abdul Halim Bahtiar conducted a study comparing algorithm between logistic regression and naïve bayes[13]. Furthermore, another study by Arash Hajikhani comparing the accuracy of multiples algorithms using naïve bayes, linear support vector machine, and logistic regression[14]. Thus, another similar study conducted by Pinky Yadav compared multiples algorithms such as BERT, LSTM, VADER, and SVM and has achieve an exceptional result[15]. Moreover, another study conducted by Umesh Gurnani using binary logistic regression to determine associate of postural risk factors[16]. Building upon these previous works, this study aims to develop and evaluate machine learning–based text classification systems to automatically categorize research titles according to the 17 United Nations Sustainable Development Goals (UN SDGs).

2. Literature Review

In this section, there are numerous studies that examine various algorithms, approaches, and discoveries relevant to the research problem-solving are reviewed in the following section. The following numerous studies are explained in Table 1.

Table 1: Numerous Studies

	Research Title	Algorithm	Result
1	Comparing BERT against traditional machine learning text classification	Bidirectional Encoder Representations from Transformers (BERT) & Term Frequency-Inverse Document Frequency(TD-IDF)	The BERT algorithm has outperformed the TF-IDF algorithm in sentiment analysis, with an accuracy of 90.90% compared to 84.80% for TF-IDF
2	Multi-Label Text Classification with Transfer Learning for Policy Documents	Bidirectional Encoder Representations from Transformers (BERT) & logistic regression & multinomial naive bayes	Outperforming the logistic regression algorithm (which receives a value of 0.89) and the multinomial Naïve Bayes Algorithm (which receives a value of 0.82), whereas the BERT algorithm achieves the best AUROC value of 0.92, which constitute the study's comparison

2.1. Text Classification

Text classification is a machine learning technique that automatically categorizes unstructured text into predefined categories[17]. Identifying specific patterns and features in the text, facilitates efficient grouping and better filtering of data, enhancing the system's ability to quickly provide relevant information[18]. This technology has extensive applications across various industries, improving operational efficiency.

For instance, in business, text classification is used to analyze customer feedback in real-time, identify trends or issues, and respond promptly to enhance customer satisfaction. In healthcare, it categorizes medical records, aiding doctors and researchers in easily finding important information, thus improving patient care quality. In the education sector, these algorithms organize research papers, journal articles, and learning materials, making access easier for students and academics. As machine learning and data analytics technologies advance, the potential applications of text classification continue to expand, offering significant benefits across multiple sectors[18]. Hence, information retrieval, sentiment analysis, topic labeling, and news classification are among of the most frequently utilized varieties of text classification[19]. Furthermore, the classification process on a large volume of data will be much more tedious and challenging to complete manually than it would be automatically[20].

2.2. BERT Algorithm

The BERT (Bidirectional Encoder Representations from Transformers) algorithm stands as a remarkable natural language processing (NLP) innovation pioneered by

Google[21]. Its key advantage lies in its capability to comprehend words bidirectionally within context, surmounting the constraints of prior models that only analyze the context in one direction. Through pre-training on extensive tasks like missing word prediction and sentence relationship comprehension, BERT acquires a profound understanding of language structure. Leveraging an attention mechanism, BERT adeptly assesses the significance of individual words in a sentence, capturing intricate relationships and dependencies, thus enhancing language comprehension[22]. Below is the equation of the BERT Algorithm[23,24]

$$Attention\ (Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{dk}}\right) \cdot V \qquad (1)$$

Whereas:

1) Q, K, and V is the matrix representing Query, Key and Value Vectors.
2) $\frac{QK^T}{\sqrt{dk}}$ explained the multiplication dot Q and K^T then divided with square from the vector Key dimension $\sqrt{dk}$
3) Softmax creates a probability distribution that adds up to one by converting the scaled dot values into attention weights.
4) Softmax$\left(\frac{QK^T}{\sqrt{dk}}\right) \cdot V$ describe the linear combination from the Vector values (V) using the attention weights.

Built on a transformer architecture, BERT processes data sequences efficiently and adaptably across various NLP tasks while retaining its core structure. BERT furnishes contextual word representations that can be further optimized or ”finetuned” for specific tasks post-pre-training, thereby bolstering performance in endeavors such as text classification and textbased entity recognition. The BERT algorithm employs the attention mechanism process, which involves several steps, including calculating attention scores and applying them to the input data. The formula used to calculate attention scores in BERT is typically based on the dot-product attention mechanism, which can be represented as[9].

2.3. Natural Language Processing

Natural Language Processing (NLP) is a technique that merges the domains of linguistics, computer science, and artificial intelligence to empower computers to comprehend, interpret, and generate human language. Utilizing advanced methods and algorithms, NLP deciphers the structure and semantics of human text, fostering a more seamless interaction between humans and computers. NLP finds diverse applications such as virtual assistants, information retrieval systems, social media sentiment analysis, and document classification[25].

This technology enhances user experiences across various facets of daily life by delivering more tailored and responsive services, rendering interactions with technology more intuitive and beneficial[26].

2.4. BERT Model

BERT modeling refers to the use of the BERT (Bidirectional Encoder Representations from Transformers) architecture, which is designed to understand the context in text in a highly efficient and deep manner[27]. BERT uses a transformer architecture that allows the model to pay attention to and understand the relationships between words in the text from both directions, known as a bidirectional approach[28]. There are several stages in the design of a BERT model, such as: Pre-training , Fine-tuning , Input representation, Model architecture, Training and Evaluation.

2.5. United Nations (UN) Sustainable Development Goals (SDG)

In 2015, all countries joining the United Nations (UN) signed onto a global plan called the Sustainable Development Goals (SDGs). These 17 goals tackle major problems facing the world today, from poverty and hunger to education, health, and environmental issues[6]. The SDGs go beyond just fixing these problems. They aim to create lasting improvements across all aspects of life. This includes fostering equal opportunities for all and tackling big challenges like climate change and inequality.

The plan succeeds through collaboration.. Ultimately, the SDGs aim to build a better future for everyone on Earth[29]. Given the inherently multi-dimensional nature of the SDGs (i.e., one research output or text may contribute to several goals simultaneously), many recent studies have adopted *multi-label classification* approaches. For example, Yao et al. (2024) developed a system to assign multiple SDG labels to open-educational-resources (OERs) using an AutoGluon framework; they addressed key challenges such as category imbalance and cross-domain transferability[30] Similarly, Hsu et al. (2022) proposed a combinatorial fusion algorithm to improve precision in mapping documents to SDGs by combining topic-model and semantic-link classifiers[31]. Another recent comparative study evaluated seven multi-label methods on real-world patent and publication datasets, highlighting that deep text-CNN/RNN models outperform more classical methods when label correlations and hierarchical structures are present[32]. These studies underscore three important insights: (i) multi-label modelling is essential for SDG-relevant text classification; (ii) label imbalance and inter-label correlations must be explicitly addressed; and (iii) ensemble or fusion strategies often yield superior performance.

2.6. Logistic Regression

Logistic Regression is one machine learning algorithm for

classification which develops a statistical model that demonstrates the relationship between a set of predictor variables and a binary result[33]. The target class's probability is predicted by Logistic Regression. In addition to proven highly effective at solving binary classification issues, this algorithm may also be used for multi-class classification[34]. The advantages of logistic regression is that it does not require an enormous amount of computational power, thus a device with high system specifications isn't required to use it. Furthermore, this algorithm provides probability values to ensure more observations can be made[35]. Logistic Regression may also be used to tackle classification challenges, this algorithm generally employ a linear combination of several explanatory variables, according to the sigmoid function argument[36].

To categorize the text of speech articles written by lecturers at private university, this research uses logistic regression. The logistic regression equation is as follows:

$$p(xi) = \frac{exp\ (w_k^T xi)}{\sum_{j=1}^{K} exp\ (w_k^T xi)} \quad (2)$$

Whereas:

1) $p(xi)$ is the probability of a D-dimensional point where xi ∈{x1,x2,…,xN}associated with class k ∈ {1,2,..., K}.
2) xi is a D-dimensional representation of the feature vector.
3) K is the number of class labels.
4) W ={w1,w2,...,wK} is the parameter vector for all classes k.

3. Methodology

3.1. Literature Study

This study includes a literature study that aims to collect information from previous studies. The literature used relates to various aspects such as multilabel text classification, preprocessing, precision, recall, f1-score, and accuracy. In addition, the BERT algorithm is also a focus in the collected literature. All of these literatures have been collected and used as the main reference in this study.

3.2. Data Gathering

In the data collection method, the author uses literature study, observation, and interviews. Literature study is conducted by reviewing scientific journals, books, and scientific articles that are relevant to the research questions. Observations and interviews are conducted directly at the research location with related sources. In addition, this study uses a dataset from the Huggingface website, which includes the UN SDG category and article titles for preprocessing and model training purposes.

3.3. Model Design & Implementation

This stage is the stage where researchers design the system to be created in the form of a flowchart design that will be used in the implementation stage. This stage is carried out to compile the previous design into a functioning system according to what has been previously designed. At this stage, a model will be created that begins with loading the dataset used to call the dataset to be used. In this study, the dataset used is the dataset from the Huggingface website. Then continued with data labeling, carrying out the preprocessing stage, training the model and evaluating the model.

3.4. Model Testing and Evaluation

At this stage, the system will perform the testing phase that has been designed in the previous stage. In this phase, several scenarios will be used, with the goal of using these various scenarios to find the best model that will ultimately be used. After conducting the tests, an evaluation will be carried out to summarize all the activities and findings during the testing process.

Figure 1 is a flowchart of the overall model. This section outlines the processes involved in designing the BERT algorithm for text classification on SDG labels. At this stage, the flowchart serve as the foundation for the model design.

3.4.1. Import Libraries

Importing libraries is a step to import the necessary libraries needed throughout the model development process. Libraries support the model creation process, such as preprocessing, training, and evaluating the model, making these tasks easier due to the assistance provided by the libraries used.

3.4.2. Load Dataset

In this research, a dataset is used for model development.

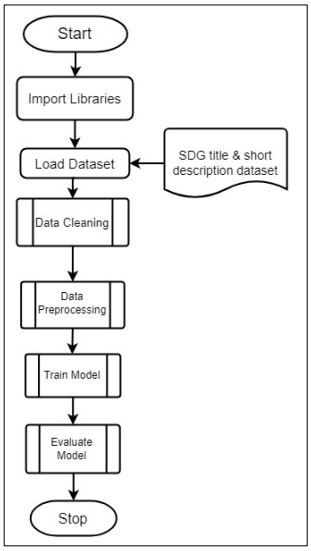


Fig. 1: Main Flowchart

This dataset is obtained from a website called Huggingface. Hugging Face is a well known platform and community for developing and sharing advanced natural language processing (NLP) models. They provide a model repository, transformers library, Application Programming Interface (API), and datasets. The dataset used in this study contains 76,958 article titles, which are used in the current model development.

3.4.3. Data Labelling

Data labeling is a step to assign labels to specific data. The purpose of data labeling is to categorize the data into several categories based on their characteristics. Data labeling is done by importing the dataset and then categorizing these datasets based on an existing CSV file, making them readable and applicable labels for the model creation process. In this research, data labeling is divided into 17 categories, corresponding to the 17 UN SDG categories.

3.4.4. Preprocessing

Figure 2 shows the preprocessing step, the purpose for preprocessing steps is to render the data more organized and accessible for model development. These stages provide an overview of the research object, an analysis of all existing problems, and the approaches used in the research. This section also details the design of the research conducted, both the general design of the system built and the more specific design aspects. The following methodology used in this research is as follows:

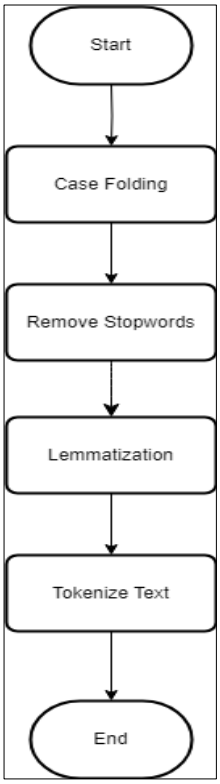


Fig. 2: Preprocessing Flowchart

The goal of the preprocessing stage is to simplify and clean textual data to make it easier to process during m odel development. The steps involved include:

a) Case Folding: All characters in the text are converted to lowercase to maintain consistency and reduce dimensionality (e.g., treating “Education” and “education” as the same word).
b) Stopword Removal: Commonly used words that do not contribute significant meaning to the analysis—such as *and*, *or*, *that*, *with*, *the*—are removed to focus on more informative terms.
c) Lemmatization: Words are reduced to their base or dictionary form (lemma) without changing their core meaning, e.g., *running* → *run*, *studies* → *study*.
d) Tokenization: Text is divided into smaller units called tokens (usually words or subwords). In this research, tokenization was performed using the Tokenizer function from the Transformers library, enabling models such as BERT to process contextual relationships between words more effectively. By performing these preprocessing steps, textual data is expected to become more structured and ready for use in training models like BERT or other models for text analysis or classification.

3.4.5. Training Model

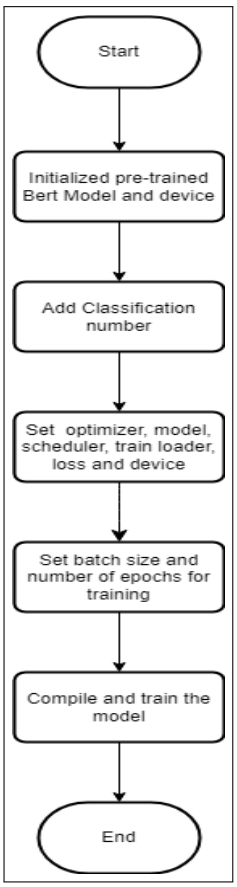


Fig. 3: Training Model Flowchart

Figure 3 illustrates the model development process, namely the model training. Pre-trained model, add classification number, set optimizer, model, scheduler, train loader, loss and device, set batch size and number of training epochs, lastly compile and train the model using dataset are the steps involved in the training model. In the training stage of the model, there are several steps that need to be performed, which will be explained as follows:

a) Pre-trained model: At this stage, the BERT algorithm utilizes tokenization performed during the preprocessing step. Tokenization is crucial to prepare the data so that the BERT model can train on the dataset while considering the context of each token.
b) Add classification number: This step refers to determining the number of categories or labels that will be used in training the model. For example, in multilabel classification, it involves specifying how many labels the model will predict.
c) Set optimizer, model, scheduler, train loader, loss, and device.
d) Set batch size and number of epochs for training: In this step, the batch size for training and the number of epochs (iterations) over the data are determined. The appropriate batch size allows efficient use of computational resources, while the suitable number of epochs enables the model to learn relevant patterns from the data without overfitting or underfitting.
e) Compile and train the model using dataset: The final step involves executing the model training process using the prepared dataset and the configurations set earlier. This process aims to ensure that the model is well-trained and performs optimally on the given data. Overall, this flow depicts the necessary steps from initial data preparation to model training using appropriate techniques and configurations to achieve desired results in text classification tasks using BERT or similar models.

3.4.6. Evaluate Model

Following the training model steps, the mode has been evaluated as shown in Figure 4. In order to demonstrate the model's quality, the previously trained model will be tested using the testing dataset during the evaluate model steps. Additionally, it measures how effectively the model matches the training data at the calculate training loss and validation loss steps, while the validation loss gauges the mode's performance on the data with the goal of enabling the model to adapt to new data. Precision, recall, F1-Score, and accuracy of the constructed mode will be grouped into categories that will organize the prediction outcomes.

In the evaluate model stage, the trained model is evaluated using a testing dataset to measure its quality. Evaluation involves calculating training loss and validation loss to assess how well the model fits the training data and adapts to new data. The results of this evaluation include precision, recall, F1-score, and accuracy, providing an overview of how well the model can predict or classify new data. The primary goal of the evaluate model stage is to assess the model's performance and ensure that it can provide accurate results as expected.

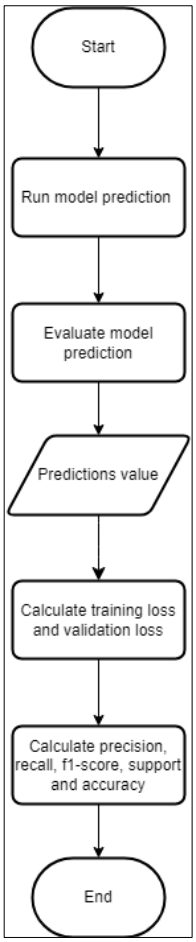


Fig.4: Evaluate Model Flowchart

4. Result and Discussion

4.1. Implementation Model

The implementation of the model is the development stage following the earlier design phase, where the planned aspects are put into practice. The implementation phase involves several key processes: importing libraries, data labeling, and preprocessing. It progresses with the training of the model, where the model is trained using the data that has been processed during the preprocessing stage. The model will implemented using BERT algorithms and Logistic Regression. This is followed by the evaluation of the model, aimed at identifying the best performance from the previously developed model.

	sdg	label_ids	text	sentence_2	sentence_type
0	1	0	End hunger, achieve food security and improved...	No poverty	Headline
1	1	0	Ensure healthy lives and promote well-being fo...	No poverty	Headline
2	1	0	Ensure inclusive and equitable quality educati...	No poverty	Headline
3	1	0	Achieve gender equality and empower all women ...	No poverty	Headline
4	1	0	Ensure availability and sustainable management...	No poverty	Headline
...	...	...	...	...	...
76954	17	2	Developmentalities and Calculative Practices: ...	Partnerships for the goals	Labeled sample
76955	17	2	Responsible supply chain management- A key for...	Partnerships for the goals	Labeled sample
76956	17	2	The university and the city	Partnerships for the goals	Labeled sample
76957	17	2	Future socio-economic and environmental sustai...	Partnerships for the goals	Labeled sample
76958	17	2	Trialing the Community-Based Collaborative Act...	Partnerships for the goals	Labeled sample

76959 rows × 5 columns

Fig. 5: Load Dataset

4.1.1. Load Dataset

In Figure 5, dataset is used for model development. This dataset is obtained from a website called Huggingface. The dataset used in this study contains 76,958 article titles, which are used in the current model development.

4.1.2. Data Labelling

The Table 2, initial phase of data preparation involved Data Labeling, where each text instance from the dataset was assigned a specific label corresponding to one of the 17 United Nations Sustainable Development Goals (UN SDGs).

For instance, SDG 3 (Good Health and Well-being) represents the majority class with 20,948 samples, while SDG 1 (No Poverty) is a pronounced minority class with only 514 samples. This imbalance is typical for real-world SDG research data due to historical differences in research focus and funding. This heterogeneity in sample sizes necessitates a rigorous evaluation methodology, particularly the reporting of per-category performance metrics (Precision, Recall, F1-Score) in the Results section to ensure the model's true effectiveness on all goals is accurately assessed, rather than being biased by the majority classes on the Table 2.

Table 2: View of 17 SDG Categories Along with Sample Sizes

SDG	Samples	SDG	Samples
SDG 1	514	SDG 2	2401
SDG 3	20948	SDG 4	5223
SDG 5	6005	SDG 6	5376
SDG 7	4441	SDG 8	1619
SDG 9	4288	SDG 10	2605
SDG 11	3608	SDG 12	3608
SDG 13	3642	SDG 14	1959
SDG 15	5478	SDG 16	3659
SDG 17	1585		

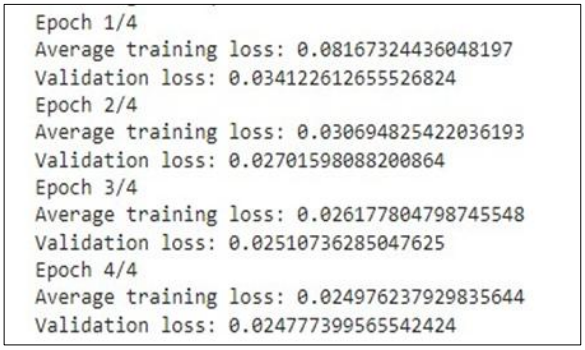
4.1.3. Preprocessing

In the preprocessing stage using the BERT algorithm, the model first accesses the dataset. Then, it initializes a one hot encoder to convert labels into the one hot encoding format. Next, the model utilizes the BERT and Logistics Regression base uncased model, which includes case folding to convert letters in words or sentences to lowercase. The NLTK library is used for further processing, including removing stop words and performing lemmatization to clean the text and normalize words to their base form. Following this, tokenization is performed to break down the text into smaller tokens

4.1.4. Training Model

In this stage, a batch size of 32 is used to compute gradients and update model parameters, while the model is trained for a total of 4 epochs. Figure 6 indicates that the average training loss and validation loss tend to decrease as training progresses. This trend suggests that the model is becoming better at recognizing patterns in both the training and validation data.

The proximity of the training loss and validation loss values indicates that the model generalizes well. This means it can produce good results on unseen data without overfitting (where the model performs well on training



```
Epoch 1/4
Average training loss: 0.08167324436048197
Validation loss: 0.034122612655526824
Epoch 2/4
Average training loss: 0.030694825422036193
Validation loss: 0.02701598088200864
Epoch 3/4
Average training loss: 0.026177804798745548
Validation loss: 0.02510736285047625
Epoch 4/4
Average training loss: 0.024976237929835644
Validation loss: 0.024777399565542424
```

Fig. 6: Output of Training Model

Cite: E. Surbakti et al., "The 17 UN Sustainable Development Goals: Classification of Research Topics Using BERT and Logistic Regression". Evergreen, 13 (01) 417-431 (2026). https://doi.org/10.5109/7411078.

data but poorly on new data) or underfitting (where the model fails to capture the underlying patterns of the data).

4.1.5. Evaluate Model

In the Figure 7 example of evaluate model result, the model's performance is assessed using several metrics including validation loss, precision, recall, F1-score, and accuracy. The process begins by using the evaluate function to obtain predictions from the validation data using the previously trained model. These predictions are then converted into labels based on a specified threshold, followed by probability calculations. Evaluation involves comparing the predicted results with the actual labels from the validation data. The results of the model evaluation are displayed in a classification report that covers 17 categories, using the macro average as a benchmark. Macro average is chosen because it provides a balanced view of the model's performance across each class, ensuring that the performance on minority classes is also considered.

Validation loss: 0.024777399565542424

	precision	recall	f1-score	support
SDG 1	1.00	0.13	0.23	107
SDG 2	0.99	0.82	0.90	490
SDG 3	1.00	0.98	0.99	4176
SDG 4	1.00	0.92	0.96	1022
SDG 5	0.99	0.91	0.95	1228
SDG 6	0.94	0.93	0.94	1102
SDG 7	0.93	0.90	0.92	906
SDG 8	0.97	0.71	0.82	329
SDG 9	1.00	0.89	0.94	832
SDG 10	0.99	0.82	0.90	511
SDG 11	0.99	0.87	0.93	725
SDG 12	1.00	0.88	0.93	698
SDG 13	1.00	0.88	0.94	709
SDG 14	0.99	0.77	0.87	391
SDG 15	1.00	0.91	0.95	1127
SDG 16	1.00	0.88	0.94	735
SDG 17	0.94	0.73	0.82	304
micro avg	0.99	0.90	0.94	15392
macro avg	0.98	0.82	0.88	15392
weighted avg	0.99	0.90	0.94	15392
samples avg	0.90	0.90	0.90	15392

Fig. 7: Example of Evaluate Model Result

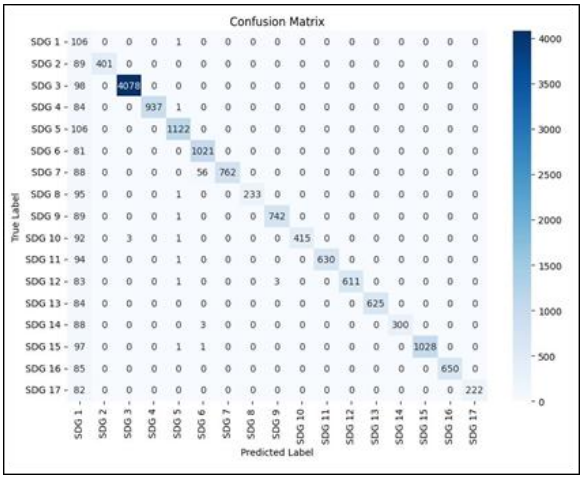


Fig. 8: Confusion Matrix result

After obtaining the results from the evaluate model function, the next steps typically include calculating accuracy and constructing a confusion matrix. The confusion matrix provides detailed information about the model's correct and incorrect predictions, aiding in understanding the classification model's performance. Figure 8 below shows the confusion matrix's results, which include true and predicted labels as well as the category with the most significant data. The precision, recall, F1-Score, and accuracy values can be calculated using the confusion matrix that is acquired. The confusion matrix provide the values of TP, FP, FN, and TN which is explained as follows.

a) The True Positive (TP) metric measures the number of accurate predictions in which the model correctly identified the positive class.
b) FN (False Negative): The quantity of prediction errors in which the model correctly identified the positive class as opposed to the negative class.
c) False Positive (FP): The quantity of prediction errors in which the model identified the negative class as the positive class.
d) TN (True Negative): The proportion of accurate predictions in which the model correctly identified the negative class as the real negative class.

In the Table 3 from the confusion matrix, we can compute precision, recall, F1-score, as well as determine the values of True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN). After getting the confusion matrix, the next step is to do the calculation to ensure the precision, recall, and f1-score values for each label. This is done using the following formula based on the confusion matrix.

Table 3: Confusion Matrix Each SDG Categories

SDG	TP	FP	FN	TN
SDG 1	106	1435	1	13850
SDG 2	401	0	89	14902
SDG 3	4078	3	98	11213
SDG 4	937	0	85	14370
SDG 5	1122	8	106	14156
SDG 6	1021	60	81	14230
SDG 7	762	0	144	14486
SDG 8	233	0	96	15063
SDG 9	742	3	90	14557
SDG 10	415	0	96	14881
SDG 11	630	0	95	14667
SDG 12	611	0	87	14694
SDG 13	625	0	84	14683
SDG 14	300	0	91	15001
SDG 15	1028	0	99	14265
SDG 16	650	0	85	14657
SDG 17	222	0	82	15088

4.2. BERT Model Testing

After completing preprocessing and model training stages, a sequence of trials and evaluations was conducted to optimize performance. The model underwent evaluation using dataset splits with varying ratios between training and testing data: 50% training and 50% testing, 60% training and 40% testing, 70% training and 30% testing, and 80% training and 20% testing. These four scenarios influenced the outcomes of the model evaluations performed during the trials. Each number in the numbering table from 1 to 4 indicates a different percentage split between training and testing data, which affects the model evaluation results during the trial process. Here are the details of the dataset splits for each number:

1) 50% training and 50% testing
2) 60% training and 40% testing
3) 70% training and 30% testing
4) 80% training and 20% testing

In each scenario, the varying percentage splits between training and testing data are used to test the model's performance in classifying previously unseen data. Below are the specific trial scenarios conducted.

a) BERT model without using remove stopwords, lemmatization, and sampling method.

In the Table 4, the dataset will be used without removing stop words (commonly occurring words) and without performing lemmatization (converting words to their base form). This approach allows the model to observe the effects of stop words and variations in word forms on its performance. The high performance demonstrated by the BERT models across the four trial scenarios, evidenced by reaching up to 90.27% and precision up to 0.98, necessitates a deeper discussion regarding the underlying mechanisms of the model.

Specifically, the results in Table 4 were achieved using the BERT model without standard preliminary text preprocessing techniques, such as stopword removal and lemmatization. The sustained high performance in this minimally preprocessed setup directly validates the core strength of the BERT (Bidirectional Encoder Representations from Transformers) architecture.

b) BERT model using remove stopwords and lemmatization but without sampling method.

In this scenario, the dataset will be processed by removing stopwords and applying lemmatization before training. This aims to enhance the quality of data provided to the model by eliminating irrelevant words and transforming words into their base forms. The result show in the Table 5.

c) BERT model using remove stopwords. lemmatization, and undersampling.

In this scenario, dataset on Table 6 under sampling will be applied to the training data to balance the number of samples between positive and negative classes. This approach can assist the model in addressing class imbalance issues and mitigate biases that may arise due to differences in sample sizes between classes.

d) BERT model using remove stop words, lemmatization, and oversampling.

In this scenario Table 7, oversampling will be applied to the training data to balance the number of samples between positive and negative classes. By increasing the number of minority class samples, the model can learn more effectively about that class and generate more accurate predictions.

Table 4: Confusion Matrix Each SDG Categories

Model	Precision	Recall	F1-Score	Accuracy (%)
Model 1	0.96	0.81	0.86	89.82
Model 2	0.98	0.81	0.86	89.88
Model 3	0.98	0.82	0.88	90.14
Model 4	0.98	0.82	0.88	90.27

Table 5: Confusion Matrix Each SDG Categories

Model	Precision	Recall	F1-Score	Accuracy (%)
Model 5	0.98	0.81	0.86	89.90
Model 6	0.99	0.82	0.88	90.43
Model 7	0.99	0.82	0.87	90.68
Model 8	0.98	0.82	0.88	90.20

Table 6: Confusion Matrix Each SDG Categories

Model	Precision	Recall	F1-Score	Accuracy (%)
Model 9	0.96	0.82	0.87	90.05
Model 10	0.98	0.81	0.86	89.90
Model 11	0.98	0.81	0.87	90.02
Model 12	0.98	0.82	0.87	90.44

Table 7: Confusion Matrix Each SDG Categories

Model	Precision	Recall	F1-Score	Accuracy (%)
Model 13	0.98	0.81	0.86	89.91
Model 14	0.98	0.81	0.87	90.12
Model 15	0.98	0.82	0.87	90.29
Model 16	0.98	0.81	0.87	90.20

Table 8: BERT Top High Accuracy from SGD Models

Model	Precision	Recall	F1-Score	Accuracy (%)
Model 4	0.98	0.82	0.88	90.27
Model 7	0.99	0.82	0.87	90.68
Model 12	0.98	0.82	0.87	90.44
Model 15	0.98	0.82	0.87	90.29

4.3. Evaluation

Based on the conducted experiments with four scenarios and four models per scenario, it can be concluded that the second scenario, using 70% of the training data from the entire dataset, performed the best. This scenario achieved the highest results based on the classification report and accuracy compared to the other scenarios. The model obtained a precision score of 0.99, a recall score of 0.82, an f1-score of 0.87, and an accuracy of 90.68%.

High precision indicates how accurately the model classifies text as positive predictions. The obtained recall score shows how well the model finds all true positive labels in the data. F1-score, as a combination of precision and recall, provides an overall picture of the model's performance. Meanwhile, accuracy indicates how accurately the model classifies text overall. Table 8 is a summary of the experiment results in tabular form, displaying the best model from each scenario with the highest precision, recall, f1-score, and accuracy:

4.4. Logistic Regression Testing Model

After training and initial testing, the model was subjected to a series of evaluation scenarios to determine its optimal performance. Multiple dataset split configurations were applied to assess the impact of different training-to-testing ratios. Specifically, datasets were partitioned as follows:

- 70% training / 30% testing,
- 75% training / 25% testing, and
- 80% training / 20% testing.

These configurations were used to evaluate how the size of the training set affects model accuracy and generalization. Two primary experimental scenarios were designed to evaluate the influence of data preprocessing techniques and data quality on model performance:

a) Scenario 1: Logistics Regression Evaluation Without Data Cleaning

In this scenario, the model was trained using uncleaned, raw data and three distinct text vectorization techniques:

- Count Vectorizer,
- Term Frequency–Inverse Document Frequency (TF-IDF), and
- Word2Vec.

In the Table 9, each technique was applied to datasets split using the 70:30, 75:25, and 80:20 ratios. The aim was to examine the model's performance when trained on noisy data and to assess the comparative impact of each vectorization method on the classification results. This scenario also allowed for evaluation of how raw textual inputs influence the logistic regression model's effectiveness when using different feature extraction methods.

b) Scenario 2: Logistics Regression Evaluation With Data Cleaning datasets.

In contrast to scenario 1, this experiment utilized cleaned the same three vectorization methods—Count vectorizer, TF-IDF, and Word2Vec—were applied to the cleaned datasets using the same training-testing splits (70:30, 75:25, 80:20). In the Table 10, the scenario aimed to determine whether data cleaning improves model performance and to what extent each preprocessing technique benefits from cleaner textual input.

Table 9: First Scenario: Logistics Regression Evaluation Without Data Cleaning

No	Model & Divide Dataset	Precision	Recall	F1-Score	Accuracy (%)
1	Count Vectorizer 70:30%	0.85	0.81	0.82	89.43
2	Count Vectorizer 75:25%	0.84	0.81	0.83	89.38
3	Count Vectorizer 80:20%	0.85	0.81	0.83	89.69
4	TF IDF 70:30%	0.85	0.82	0.83	89.73
5	TF IDF 75:25%	0.85	0.81	0.83	89.72
6	TF IDF 80:20%	0.86	0.82	0.84	90.01
7	Word2Vec 70:30%	0.39	0.32	0.33	49.69
8	Word2Vec 75:25%	0.4	0.33	0.34	50.69
9	Word2Vec 80:20%	0.39	0.33	0.34	50.51

Table 10: First Scenario: Logistics Regression Evaluation with Data Cleaning

No	Model & Divide Dataset	Precision	Recall	F1-Score	Accuracy (%)
1	Count Vectorizer 70:30%	0.85	0.81	0.83	89.45
2	Count Vectorizer 75:25%	0.85	0.81	0.81	89.40
3	Count Vectorizer 80:20%	0.86	0.81	0.83	89.68
4	TF IDF 70:30%	0.86	0.82	0.83	89.77
5	TF IDF 75:25%	0.85	0.81	0.83	89.73
6	TF IDF 80:20%	0.86	0.82	0.83	89.92
7	Word2Vec 70:30%	0.46	0.40	0.42	56.28
8	Word2Vec 75:25%	0.45	0.40	0.41	56.36
9	Word2Vec 80:20%	0.47	0.41	0.43	57.33

The two evaluation scenarios and nine models tested in each, the best performance was achieved in Scenario 1, Model 6**,** which used TF-IDF without data cleaning and an 80:20 training-testing split**.** This model achieved the highest scores, with 90.01% accuracy**,** 0.86 precision**,** 0.82 recall, and F1-score of 0.84. On the other hand, models using Word2Vec performed the worst in both scenarios. In Scenario 1, Word2Vec only reached 50.51% accuracy, with low precision 0.39, recall 0.33, and F1-score 0.34. Scenario 2 showed slight improvement, but performance remained relatively low. These results show that TF-IDF is more effective for classifying research titles in this context, especially compared to Word2Vec.

From evaluation method was then implemented using Receiver Operating Characteristics (ROC) and Area Under the Curve (AUC). These metrics were visualized through ROC curves for each of the two models under comparison. Figure 9 and Figure 10 shows the ROC and AUC results for the model using TF-IDF (Term Frequency-Inverse Document Frequency) combined with data cleaning and without data cleaning.

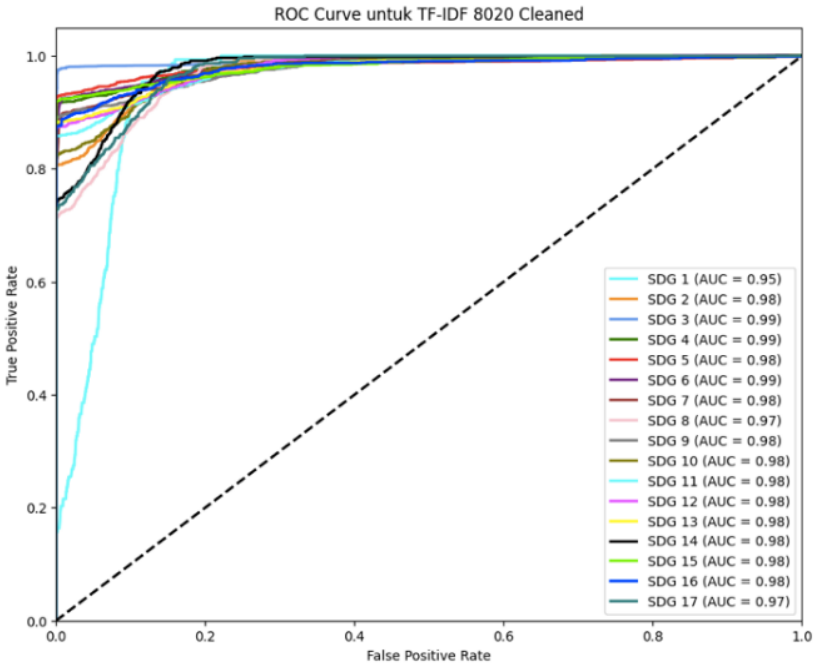


Fig. 9: ROC and AUC Results for the Model Using TF-IDF with Data Cleaning

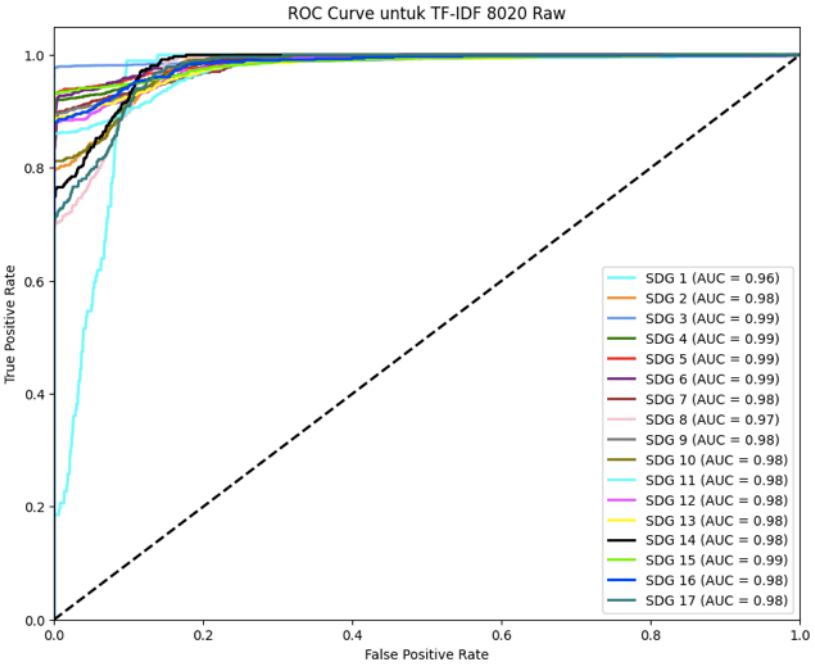


Fig. 10: ROC and AUC Results without Data Cleaning

The ROC curve effectively illustrates the model's discriminative performance in text classification, mapping the False Positive Rate (X-axis) against the True Positive Rate (Y-axis), with each line corresponding to one of the 17 SDG categories.The resulting AUC (Area Under the Curve) values are notably high, ranging from 0.95 to 0.99. This indicates the model, utilizing TF-IDF feature representation, has a strong ability to distinguish effectively between the different SDG classes. Specifically, the highest AUC score of 0.99 was achieved for SDG 3 (Good Health and Well-being), SDG 4 (Quality Education), and SDG 6 (Clean Water and Sanitation).

This strong performance is attributed to TF-IDF's efficacy in highlighting distinct keywords that are highly specific to individual, well-defined SDG categories, especially when paired with data cleaning to maximize feature relevance. However, it is important to note that TF-IDF provides only a frequency-based, sparse representation of the text, inherently lacking the deep contextual understanding and complex semantic pattern recognition capabilities found in transformer models like BERT. Therefore, while effective for classification based on prominent vocabulary, its ability to handle nuanced and highly contextual language may be constrained compared to the later BERT models discussed in the study.

In the Figure 11, the ROC curve shows the model's performance in classifying research titles into the 17 SDGs. The X-axis represents the false positive rate, and the Y-axis shows the true positive rate. Each line corresponds to one SDG category. The model achieved high AUC scores, ranging from 0.96 to 0.99, with the highest (0.99) seen in SDG 3, 4, 5, 6, and 15. These results are slightly better than those from the cleaned-data version, indicating strong classification performance even without data cleaning.

After generating the ROC and AUC results for both models, a learning curve test was conducted to check for overfitting. This test helps determine whether the model can generalize well to new data, rather than just performing well on the training data The learning curve shows that the model's performance improves as the training data increases. Initially, there was a large gap between training and validation scores, indicating overfitting. For example, at 10,000 samples, training accuracy was 0.82739, while validation was 0.75991. However, as the dataset grew beyond 30,000 samples, the scores became closer—at 40,000 samples, training was 0.90407 and validation was 0.89826—showing the model reached a good fit and could generalize well to new data.

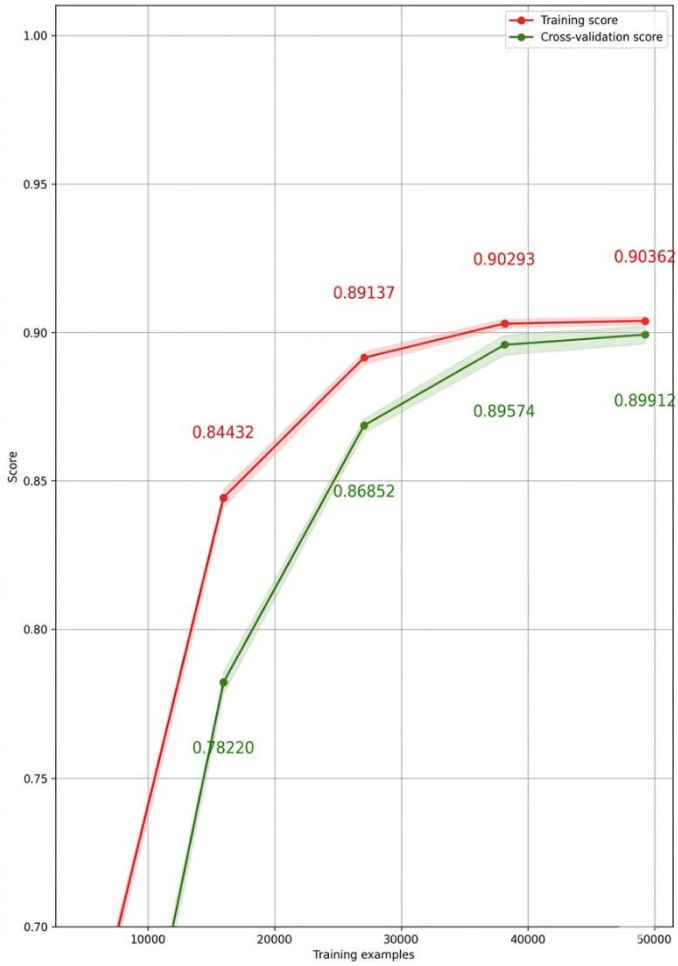


Fig. 11: Learning Curve Results for the Model Using TF-IDF with Data Cleaning

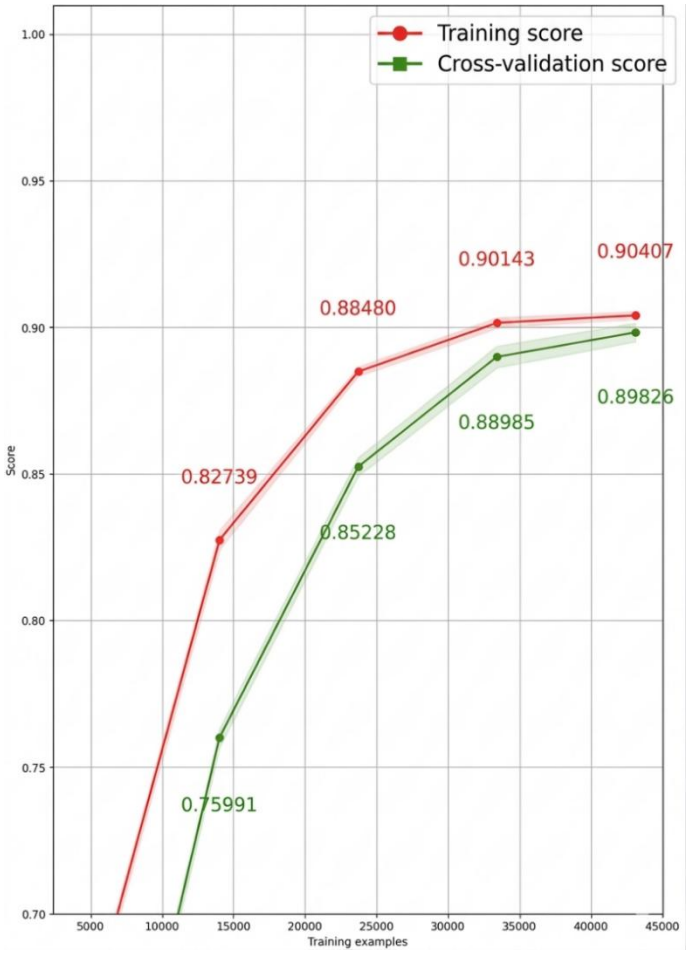


Fig. 12: Learning Curve Results for the Model Using TF-IDF without Data Cleaning

The learning curves demonstrate that the model's performance initially suffers from high variance (overfitting), evidenced by the significant gap between the high Training Score and the lower Cross-Validation Score when using fewer than 30,000 training examples. However, as the dataset size increases beyond 30,000 examples, the model's performance stabilizes, and the scores converge closely at 50,000 samples, indicating that the sufficient data volume effectively mitigates overfitting, leading to a robust model that generalizes well. The plateauing of both curves near the 0.90 mark suggests the model has reached its performance limit (constrained by bias) given its current architecture (TF-IDF features), meaning further increases in training data would yield negligible returns. This analysis confirms the high performance reported in the results is due to effective variance reduction from using a large dataset.

Figure 12 presents the learning curve of the TF-IDF model without data cleaning. The graph shows how the model performs during training (red line) and cross-validation (green line) as the training sample size increases from 5,000 to 50,000 examples. Initially, the training score rises sharply and then stabilizes as more data is added. Between 5,000 and 20,000 samples, there is a notable gap between the training and validation scores, indicating early signs of overfitting. For instance, with 10,000 training samples, the training score reaches 0.84432 while the cross-validation score is 0.78220—a considerable difference.

However, as the number of training samples increases beyond 30,000, the gap between training and validation scores narrows. At 50,000 samples, the training score is 0.90362 and the cross-validation score is 0.89912, indicating minimal difference. This suggests that the model generalizes well with larger datasets and does not suffer from overfitting.

Moreover, the learning curve indicates that after a certain point, adding more data yields diminishing returns in terms of performance improvement. Overall, the model can be classified as a *good fit*—neither overfitting nor under fitting. Additionally, ROC and AUC results support this conclusion, showing that both models perform well in classification tasks. However, the TF-IDF model without data cleaning demonstrates slightly better performance.

5. Conclusion

This study successfully implemented BERT and Logistic Regression models to classify university research outputs according to the 17 UN Sustainable Development Goals (SDGs) using a dataset of 76,958 records. The BERT-based model 4 epochs, learning rate 2e-5, batch size 32

achieved superior performance with 90.68% accuracy, 0.99 precision, 0.82 recall, and an F1-score of 0.87. While the Logistic Regression model (TF-IDF, L1 penalty, SAGA solver) offered a simpler, more interpretable alternative with 90.01% accuracy, paired t-test results confirmed that BERT's superior performance was statistically significant ($p < 0.05$). Despite limitations in generalizability to ambiguous data and lower interpretability compared to Logistic Regression, the BERT-based approach provides a robust framework for identifying institutional research strengths. By automating SDG alignment, these models facilitate data-driven resource allocation and international collaboration, ultimately accelerating global progress toward the UN SDGs.

Acknowledgment

The authors would like to express their sincere gratitude to University Multimedia Nusantara (UMN) for the support and resources provided throughout this research. This study would not have been possible without the facilities, guidance, and encouragement from the university's academic and research community.

References

1) M. D. Abdulrahaman, N. Faruk, A. A Oloyede, N. T Surajudeen-Bakinde, L. A Olawoyin, O. V Mejabi, Y. O Imam-Fulani, A. O Fahm, and A. L Azeez, "Multimedia Tools in the Teaching and Learning Processes: A Systematic Review," Heliyon (2020). doi: https://doi.org/ 10.1016/j.heliyon.2020.e05312.
2) I. Mergel, N. Edelmann, and N. Haug, "Defining digital transformation: Results from expert interviews," Government Information Quarterly, vol. 36, no. 4, p. 101385 (2019). doi : https://doi.org/10.1016/j.giq.2019.06.002.
3) A. Qingyao, T. Bai, Z. Cao, Y. Chang, and J. Chen. "Information Retrieval Meets Large Language Models: A Strategic Report from Chinese IR Community," AI Open volume 4 80-90 4 (2023). doi: https://doi.org/10.1016/j.aiopen.2023.08.001.
4) U. Hanani, B. Shapira, and P. Shoval. "Information Filtering: Overview of Issues, Research and Systems." User Model. User-Adapted Interaction 11 203-259 (2001).doi: https://doi.org/10.1023/A:1011196000674
5) M. Chankseliani, and T. McCowan, "Higher education and the Sustainable Development Goals," Higher Education, vol. 81 no. 1 pp. 1–8 (2021). doi : https://doi.org/10.1007/s10734-020-00652-w.
6) "Take Action for the Sustainable Development Goals - United Nations Sustainable Development." https://www.un.org/sustainabledevelopment/sustainable-development-goals/ (accessed December 25, 2024).
7) J. Liu, W. C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multilabel text classification," Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval pp 115–124 (2019). doi:https://doi.org/10.1145/3077136.3080780.
8) S. R. Medina, "Multi-Label Text Classification with Transfer Learning for Policy Documents The Case of the Sustainable Development Goals," Uppsala Universitet Publications. (2019). https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-395186
9) E. C. Garrido-Merchan, R. Gozalo-Brizuela, and S. Gonzalez-Carvajal, "Comparing BERT Against Traditional Machine Learning Models in Text Classification," Journal of Computational and Cognitive Engineering, vol. 2, no. 4, pp. 352–356, (2023), doi: 10.47852/bonviewJCCE3202838.
10) E. E. Surbakti, B. Purwandari, I. Solichah, and L. Kumaralalita, "Analysis of software development method selection: A case of a private financial institution," ACM Int. Conf. Proceeding Ser., pp. 168–173 (2019). doi: 10.1145/3361785.3361806.
11) C. Maximiliano, E. E. Surbakti, R. Winantyo, Jantianus, M. Vasty, and W. Istiono, "Long Short-Term Memory Networks for Predicting LQ45 Index Trends in Infrastructure Stocks," 2024 9th IEEE Int. Conf. Adv. Robot. Mechatronics, pp. 57–61 (2024). doi: 10.1109/ICARM62033.2024.10715962.
12) K. Ravikant, "Text-classify: A comprehensive comparative study of logistic regression, random forest, and knn models for enhanced text classification performance," International Journal of Advances in Engineering & Technology (2023). doi: https://doi.org/10.5281/zenodo.10148008
13) S. A. Bahtiar, C. K. Dewa, and A. Luthfi, "Comparison of naıve bayes and logistic regression in sentiment analysis on marketplace reviews using rating-based labeling," Journal of Information Systems and Informatics vol. 5 no 3 pp 915–927 (2023). doi: 10.51519/journalisi.v5i3.539
14) A. Hajikhani and A. Suominen, "Mapping the sustainable development goals (sdgs) in science, technology and innovation: application of machine learning in SDG-oriented artefact detection," Scientometrics vol. 127 pp. 6661–6693 11 (2022). doi: https://doi.org/10.1007/s11192-022-04358-x
15) P. Yadav, I. Kashyap, and B. S. Bhati, "Impact of Double Negation through Majority Voting of Machine Learning Algorithms," Joint Journal of Novel Carbon Resource Sciences & Green Asia Strategy Vol 11 Issue 01, pp331-342 (2024). doi: https://doi.org/10.5109/7172289.

16) U. Gurnani, S. K. Singh, M. K. Sain, and M. L. Meena, "Musculoskeletal Health Problems and their Association with Risk Factors among Manual Dairy Farm Workers," Joint Journal of Novel Carbon Resource Sciences & Green Asia Strategy vol. 9 no. 4 pp 950–961(2022) doi: 10.5109/6622881.
17) X. Zhou, R. Gururajan, Y. Li, R. Venkataraman, X. Tao, G. Bargshady, P. D. Barua, and S. Kondalsamy-Chennakesavan, "A survey on text classification and its applications," Web Intelligence vol. 18 no. 3 pp 205–216(2020). doi:10.3233/WEB-200442
18) X. Luo, "Efficient english text classification using selected machine learning techniques," Hunan University of Technology and Business (2021). doi:10.1016/j.aej.2021.02.009.
19) A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, "Survey on text classification algorithms: From text to predictions," Information Switzerland, vol. 13 no 2 (2022). doi: https://doi.org/10.3390/info13020083.
20) J. Grace, A. Maneengam, P. Kumar, and J Alanya-beltran. "Design and Implementation of Machine Learning Modelling through Adaptive Hybrid Swarm Optimization Techniques for Machine Management," Joint Journal of Novel Carbon Resource Sciences & Green Asia Strategy Vol 10 Issue 02 pp 1120-1126 (2023). doi: https://doi.org/10.5109/6793672.
21) A.S. Talaat, "Sentiment analysis classification system using hybrid BERT models", J Big Data 10 110 (2023). https://doi.org/10.1186/s40537-023-00781-w
22) A. F, Bangi, "Exploring the Role of Transformers in NLP: From BERT to GPT-3," International Research Journal of Engineering and Technology, pp 243–251, (2023).
23) E. C. Garrido-Merchan, R. Gozalo-Brizuela, and S. Gonzalez-Carvajal, "Comparing BERT against Traditional Machine Learning Models in Text Classification," Journal of Computational and Cognitive Engineering no. 1 (2023), doi: https://doi.org/10.47852/bonviewJCCE3202838.
24) N. D. Robert, C.Y Peng, R, Khavari, " HHS Public Access," Physiology behavior vol. 176 no. 3 pp. 139–148(2019). doi: https://doi.org/10.1002/jnr.23963.
25) "Natural language processing (NLP): What it is and why it matters." https://www.sas.com/en us/insights/analytics/what-is-natural-language-processing-nlp.html (accessed January 14, 2025)
26) "Using natural language processing to improve everyday life u-m information and technology services." https://its.umich.edu/news/ article/using-natural-language-processing-improve-everyday-life (accessed December 29, 2024)
27) H. Zhang and M. O. Shafiq, "Survey of transformers and towards ensemble learning using transformers for natural language processing," Journal Big Data 11 25 (2024). doi: https://doi.org/10.1186/s40537-023-00842-0
28) "What is the BERT language model? | definition from TechTarget.com."https://www.techtarget.com/ searchenterpriseai/definition/BERT-language-model. (accessed December 29, 2024)
29) "Global Strategic Institute for Sustainable Development — Department of Economic and Social Affairs." https://sdgs.un.org/ partnerships/global-strategic-institute-sustainable-development. (accessed December 22, 2024)
30) R. Yao, M. Tian, C.U. Lei, and D. K. W. Ciu, "Assigning multiple labels of sustainable development goals to open educational resources for sustainability education," Educ Inf Technol 29 18477–18499 (2024). doi: https://doi.org/10.1007/s10639-024-12566-6.
31) D. F. Hsu, M. T. LaFleur, and I. Orazbek, "Improving SDG classification precision using combinatorial fusion," Sensors vol. 22 no. 3 art no 1067 (2022), doi : https://doi.org/10.3390/s22031067.
32) S. Xu, "Performance evaluation of seven multi-label classification methods on real-world patent and publication datasets," Journal of data and information Science vol. 9 no 2 pp. 81–103(2024). DOI:10.2478/jdis-2024-0014
33) A. Das, "Logistic Regression," Springer International Publishing pp. 1–2. doi: https://doi.org/10.1007/978-3-319-69909-7 1689-2.
34) M. Adi Widyatmika and N. B. Bolia, "Unveiling Segregation and Composting Behavior in Urban Communities: A Study Case of Sarbagita Municipality, Indonesia," Joint Journal of Novel Carbon Resource Sciences & Green Asia Strategy Vol 11 Issue 02 pp 612-623 (2024). doi: https://doi.org/10.5109/7183356.
35) A. Subasi, "Chapter 3- Machine learning techniques," Academic Press pp. 91–202. 2020. https://www.sciencedirect.com/science/ article/pii/B9780128213797000035
36) "Text classification. using logistic regression,". https://medium.com/@ashins1997/ text-classification-dfe370bf7044. (accessed December 20, 2024)