Studying GDPR and Classifying it into Privacy Requirements with NLP, Machine Learning, and BERT

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

In a world where data privacy and security are ongoing issues, the European Union (EU) enacted a game-changing piece of legislation known as the General Data Privacy Regulation (GDPR) in 2018. This legislation stands as a momentous stride in the realm of data protection and law enforcement. As highlighted by Štarchoň and Pikulík (2019), the GDPR is a complicated and all-encompassing legal framework. It comprises a series of regulations, each meticulously addressing distinct sides encompassing data protection, privacy, accountability, and the responsibilities associated with data compliance. This landmark law not only strengthened peoples' rights over their personal information but also established a thorough structure that businesses must follow while managing this priceless resource (Sharma, 2019).

In this Study, I aim to figure out the GDPR's complex structure by looking at its content, implications, and, most importantly, implementation in the context of software and data-driven systems. It helps to look at how cutting-edge technology, such as machine learning, BERT, and natural language processing, may be utilized to help GDPR compliance and enhance data security practices via a multidisciplinary lens.

The usefulness and efficacy of GDPR implementation are at the foundation of the research topics that frame this study. I want to offer insights into the most effective methods for dividing GDPR legislation into privacy requirements by examining the interactions between conventional machine learning algorithms and cutting-edge language models, such as BERT and GPT. The Study aims to unveil the layers of GDPR, unraveling its significance and implications in an increasingly data-centric world.

1.1. Problem Description

The problem addressed in this Study revolves around the necessity of categorizing GDPR into separate privacy requirements. Since the GDPR has global implications for companies that handle the personal data of EU people, compliance with it is non-negotiable for organizations throughout the world.

The manual classification of GDPR rules is a time-consuming, labor-intensive, and error-prone procedure that mostly depends on legal professionals. To achieve compliance, legal professionals must carefully analyze, interpret, and classify GDPR legislation. This manual method has the following drawbacks, though:

Resource-intensive: Hiring legal professionals for this job necessitates a large investment of both financial and human resources, particularly for businesses dealing with a heavy workload of GDPR rules.

Subjectivity and Inconsistency: Human interpretation is inherently subjective and might lead to uneven categorizations, which could result in compliance issues and legal dangers.

Inefficiency: The immense number volume and complexity of GDPR make manual categorization an inefficient process, often resulting in delays in compliance efforts.

Scalability: As GDPR develops and produces new requirements, it is more and harder to scale manual classification.

To overcome these difficulties, machine learning offers a promising approach (Deng, Hinton, and Kingsbury, 2013). Machine learning models may be taught to automatically categorize new regulations into predetermined privacy requirements using past GDPR data, which considerably lowers the compliance effort and improves accuracy (Venkatesh and Anuradha, 2019).

The two issues that this Study focuses on are:

Automating GDPR Categorization: Establishing an accurate and reliable automated system that can replace or improve manual efforts in classifying GDPR into distinct privacy requirements.

Selecting Effective Techniques: Identifying the most effective feature representation techniques and machine learning algorithms for this specific task. This involves understanding how different approaches impact accuracy, efficiency, and scalability.

By tackling these issues, the research hopes to make a positive impact on the larger field of data privacy and compliance and offer insightful information on automating difficult legal categorization jobs. Additionally, it investigates how sophisticated language models, such as BERT, could improve the effectiveness of GDPR compliance initiatives (Liu et al., 2021).

1.2. Research aim and objective

The aim of this Study is to investigate and develop automated methods for categorizing General Data Protection Regulation (GDPR) into distinct privacy requirements, thereby enhancing the efficiency and accuracy of GDPR compliance efforts.

The following research questions are addressed in this study:

RQ1: How does the combination of different Vectorization methods and machine learning algorithms impact the overall accuracy of classifying GDPR into privacy requirements?

RQ2: Do the traditional machine learning algorithms, such as SVM, decision trees, logistic regression, and random forest demonstrate superior performance compared to pre-trained language models like BERT in classifying GDPR into privacy requirements?

To realize this overarching aim, seven specific research objectives are formulated below:

1. **AI-Driven Data Extraction:** Collect and clean relevant data using advanced AI techniques.
2. **Exploratory Data Analysis (EDA):** Investigate the dataset thoroughly to comprehend its features.
3. **Text Preprocessing using NLP:** Using Natural Language Processing (NLP) approaches, the text is pre-processed to ensure its quality and consistency as it is being prepared for analysis.
4. **Feature Representation Evaluation:** Compare two approaches, like as Count Vectorizer and TF-IDF, to determine which is best for GDPR language analysis.
5. **Model Selection:** Investigate several machine learning models (Decision Trees, Logistic Regression, SVM, and Random Forests) to suggest the best appropriate for GDPR categorization.
6. **Benchmarking Advanced Language Models:** To evaluate the benefits and drawbacks of various models for categorizing privacy, compare modern models like BERT with traditional models.
7. **Practical Implications:** Convert research results into tips that businesses can use to enhance their GDPR compliance and data protection practices.

1.3. Research Approach

The research approach for this study adopts a quantitative analysis methodology, primarily driven by the utilization of a small and structured dataset. The study technique begins with a thorough descriptive and visual examination of the information, with the goal of acquiring a thorough grasp of the distribution and features of each variable. Various preparation procedures are used to ensure the dataset's quality and suitability for analysis, including text cleaning and the application of natural language processing algorithms to prepare the text data for further analysis.

Several machine learning models like Logistic Regression, Random Forest, SVMs, Decision Trees, and the complex BERT model are considered for the core task of categorizing GDPR legislation into privacy requirements. To get insights into their performance, rigorous model evaluation is performed using known criteria like accuracy, F1 score, precision, and recall.

The last phases of the research entail a thorough comparison and analysis of the findings collected from various models in order to establish the most effective way for categorizing GDPR laws into privacy requirements. Finally, this Study presents the facts and insights generated from the research in a systematic manner, ending in a detailed conclusion that clearly summarises the important takeaways and contributions of the study.

1.4. Study Outline

The Study's chapter summaries are as follows:

Chapter 1: Introduction

Introduces the study subject and its relevance, describes the research's goals and methods, and establishes the overall framework for the Study.

Chapter 2: Literature Review

Examines relevant research, including articles on GDPR laws, artificial intelligence, and language models like BERT and GPT. Determine where there are research gaps and how my work can fill them.

Chapter 3: Methodology

The study approach is described in detail, including data collection, exploratory data analysis, feature representation evaluation, model selection, and NLP-based data preparation.

Chapter 4: Evaluation

This chapter focuses on the application of my study. To guarantee openness in the approach, it provides a detailed explanation of the experimental setup, including all required hardware and software. The experimental designs are interpreted using visual aids such as tables and figures.

Chapter 5: Results and Analysis

In this part, A detailed comparison of the effectiveness of several machine learning models employing various vectorization approaches and the BERT model for classifying GDPR into privacy needs is given.

Chapter 6: Threats to Validity

The possible biases or inaccuracies that may impact the correctness of the results, including the dangers to both internal and external validity are addressed in this chapter.

Chapter 7: Conclusion

I conclude by summarising the Study results, contributions, and application. I also discuss my personal development and point up potential future study possibilities.

CHAPTER 2: LITERATURE REVIEW

This chapter is an investigation of the existing literature relevant to the main themes of the Study: the General Data Protection Regulation (GDPR), natural language processing (NLP), machine learning (ML), and bidirectional encoder representations from transformers (BERT), as well as their interactions about data privacy and legal compliance. The structure of the review of the literature is to provide readers with a thorough comprehension of the fundamental ideas and investigative results that form the basis of the investigation.

2.1. GDPR Documents and Regulations

The GDPR's goal is to offer customers greater control over their data while also harmonizing data protection rules among EU member states. The framework of the GDPR has 99 articles that define the criteria for safeguarding personal information and privacy throughout the EU (Intersoft Consulting, 2018). Principles, human rights, the legal basis for processing, notification of data breaches, cross-border data transfers, and other issues are covered. The EU's organized approach ensures a distinct and uniform set of data protection requirements.

The Voigt and von dem Bussche (2017) guide provides an extensive examination of the GDPR's framework and a thorough knowledge of its articles and recitals. It goes through the GDPR's history, application, guiding principles, and data subject rights. They also cover numerous implementation and application-related elements of GDPR.

Each of the numerous elements that make up the GDPR covers a different aspect of data protection:

Preamble: An explanation of the objectives and purpose of the GDPR that places a strong emphasis on the need to safeguard personal data.

Articles: It is divided into 11 chapters, each of which has several articles outlining precise guidelines, regulations, and provisions for handling data, as well as details on how to enforce such rights.

Recitals: Explanatory remarks that offer the articles' background, interpretation, and rationale. The implementation and comprehension of the GDPR rules are guided by these recitals.

2.2. Generative Pre-trained Transformer (GPT)

The chatbot GPT is a powerful natural language processing model that has attracted a lot of interest for its capacity to comprehend and produce human-like text answers. GPT models were created by OpenAI and are based on the transformer architecture, which allows them to efficiently capture contextual connections in text data. GPT models may understand patterns, semantics, and even subtleties found in human language since they are trained on enormous volumes of text data.

Understanding Chatbot GPT:

Chatbots and conversation systems are only a few of the many language activities for which GPT models are built. They are "unsupervised" models, which means they pick up knowledge from huge text corpora without the need for special task-specific annotations. The pre-training step gives GPT models a comprehensive command of language.

Input Extraction with GPT:

The capacity of GPT models to extract useful information from user inputs is one of its notable characteristics. Even when the input is complicated or complex, GPT models can comprehend the context and purpose behind a user's communication. This is so because GPT models are taught to anticipate the next word in a phrase, which by its very nature necessitates comprehension of word meaning and context.

The study by Biswas (2023) explains how Military applications could require input in the form of conversations between multiple languages that need to be translated, training manuals, or intelligence reports. The input might also include unstructured data from sources like news, social media, or other information sources that have to be analyzed to draw conclusions that can be utilized to assist military operations, such as sensor data from unmanned systems like drones or satellite photography. Training ChatGPT on a broad and varied dataset pertinent to the particular job at hand may increase the accuracy of its output text.

The text-preprocessing stage of sentiment analysis in the research by Tri Julianto et al. (2023) uses Chat GPT. To do this, you must use the "Give label the sentences" instruction to label Positive, Neutral, and Negative text. The authors contend that Chat GPT can increase sentiment analysis models' accuracy by producing more insightful and contextually appropriate text labels. Additionally, compared to conventional approaches, using Chat GPT for text preprocessing is quicker and takes less time.

It is crucial to remember that ChatGPT is not flawless and that it may produce erroneous or biased text, particularly if it has been trained on faulty or incomplete data. As a result, it is crucial to carefully assess ChatGPT's generated text's correctness and utilize it as a tool to aid in decision-making rather than as a replacement for human judgment.

2.3. Text Extraction Process

A crucial step in converting unstructured textual material into a structured format appropriate for analysis is text extraction. Here, the importance of text extraction as well as the methods used is examined.

2.3.1. Natural Language Processing

GDPR is written in unstructured text. Therefore, it needs techniques to structure it. To transform the unstructured textual data into a structured format appropriate for data analysis, natural language processing must be done. This is because natural language text data is frequently disorganized and challenging to analyze using conventional data analysis approaches. Jindal, Malhotra, and Jain's (2016) work presents an approach for extracting security requirements from documents through NLP by eliminating unnecessary words and lowering the size of the feature set by using pre-processing techniques including tokenization, stop words removal, and stemming, which makes it simpler to analyze the data. Also, the semantic content, syntactic structure, and presence or absence of terms linked to privacy are all detectable from user stories using attributes that are extracted using NLP approaches (Casillo, Deufemia, and Gravino, 2022)

In the recent work of Sangaroonsilp et al. (2023a), privacy-related issues are extracted from the dataset, and after that, the rundown and depiction of issues with the label(s) of pertinent security prerequisites are utilized as input writings. Distinctive printed highlight extraction and learning procedures are connected to create a vector representation of writings, and those vectors and protection condition names are at that point boosted to a classifier for preparation and approval. The study investigates six prevalent methods, to be specific Bag-of-Words (BoW), N-gram Inverse Document Frequency (N-gram IDF), Term Frequency-Inverse Document Frequency (TF-IDF), Word2Vec, Convolutional Neural Network (CNN), and Bidirectional Encoder Representations from Transformers (BERT) to perform the classification on privacy-related issue reports in Google Chrome and Moodle projects.

The TF-IDF vectorizer could be a broadly utilized procedure in NLP for data recovery and text mining (Munaiah, Meneely, and Murukannaiah, 2017). It is considered one of the most excellent methods for speaking to content information in a vector space since it takes into consideration the significance of each word in a report and overall documents in a corpus. Das, Kamalanathan, and Alphonse (2021) suggested a model for sentiment analysis that took into account the two features N-Grams and TF-IDF on the IMDB movie reviews and Amazon Alexa reviews dataset. They discovered that TF-IDF obtained the highest accuracy, precision, recall, and F1-score value in the Random Forest classifier when they used TF-IDF for feature selection and modified the traditional TF-IDF algorithm formula by including the concept of intra-class dispersion and excluding the inner impact to disturb characteristic.

Count Vectorizer, sometimes referred to as the Bag-of-Words (BoW) method, is a key method in natural language processing (NLP) for transforming text input into a numerical representation appropriate for machine learning algorithms. Count Vectorizer has been employed as a feature extraction method in several text classification applications, including sentiment analysis, spam detection, topic modeling, and others.

In the study by Hu, and Zhang (2022), it was discovered that Count Vectorizer and machine learning algorithms can be combined to achieve high accuracy in text classification. It was also noted that the quality of feature extraction is particularly crucial in text classification and that this combination can produce competitive performance when classifying text data.

2.3.2. Machine Learning

In software engineering, non-functional needs may be automatically detected using machine learning. As they frequently rely on human evaluation of requirements papers by specialists, traditional techniques of detecting non-functional needs can be time-consuming and error-prone. By building a model to identify trends in the language and structure of requirements documents that are suggestive of non-functional requirements (NFR), machine learning offers a technique to automate this process. This can reduce the amount of time and work required as well as potentially increase the consistency and accuracy of detecting non-functional requirements.

Non-functional requirements can be integrated into the initial architectural design if they are discovered early rather than being refactored at a later time. The method outlined in the study conducted by Cleland-Huang et al. (2007) uses machine learning's NFR classifier to identify and categorize stakeholders' quality complaints. A set of labeled data that contains instances of non-functional criteria about security, performance, and usability is used to train the NFR classifier. A candidate list of NFRs may be automatically retrieved from free-form documents like meeting minutes, interview notes, and memoranda once the NFR classifier has been trained. During the requirements analysis process, stakeholders may easily discover and rank quality issues by using the NFR classifier's ability to categorize the retrieved NFRs according to their quality problems.

An ensemble learning method known as Random Forest has become more well-liked because of its capacity to lower overfitting and raise classification accuracy. To provide more reliable forecasts, it blends different decision trees. Breiman (2001) ground-breaking work on Random Forest, introduced an original ensemble learning method that completely changed the area of machine learning. Their research focused on the shortcomings of individual decision trees while processing large, complicated information. While straightforward and simple to understand, decision trees can suffer from overfitting, where they fail to generalize to new instances and capture noise in the data.

According to their study, Random Forest improves accuracy by lowering the variation associated with individual decision trees. The result is more dependable and steadier when these several projections are added together. Furthermore, Random Forest is particularly useful for real-world applications since it can manage noisy and high-dimensional datasets.

Further research conducted by Liaw and Wiener (2002) further demonstrated the potency of Random Forest. They emphasized its uses in a variety of fields, such as bioinformatics, where precise categorization of biological data is essential. This flexibility demonstrated Random Forest's capacity to learn from a wide range of datasets and problem areas.

Due to its simplicity of understanding, interpretation, and implementation, decision tree algorithms are frequently employed in machine learning categorization. They can manage missing values and noisy data, and they are efficient in managing both category and numerical data. Charbuty and Abdulazeez (2021) discuss the hierarchical organization, numerous types, and important ideas like entropy and information gained about the algorithm. Decision tree classifiers' accuracy varies based on the dataset and the chosen method. The accuracy rates in the studies ranged from 58.11% to 99.93%. The decision tree algorithm's greatest accuracy is 99.93% when a machine learning repository is used as the dataset.

Dealing with high-dimensional and sparse feature spaces, where each word or phrase becomes a feature, is one of the main difficulties in text categorization. SVM is especially well suited for text classification problems due to its capacity for handling high-dimensional input and efficacy in determining the ideal decision boundary.

The application of SVMs for text categorization problems is thoroughly examined by Cervantes et al. (2020). He discussed several SVM-based text classification-related topics, including feature extraction, kernel functions, and optimization techniques. He also gave an overview of the potential and difficulties associated with using SVM to classify various kinds of text data. It covers the benefits and drawbacks of SVM for text classification as well as the function of ensemble methods and hybrid approaches that combine SVM with other methods to enhance classification performance.

Given its amazing qualities of simplicity, interpretability, and the production of probabilistic results, the logistic regression model continues to be a vital pillar in the world of classification problems. Zou et al. (2019) developed a model that makes use of the logistic function to represent the complex interaction between binary outcomes and predictor factors. This calls for the creation of methods to improve the performance of the model by addressing issues like overfitting and parameter adjustment. This also includes a perceptive case study that applies the improved logistic regression model to a real-world setting.

In another study, Thabtah, Abdelhamid, and Peebles (2019), investigated the classification of autism through the use of machine learning methods, in particular logistic regression analysis. It deals with the difficulty of correctly classifying somebody as having autism or not based on a variety of characteristics. They gathered a variety of data aspects from people, including behavioral attributes, medical history, and demographic data. The logistic regression model, which calculates probabilities for classifying autism, uses these characteristics as inputs. The study's importance can be attributed to its prospective effects on early autism diagnosis and treatment plans.

2.3.3. Bidirectional Encoder Representations from Transformers (BERT)

A major development in natural language processing is the approach, created by Google AI. BERT surpasses unidirectional models in its comprehension of words within context through the use of bidirectional context understanding. This development is crucial for text classification, as labeling text materials necessitates an understanding of context. BERT's performance for classification tasks is improved by fine-tuning its pre-trained knowledge of task-specific data.

BERT is introduced in the study by Devlin et al. (2019), who goes into depth about its design, pre-training, and fine-tuning procedures. It highlights the value of BERT in text categorization and demonstrates how it has transformed how text is understood and categorized.

By applying a pooling operation to the BERT model's output, Elluri et al. (2021) use it to embed sentences from policy papers in the corpus. The pre-trained BERT model has 110 million parameters, 12 layers of transformer blocks, and 12 attention heads. The output of the BERT model creates a matrix using the formula N = number of sentences, W = number of tokenized words, and E = dimensions of embedding. The outcomes, however, do not offer phrases with optimal embeddings. For improved representation, the produced output embeddings are averaged out to produce an N E matrix in the N-2 layer.

The study by Van Hofslot et al. (2022) discusses utilizing deep learning models to automatically identify legal issues in cookie banner messages. To find legal infractions, the authors analyzed a collection of cookie banners that had already been annotated by five professionals. To automatically classify legal infractions in this dataset, they tried several cutting-edge deep learning models, including BERT, LEGAL-BERT, and BART. They also paired a dictionary-based strategy, i.e., LIWC embeddings with BERT, to see whether performance would be improved. The study's findings imply that for all classes that need to be discovered, no single model performs better than the others. BERT and LEGAL-BERT perform well for all classes in general, although they are also impacted by the skewed data distribution for some classes. Contrarily, BART performs the poorest for the majority of the classes but is unaffected by the limited dataset size and the unequal distribution of classes.

2.4. Research GAP

The research fills a huge gap in the corpus of knowledge already available on GDPR compliance and the use of sophisticated language models like GPT. While there has been a significant amount of study on the GDPR itself, machine learning algorithms, and text classification techniques, there has been a notable lack of research that integrates these fields to effectively and properly convert GDPR laws into privacy requirements.

Integration of GPT in GDPR Compliance: The incorporation of sophisticated natural language processing models like GPT has not been extensively studied in prior studies on GDPR compliance. GPT is a useful tool for understanding and classifying complicated legal documents because it can comprehend and provide text replies that resemble those of humans. This gap is filled by my study, which shows how GPT may be used for input data extraction in the context of GDPR compliance.

Comparison with Pre-trained Language Models: Although BERT has been used for several NLP tasks, including text classification, little research has been done on how successful it is at categorizing GDPR rules into privacy requirements. My research directly contrasts traditional machine learning algorithms with trained language models like BERT, revealing which strategy is more appropriate for this particular problem.

Optimal Feature Representation: Prior research frequently falls short of a thorough analysis of various feature representation strategies tailored to GDPR content. To assist practitioners in choosing the most suitable strategy, my study systematically evaluates the performance of techniques like Count Vectorizer and TF-IDF Vectorizer for GDPR language.

In summary, the Study fills the research gap by combining GDPR compliance, machine learning, and advanced language models, offering real-world GDPR compliance solutions while illuminating the relative efficacy of various techniques.

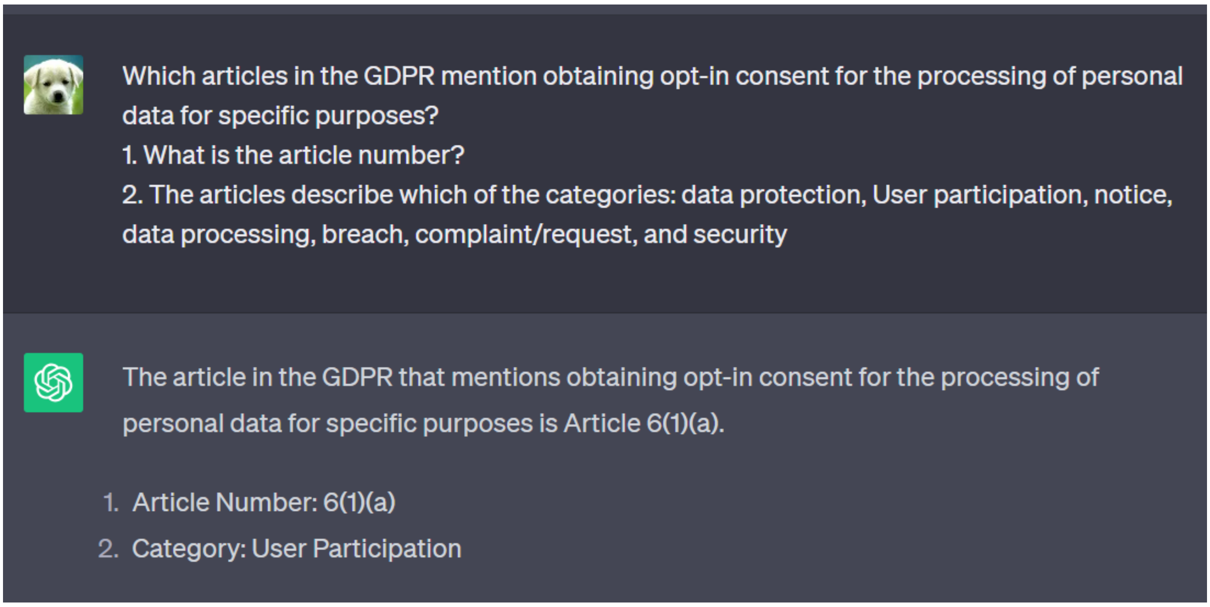
CHAPTER 3: METHODOLOGY

The methodology used in this study uses machine learning approaches to extract, analyze, and systematically classify privacy needs. It also looks at integrating the BERT model, which is renowned for its contextual awareness skills. The examination of privacy needs is streamlined with this method, enabling organizations to better manage data protection, regulatory compliance, and decision-making procedures. The process for the project is broken down into many crucial phases.

3.1. Data Collection

The primary data source for this project was the work of Sangaroonsilp et al. (2023b), which provided an assortment of privacy requirement descriptions. These descriptions served as the building blocks for further phases of the analytical process for the project. The explanations adequately covered the range of privacy concerns.

Then leveraging the capabilities of ChatGPT, an AI language model, it extracted the article number and category from GPT by giving the descriptions. It used ChatGPT's capacity to comprehend context, recognize significant components, and produce relevant replies to pinpoint the article numbers and categories connected to each privacy need. The procedure was greatly accelerated by this advanced automation. The below figure 3.1, shows how the information is extracted from GPT.



Which articles in the GDPR mention obtaining opt-in consent for the processing of personal data for specific purposes?
1. What is the article number?
2. The articles describe which of the categories: data protection, User participation, notice, data processing, breach, complaint/request, and security

The article in the GDPR that mentions obtaining opt-in consent for the processing of personal data for specific purposes is Article 6(1)(a).

1. Article Number: 6(1)(a)
2. Category: User Participation

Figure 3.1 Extracting the article number and category from GPT.

The privacy requirement description was carefully divided into seven categories, each representing a different aim or component of data protection. User participation, Data protection, notice, data processing, breach, complaint/request, and security are the seven categories. ‘Data protection’ was not identified as a distinct category from GPT, so there are a total of six categories. This framework for categorizing privacy needs gave an organized overview of them, making it easier to comprehend the goals and parameters of each requirement. The GDPR articles were then taken from the official GDPR website (Intersoft Consulting, 2018). The final dataset has 70 rows and 5 variables, including Requirement number, Description, Article No, Article Text, and Category is given below.

Column name	Description	Type
Requirement No	A unique identifier or number is assigned to each privacy requirement.	Alphanumerical
Description	Description of the privacy requirement.	Text
Article No from Chat GPT	The article number in the GDPR framework corresponds to the specific requirement.	Alphanumerical
Article Text	The text from the GDPR framework corresponds to the specific requirement.	Text
Category	The category of the privacy requirement	Category (Data protection, User participation, notice, data processing, breach, complaint/request, and security.

Table 3.1 Metadata of the dataset

3.2. Data Preprocessing

Data preparation is an essential stage in the pipeline for data analysis and machine learning. To prepare raw data for analysis or training machine learning models, it must be cleaned, transformed, and organized. The quality and efficiency of your analysis or model performance can be considerably impacted by proper data preparation.

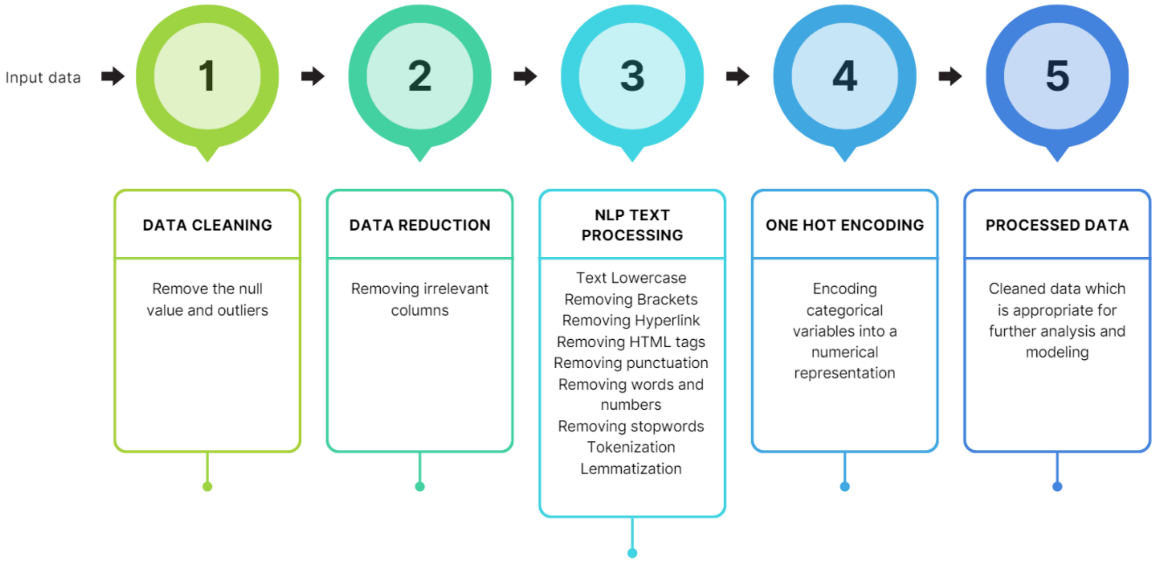
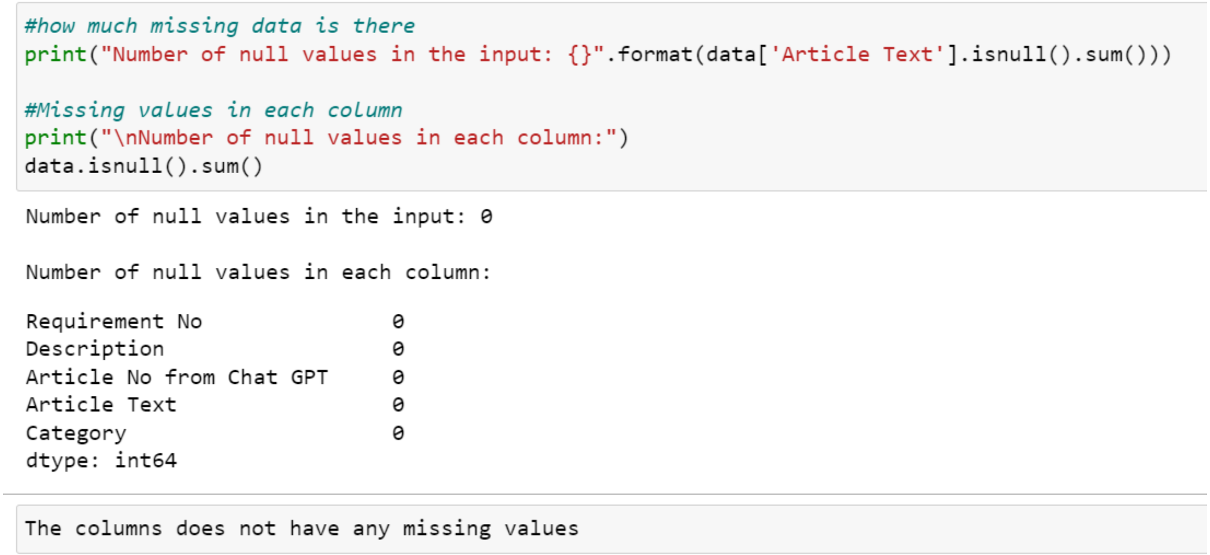


Figure 3.2 Data Preprocessing

3.2.1. Data Cleaning

The dataset's quality and usefulness were carefully considered throughout the construction process. To achieve this, a thorough data collection process was used, carefully collecting descriptions of the privacy requirements from reliable sources. Due to the strict data curation procedure and the fact that it had total control over dataset production, this strategy produced a dataset free of null values and outliers. The below figure 3.3 shows the code snippet of the missing values check.



```
#how much missing data is there
print("Number of null values in the input: {}".format(data['Article Text'].isnull().sum()))

#Missing values in each column
print("\nNumber of null values in each column:")
data.isnull().sum()
```

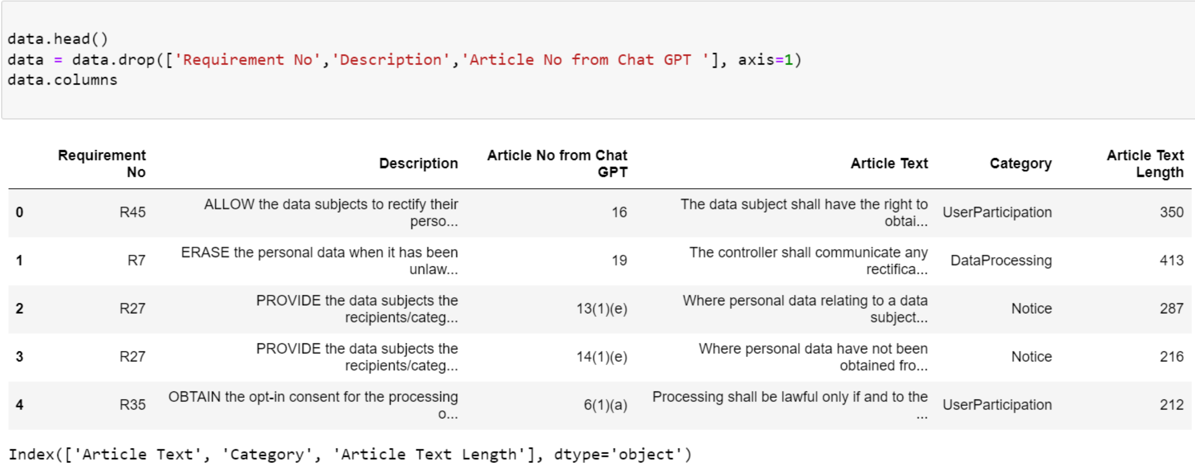
```
Number of null values in the input: 0

Number of null values in each column:

Requirement No             0
Description                0
Article No from Chat GPT   0
Article Text               0
Category                   0
dtype: int64
```

```
The columns does not have any missing values
```

Figure 3.3 Data Cleaning Code Snippet

3.2.2. Data Reduction

The dataset underwent a focused optimization process where carefully eliminated columns that were analyzed unnecessarily for the goals of the Study or held little significance. It simplified the dataset by purposefully deleting these irrelevant columns, concentrating solely on the attributes that were essential for the analysis and modeling objectives. Below figure 3.4 shows how the irrelevant columns ‘Requirement No’,’ Description’, and ’Article No from Chat GPT’ are removed from the dataset.



```
data.head()
data = data.drop(['Requirement No','Description','Article No from Chat GPT '], axis=1)
data.columns
```

	Requirement No	Description	Article No from Chat GPT	Article Text	Category	Article Text Length
0	R45	ALLOW the data subjects to rectify their perso...	16	The data subject shall have the right to obtai...	UserParticipation	350
1	R7	ERASE the personal data when it has been unlaw...	19	The controller shall communicate any rectifica...	DataProcessing	413
2	R27	PROVIDE the data subjects the recipients/categ...	13(1)(e)	Where personal data relating to a data subject...	Notice	287
3	R27	PROVIDE the data subjects the recipients/categ...	14(1)(e)	Where personal data have not been obtained fro...	Notice	216
4	R35	OBTAIN the opt-in consent for the processing o...	6(1)(a)	Processing shall be lawful only if and to the ...	UserParticipation	212

```
Index(['Article Text', 'Category', 'Article Text Length'], dtype='object')
```

Figure 3.4 Data Reduction Code Snippet.

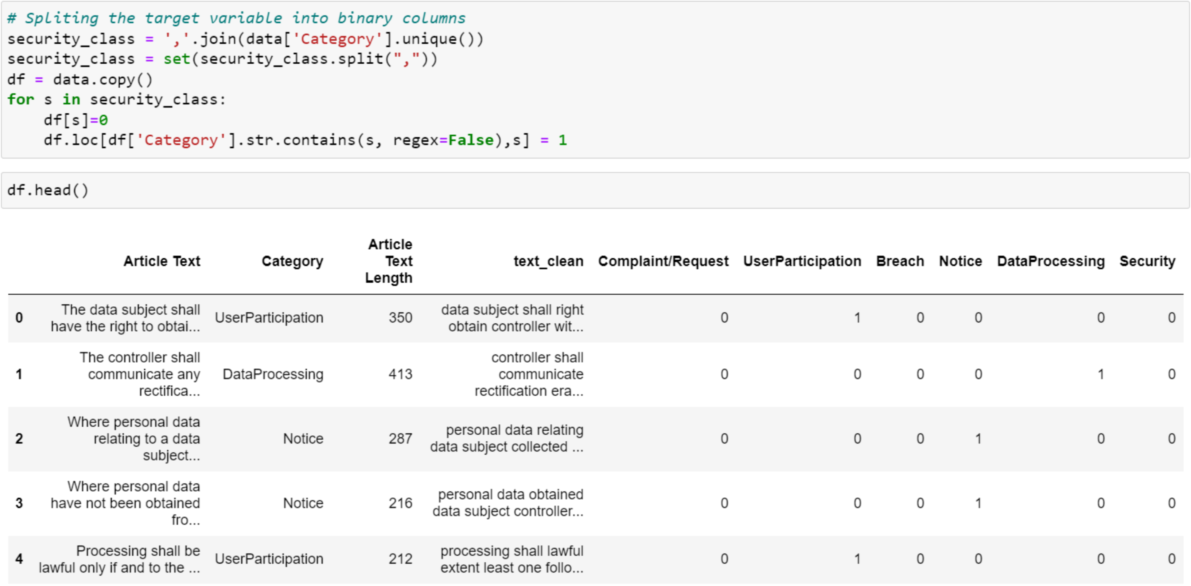
3.2.3. One hot encoding

The fundamental goal of this technique is to make it possible to use categorical data in analytical or predictive models. Categorical variables, like the 'Category' column in this instance, indicate discrete designations that are not necessarily numerical. However, a lot of machine learning algorithms operate using numerical inputs. One-hot encoding eliminates this discrepancy by establishing a collection of binary columns, where each column represents a different category. Each binary column's value denotes whether or not that category is present in a given record. Figure 3.5 shows how one hot encoding is done in the Study.

Without assuming any ordinal relationship or magnitude between the categories, the algorithm may recognize the lack or existence of particular categories such as "Security," "Notice," "DataParticipation," and others.

It enables computers to discover the patterns, correlations, and linkages within the data that are inherent in categorical information by one-hot encoding categorical variables into a numerical representation.

The dataset's integrity and effectiveness were strengthened by the lack of null values, outliers, and the deliberate removal of non-contributory columns. The following stages of the project were made possible by this strong foundation, which also guaranteed that the performance of the model and the analytical results that were obtained were based on accurate and insightful data representation.



```
# Spliting the target variable into binary columns
security_class = ','.join(data['Category'].unique())
security_class = set(security_class.split(","))
df = data.copy()
for s in security_class:
    df[s]=0
    df.loc[df['Category'].str.contains(s, regex=False),s] = 1
```

```
df.head()
```

	Article Text	Category	Article Text Length	text_clean	Complaint/Request	UserParticipation	Breach	Notice	DataProcessing	Security
0	The data subject shall have the right to obtai...	UserParticipation	350	data subject shall right obtain controller wit...	0	1	0	0	0	0
1	The controller shall communicate any rectifica...	DataProcessing	413	controller shall communicate rectification era...	0	0	0	0	1	0
2	Where personal data relating to a data subject...	Notice	287	personal data relating data subject collected ...	0	0	0	1	0	0
3	Where personal data have not been obtained fro...	Notice	216	personal data obtained data subject controller...	0	0	0	1	0	0
4	Processing shall be lawful only if and to the ...	UserParticipation	212	processing shall lawful extent least one follo...	0	1	0	0	0	0

Figure 3.5 One hot encoding Code Snippet.

3.2.4. Natural Language Processing Techniques

A thorough pipeline for preparing data for Natural Language Processing (NLP) is being implemented. The pipeline is made up of several text transformation procedures designed to clean up the unprocessed text data and make it more appropriate for further analysis and modeling. The preliminary text processing is described in detail as follows:

Text Lowercasing:

The first step entails making all of the text lowercase. Recognizing the capital and lowercase variants of the same word as interchangeable assures consistency and uniformity across the text.

Removing Text within Square Brackets:

All text that is encased in square brackets, such as citations, references, or other annotations, is eliminated. By doing this, unnecessary data that might not be useful to the analysis are removed.

Removal of Hyperlinks (URLs):

Any links that were previously present in the text are eliminated. This step is crucial since URLs frequently include noise and don't offer useful information for research.

Removing HTML Tags:

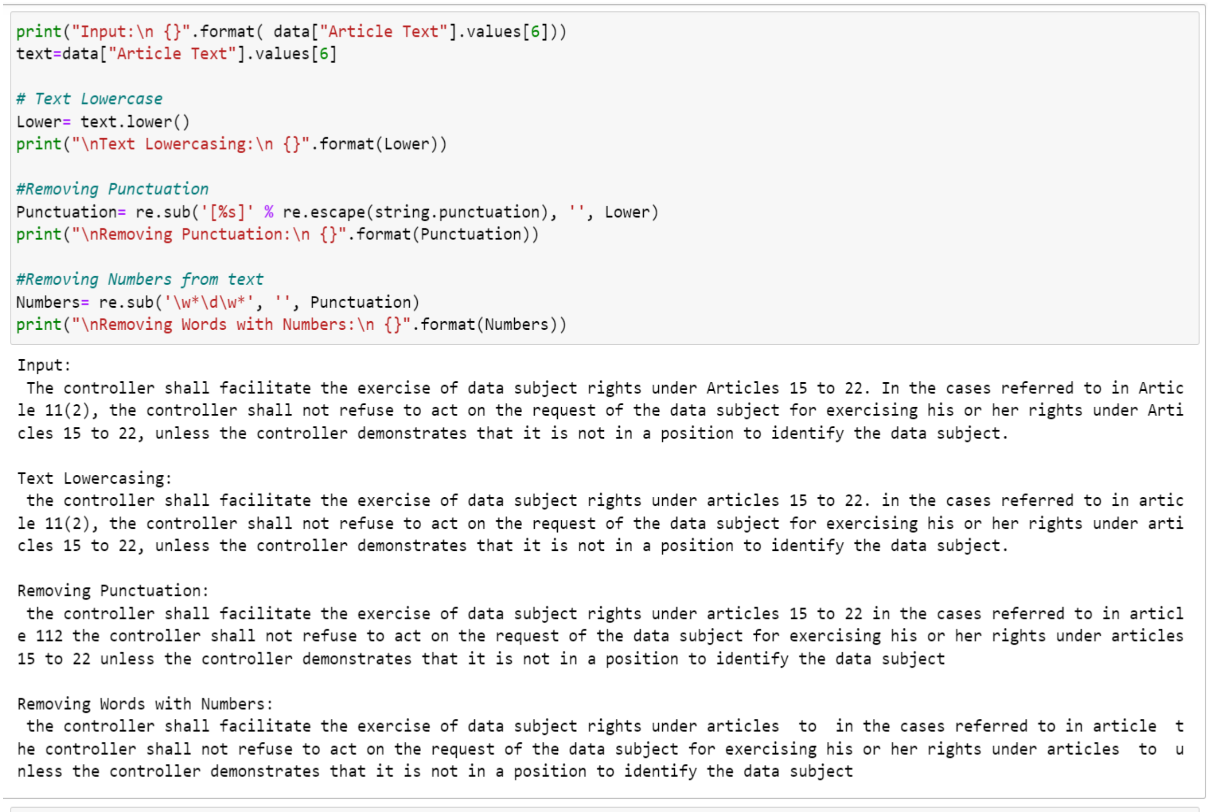
HTML tags, which are indicated in the text by angle brackets (>), are eliminated. Although these tags are used in online content to describe the formatting, most NLP analyses don't make sense with them.

Removing Punctuation:

Commas, periods, and exclamation points are among the dropped punctuation marks. Punctuation is frequently overlooked in literary analyses since it generally does not provide semantic significance.

Removing Words with Numbers:

Words containing numbers are eliminated from the text. This removes words that may be mentioned in references, numbers, or other non-textual information. Figure 3.6 shows the code snippet of NLP text processing for text lower casing, punctuation, and number removal.



```
print("Input:\n {}".format( data["Article Text"].values[6]))
text=data["Article Text"].values[6]

# Text Lowercase
Lower= text.lower()
print("\nText Lowercasing:\n {}".format(Lower))

#Removing Punctuation
Punctuation= re.sub('[%s]' % re.escape(string.punctuation), '', Lower)
print("\nRemoving Punctuation:\n {}".format(Punctuation))

#Removing Numbers from text
Numbers= re.sub('\w*\d\w*', '', Punctuation)
print("\nRemoving Words with Numbers:\n {}".format(Numbers))
```

```
Input:
 The controller shall facilitate the exercise of data subject rights under Articles 15 to 22. In the cases referred to in Artic
le 11(2), the controller shall not refuse to act on the request of the data subject for exercising his or her rights under Arti
cles 15 to 22, unless the controller demonstrates that it is not in a position to identify the data subject.

Text Lowercasing:
 the controller shall facilitate the exercise of data subject rights under articles 15 to 22. in the cases referred to in artic
le 11(2), the controller shall not refuse to act on the request of the data subject for exercising his or her rights under arti
cles 15 to 22, unless the controller demonstrates that it is not in a position to identify the data subject.

Removing Punctuation:
 the controller shall facilitate the exercise of data subject rights under articles 15 to 22 in the cases referred to in articl
e 112 the controller shall not refuse to act on the request of the data subject for exercising his or her rights under articles
15 to 22 unless the controller demonstrates that it is not in a position to identify the data subject

Removing Words with Numbers:
 the controller shall facilitate the exercise of data subject rights under articles  to  in the cases referred to in article  t
he controller shall not refuse to act on the request of the data subject for exercising his or her rights under articles  to  u
nless the controller demonstrates that it is not in a position to identify the data subject
```

Figure 3.6 Code snippet of lowercase, punctuation, and number removal.

Tokenization:

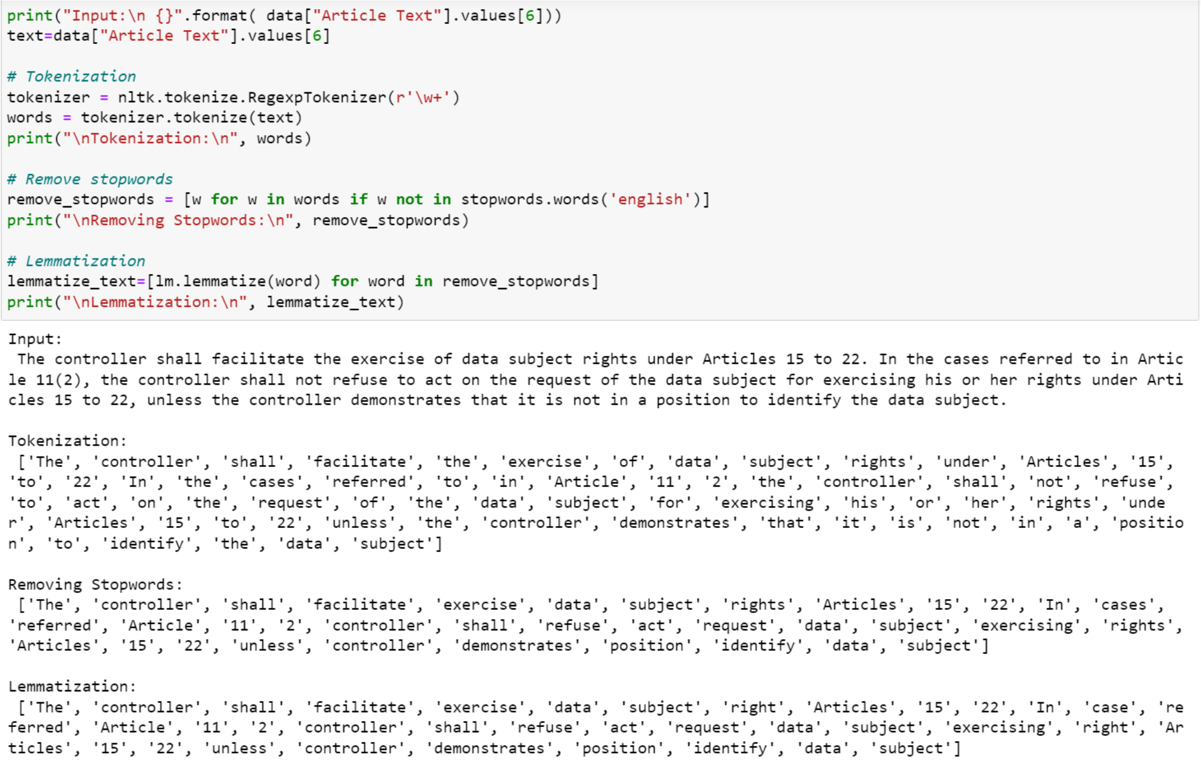
Tokenization is the process of breaking down a text into separate components, often words or subwords, known as tokens. Tokens are the foundation of text analysis and are required for tasks like text categorization, sentiment analysis, machine translation, and others.

Removing Stopwords:

Stopwords are common words within texts such as "and," "the," "in," and "is" that appear often but generally provide little significant information for analysis. Stopword removal reduces noise and complexity in text data, making it more suited for applications such as text classification, sentiment analysis, and topic modelling.

Lemmatization:

Words are condensed to their simplest or dictionary form by the process of lemmatization. This procedure makes sure that the same word is treated the same way regardless of its many grammatical variants. This standardization facilitates analysis and supports the identification of semantic parallels. Figure 3.7 shows the code snippet of NLP text processing for tokenization, stop word removal, and lemmatization.



```
print("Input:\n {}".format( data["Article Text"].values[6]))
text=data["Article Text"].values[6]

# Tokenization
tokenizer = nltk.tokenize.RegexpTokenizer(r'\w+')
words = tokenizer.tokenize(text)
print("\nTokenization:\n", words)

# Remove stopwords
remove_stopwords = [w for w in words if w not in stopwords.words('english')]
print("\nRemoving Stopwords:\n", remove_stopwords)

# Lemmatization
lemmatize_text=[lm.lemmatize(word) for word in remove_stopwords]
print("\nLemmatization:\n", lemmatize_text)
```

```
Input:
 The controller shall facilitate the exercise of data subject rights under Articles 15 to 22. In the cases referred to in Artic
le 11(2), the controller shall not refuse to act on the request of the data subject for exercising his or her rights under Arti
cles 15 to 22, unless the controller demonstrates that it is not in a position to identify the data subject.

Tokenization:
 ['The', 'controller', 'shall', 'facilitate', 'the', 'exercise', 'of', 'data', 'subject', 'rights', 'under', 'Articles', '15',
'to', '22', 'In', 'the', 'cases', 'referred', 'to', 'in', 'Article', '11', '2', 'the', 'controller', 'shall', 'not', 'refuse',
'to', 'act', 'on', 'the', 'request', 'of', 'the', 'data', 'subject', 'for', 'exercising', 'his', 'or', 'her', 'rights', 'unde
r', 'Articles', '15', 'to', '22', 'unless', 'the', 'controller', 'demonstrates', 'that', 'it', 'is', 'not', 'in', 'a', 'positio
n', 'to', 'identify', 'the', 'data', 'subject']

Removing Stopwords:
 ['The', 'controller', 'shall', 'facilitate', 'exercise', 'data', 'subject', 'rights', 'Articles', '15', '22', 'In', 'cases',
'referred', 'Article', '11', '2', 'controller', 'shall', 'refuse', 'act', 'request', 'data', 'subject', 'exercising', 'rights',
'Articles', '15', '22', 'unless', 'controller', 'demonstrates', 'position', 'identify', 'data', 'subject']

Lemmatization:
 ['The', 'controller', 'shall', 'facilitate', 'exercise', 'data', 'subject', 'right', 'Articles', '15', '22', 'In', 'case', 're
ferred', 'Article', '11', '2', 'controller', 'shall', 'refuse', 'act', 'request', 'data', 'subject', 'exercising', 'right', 'Ar
ticles', '15', '22', 'unless', 'controller', 'demonstrates', 'position', 'identify', 'data', 'subject']
```

Figure 3.7 Code snippet of Tokenization, stop word removal, and lemmatization.

These preprocessing steps work together to provide a more streamlined, organized, and tokenized representation of the original text data. Various NLP tasks, including sentiment analysis, text categorization, topic modeling, and more, may be performed on this processed text. These preprocessing processes are crucial for efficient NLP analysis and modeling because they convert the raw text into a matrix structure that computers can use. Figure 3.8 shows the flowchart of the methodology.

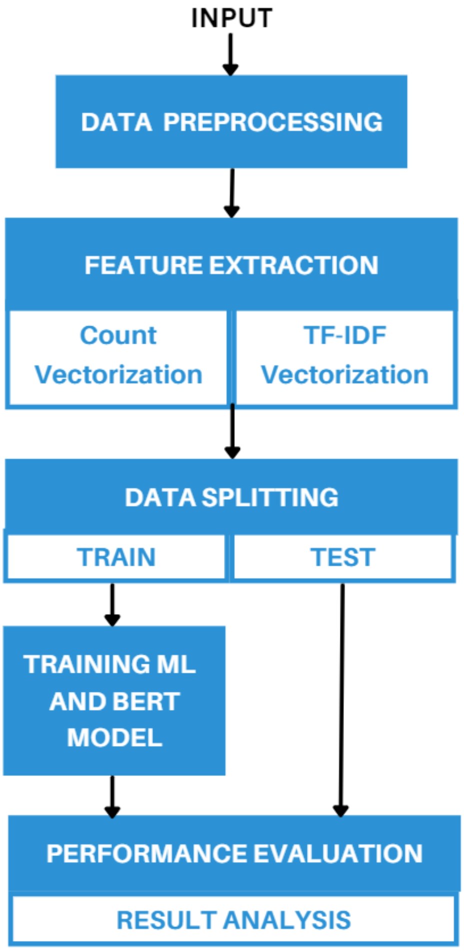


Figure 3.8 Methodology Flowchart.

3.3. Feature Extraction

The feature extraction stage is the most important one since it determines how the work will turn out overall. A wise feature choice leads to an accurate prognosis. Therefore, selecting criteria that enhance categorization properly is essential. After the pre-processing phase was complete, the Count Vectorizer and TF-IDF features were used to assess the produced data. Feature extraction boosts algorithmic efficiency, makes comparisons easier, and increases the accuracy of numerous NLP tasks including classification, clustering, and sentiment analysis. It does this by compressing information, identifying patterns, and removing noise.

3.3.1. TF-IDF Vectorizer

Words are given weights through TF-IDF (Term Frequency-Inverse text Frequency) Vectorizer depending on their rarity within a corpus and their frequency inside a text. The representation of words' relevance in a document is improved by TF-IDF, which also helps with different NLP tasks. Term Frequency (TF) and Inverse Document Frequency (IDF) are both taken into account while using TF-IDF.

Term Frequency (TF): It measures how frequently a term or word appears in a text relative to all the other terms in the document.

$$TF = \frac{\text{Number of times the word appears in the document}}{\text{Total Number of terms in the document}} \tag{3.1}$$

Inverse Document Frequency: The IDF of a term indicates the percentage of the corpus that includes the term.

Words that are only found in a limited number of papers, such as technical jargon terms, are given greater relevance ratings than words that are used in all publications, such as a, the, and.

$$IDF = \log\left\{\frac{\text{Number of the document in the corpus}}{\text{Number of documents in the corpus contain the term}}\right\} \qquad (3.2)$$

The sum of a word's TF and IDF values determines the word's TF-IDF score in a document. Unique words are crucial for characterizing a text and have high TF-IDF values, whereas common words have lower scores.

$$\text{TF-IDF} = \text{TF} * \text{IDF} \qquad (3.3)$$

The goal is to identify terms that are significant throughout all texts, not only in each one individually. While IDF concentrates on the relevance of a word across documents, TF concentrates on the importance of a word inside a document (Chirasmayee, 2022).

In the implementation, the TF-IDF computation has a limit of 1000 unique words (features), which means that only the top 1000 most significant words will be kept as features. The value (1,3) is chosen as the n-gram range. Unigrams, bigrams, and trigrams (sequences of three consecutive words) are all taken into account by the vectorizer.

One significant flaw is that it ignores the links between words in context and fails to understand the semantic meaning of words. This can be a challenge for activities that call for comprehension of word meanings in certain settings. Word order and sentence structure are also disregarded by TF-IDF, despite the fact that these factors are essential for tasks like sentiment analysis and language synthesis. Another issue is the effect of vocabulary size; a big vocabulary might result in computations that use a lot of resources and could impair algorithm performance.

3.3.2. Count Vectorizer

Count Vectorizer is based on the concept of illustrating texts as collections of words, with an emphasis on word frequency rather than word order or semantic significance. The natural richness of human language is transformed into a structured and quantitative representation thanks in large part to this method. In this study, the scikit-learn count vectorizer was employed (Varoquaux et al., 2015).

Tokenizing text data into individual words or terms is the initial step in the Count Vectorizer process. The entire corpus is then utilized to generate a vocabulary of unique terms. Then, each document is represented as a vector, with each dimension representing a lexical term and the value representing the number of times that word appears in the text as a whole. As a result, a term-document matrix is constructed, with columns representing terms and rows representing documents.

The vocabulary size is regulated and word combinations are considered by initializing the vectorizer with settings like max_features and ngram_range. The resultant matrix's rows each correspond to a document, and its columns each stand in for a term from the dictionary. The values in the matrix represent the frequency of occurrence of each term in the corresponding documents.

The simplicity of Count Vectorizer is one of its significant benefits. It is simple to use and understand, making it the best option in situations when interpretability is crucial. Additionally, it preserves the original structure of texts, which is helpful for activities like sentiment analysis, content categorization, and even searching for certain words.

3.4. Training and Testing data split

During the data-splitting step, a reliable method called K-fold cross-validation was used to divide the dataset into five different subsets, or "folds." Iteratively identifying one fold as the validation set and the remaining folds as the training set is how this technique works. The model is developed using training data, and it is subsequently assessed using validation fold. Each fold acts as the validation set once throughout the five iterations of this operation. A method like this reduces the possibility of overfitting and maximizes the use of the data, improving the model's evaluation.

A deeper knowledge of the model's performance is attained by repeatedly training and assessing it with various data combinations. After averaging the accuracy ratings acquired for each fold, a precise assessment of the model's capacity to generalize across various datasets is produced. This approach is very helpful for NLP applications since it enables the evaluation of the model's performance on different text samples. K-fold cross-validation ultimately guarantees a thorough review process and helps to produce a more accurate evaluation of the model's overall performance.

3.5. Machine Learning Model

This article uses machine learning models to automatically categorize text data into predetermined categories. The basic objective is to construct predictive models that can generalize patterns from the training data and apply those patterns to new, unforeseen data to generate precise predictions about the categories the text belongs.

To categorize text data into multiple categories (such as "Notice," "Breach," "DataProcessing," etc.), machine learning models like Logistic Regression, Random Forest, Support Vector Machines (SVM), and Decision Trees are used. The associations between the characteristics retrieved from the text and the target categories are automatically learned using these models.

3.5.1. Logistic Regression

A popular classification approach in machine learning is logistic regression. In binary classification problems, where the objective is to predict one of two potential outcomes, it is especially well suited. Estimating the likelihood that a given input belongs to a particular class is the basic idea. Logistic regression uses the logistic function to limit predictions to the range of 0 to 1, thereby converting unrefined guesses into probabilities (Zhou et al., 2019). In contrast to linear regression, which predicts continuous values.

Logistic regression determines a decision boundary that divides the two classes in feature space when dealing with binary classification. Each feature's weight is learned by the algorithm, which also determines how it affects the likelihood that will be predicted. Utilizing optimization techniques, it modifies these weights during training to reduce the discrepancy between projected probability and actual class labels in the training set.

3.5.2. Support Vector Machines (SVM)

It operates by identifying the ideal hyperplane in a high-dimensional space that optimally divides several types of data. The objective is to increase the margin between classes since it aids in better generalization to brand-new, untested material.

By transforming the data into a higher-dimensional space where classes are easier to distinguish, kernel functions used in SVM enable it to tackle both linear and non-linear classification issues. The most popular kernel types are radial basis function (RBF), polynomial, and linear.

Support Vector Machines are employed in the approach to multi-class categorize privacy requirements into several categories. K-fold cross-validation is used to train and assess the model's performance like logistic regression.

SVM looks for the optimum decision boundary to discriminate between several legal requirement groups. The cross-validation accuracy ratings show how effectively SVM can forecast the applicability of requirements in various legal scenarios.

3.5.3. Decision tree

Recursively dividing the dataset into subsets according to the values of various characteristics produces a tree-like structure with decision nodes and leaf nodes, which is how it operates. Each leaf node represents a class label or a regression value, whereas each internal node represents a judgment call based on an attribute.

By choosing the ideal qualities to partition the dataset at each node, the algorithm discovers patterns in the data (Veluri et al., 2022). The most effective separation of classes is achieved by analyzing characteristics that lead to the greatest information gain or impurity reduction.

Because of their clarity, interpretability, and capacity to identify non-linear correlations in data, decision trees are desirable. They may nonetheless produce deep trees that memorize the noise in the training set, which can lead to overfitting. Techniques like pruning and ensemble approaches, like Random Forest, can be utilized to lessen this.

In conclusion, Decision Trees are essential parts of more complicated algorithms like Random Forests and Gradient Boosting and provide a transparent approach to making judgments based on input information.

3.5.4. Random Forest

Several decision trees are combined using the ensemble machine learning method Random Forest to produce a reliable and accurate model. It deals with the drawbacks of single decision trees, which might be overfitting-prone and have low prediction accuracy. Multiple decision trees are combined to increase resilience and generalization while preserving high accuracy in Random Forest.

The approach works by building a forest of decision trees, each trained on a randomly selected fraction of the dataset (referred to as bootstrapped samples) and employing a randomly selected subset of characteristics. As a result of this unpredictability, the trees become more diverse and are less likely to overfit certain data patterns. The method integrates all of the trees' outputs to provide a final result during prediction.

Random Forest is a potent tool in several fields, including legal document analysis, because of its capacity to identify complicated connections, manage massive datasets, and avoid overfitting (Suraj Kumar Parhi and Sanjaya Kumar Patro, 2023). It offers increased precision and stability by drawing on the combined wisdom of many decision trees. However, given that Random Forests are ensembles, making them more difficult to read than single decision trees, it is crucial to set hyperparameters and take interpretability into account.

3.5.5. Implementation of the models

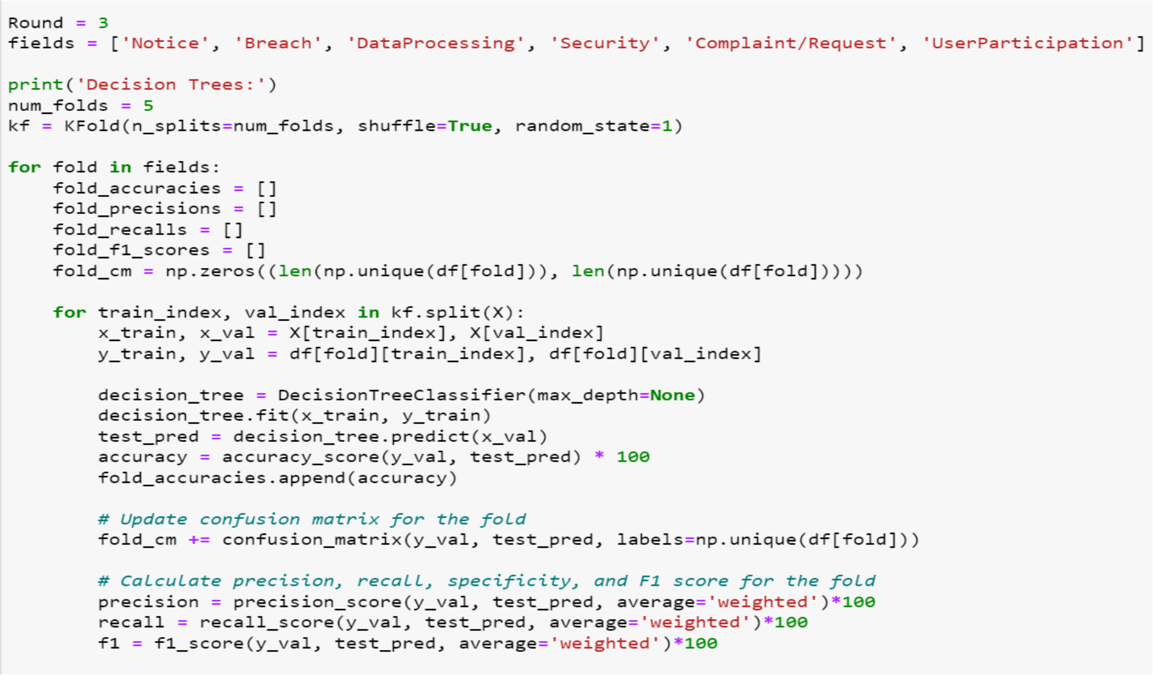
The implementation starts with defining key parameters and importing necessary libraries such as scikit-learn for machine learning, NumPy for numerical calculations, and matplotlib.pyplot for data visualization.

The implementation is centered on an iterative method that focuses on each privacy requirement area. The dataset is methodically separated into training and validation sets for each cross-validation fold during this procedure, ensuring that the model is thoroughly evaluated on various data subsets.

Using the training data, the machine learning model is trained during each fold. This classifier is specially designed to distinguish between distinct classes or categories within the selected field. When the model has been sufficiently trained, it is used to create predictions on the validation data. After training, we utilize the model to generate predictions on the validation data, and this procedure is done for all folds. We compute many essential classification metrics when we make predictions to evaluate the model's performance.

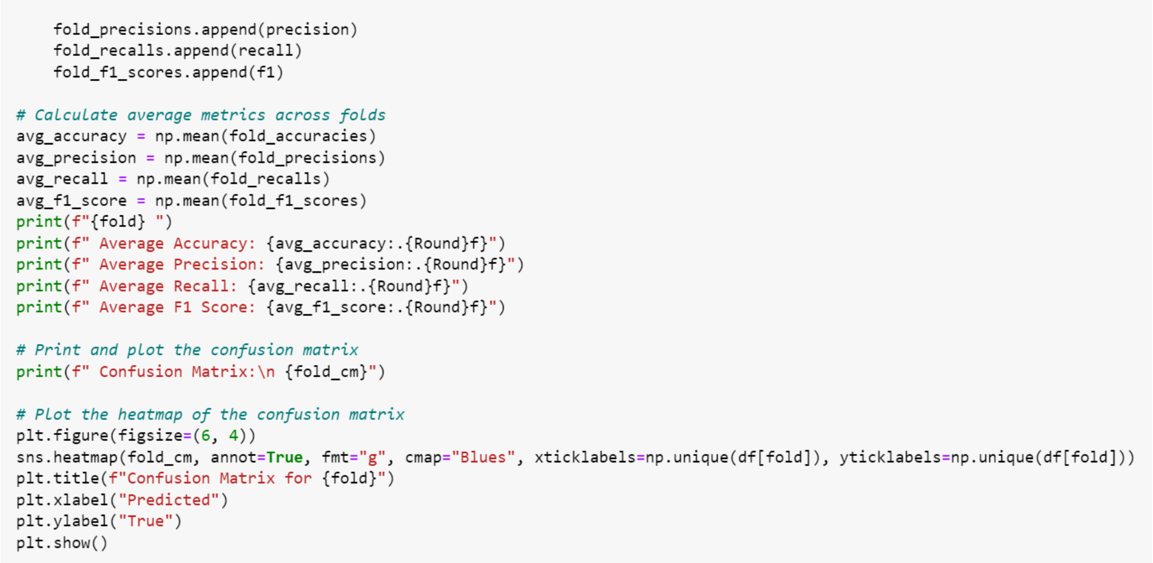
Finally, after processing all folds and categories, we compute and report the average metrics for each category. We also provide heatmap visualizations of the confusion matrices to provide a visual depiction of the model's performance in differentiating between distinct classes within each category.

This systematic methodology allows a thorough examination of the machine learning model's efficacy in categorizing GDPR into privacy requirements for diverse domains of interest, offering useful insights for model review and modification. Figure 3.9 and 3.10, shows the code snippet of decision tree model with count vectorizer.



```
Round = 3
fields = ['Notice', 'Breach', 'DataProcessing', 'Security', 'Complaint/Request', 'UserParticipation']

print('Decision Trees:')
num_folds = 5
kf = KFold(n_splits=num_folds, shuffle=True, random_state=1)

for fold in fields:
    fold_accuracies = []
    fold_precisions = []
    fold_recalls = []
    fold_f1_scores = []
    fold_cm = np.zeros((len(np.unique(df[fold])), len(np.unique(df[fold]))))

    for train_index, val_index in kf.split(X):
        x_train, x_val = X[train_index], X[val_index]
        y_train, y_val = df[fold][train_index], df[fold][val_index]

        decision_tree = DecisionTreeClassifier(max_depth=None)
        decision_tree.fit(x_train, y_train)
        test_pred = decision_tree.predict(x_val)
        accuracy = accuracy_score(y_val, test_pred) * 100
        fold_accuracies.append(accuracy)

        # Update confusion matrix for the fold
        fold_cm += confusion_matrix(y_val, test_pred, labels=np.unique(df[fold]))

        # Calculate precision, recall, specificity, and F1 score for the fold
        precision = precision_score(y_val, test_pred, average='weighted')*100
        recall = recall_score(y_val, test_pred, average='weighted')*100
        f1 = f1_score(y_val, test_pred, average='weighted')*100
```

Figure 3.9 Decision tree with count vectorizer Code snippet -1



```
    fold_precisions.append(precision)
    fold_recalls.append(recall)
    fold_f1_scores.append(f1)

# Calculate average metrics across folds
avg_accuracy = np.mean(fold_accuracies)
avg_precision = np.mean(fold_precisions)
avg_recall = np.mean(fold_recalls)
avg_f1_score = np.mean(fold_f1_scores)
print(f"{fold} ")
print(f" Average Accuracy: {avg_accuracy:.{Round}f}")
print(f" Average Precision: {avg_precision:.{Round}f}")
print(f" Average Recall: {avg_recall:.{Round}f}")
print(f" Average F1 Score: {avg_f1_score:.{Round}f}")

# Print and plot the confusion matrix
print(f" Confusion Matrix:\n {fold_cm}")

# Plot the heatmap of the confusion matrix
plt.figure(figsize=(6, 4))
sns.heatmap(fold_cm, annot=True, fmt="g", cmap="Blues", xticklabels=np.unique(df[fold]), yticklabels=np.unique(df[fold]))
plt.title(f"Confusion Matrix for {fold}")
plt.xlabel("Predicted")
plt.ylabel("True")
plt.show()
```

Figure 3.10 Decision tree with count vectorizer Code snippet -2

3.7. BERT Model

A potent pre-trained language model called BERT (Bidirectional Encoder Representations from Transformers) was created for a variety of NLP applications, including text categorization. Based on the Transformer architecture, it learned contextualized word embeddings by training on a vast quantity of text data. BERT captures the meaning of a word by taking into account both its left and right context in a phrase, in contrast to conventional models that take words into account individually.

BERT is very good at comprehending context and semantics in text categorization tasks because of its bidirectional attention mechanism, which enables it to grasp complex relationships between words (Zheng et al., 2022). Due to its extensive contextual knowledge, it can perform jobs like sentiment analysis, intent identification, and more while producing cutting-edge outcomes.

Implementation of the BERT model:

First, the model imports the GDPR dataset, with each rule connected with privacy required categories. It prepares the data for categorization by turning these categories into binary labels. The algorithm then divides the data into training and testing sets for each category to guarantee impartial performance evaluation. Tokenization is an important step in which text input is converted into numerical representations that BERT can handle. DataLoaders are then constructed to handle the training and testing data efficiently. The training process is divided into epochs, with each epoch improving the model's comprehension of the data. The model is assessed on a validation set after each epoch, and the best-performing model state is kept.

After training, the stored model with the best validation accuracy is loaded for testing on a separate test set. This final review produces crucial measures including as accuracy, precision, recall, and F1 score, which provide insight into the model's categorization skills.

Another important result is the confusion matrix, which graphically represents how effectively the model identifies distinct categories under the given privacy need category.

Comparison of traditional machine learning techniques and BERT:

BERT excels in contextual awareness by taking into account the surrounding words, allowing it to capture nuanced language nuances. Its immense pre-trained knowledge, acquired from massive amounts of text data, enables it to generalize effectively across a wide range of activities and domains, making it extremely adaptable. Transfer learning with BERT is a valuable tool since it allows for fine-tuning specific tasks, resulting in improved performance with less data. BERT is also capable of dealing with language variances and tolerating diverse writing styles and dialects, which is useful in multilingual settings. It has revolutionized NLP jobs and continues to be a basic model for many natural languages processing applications, including text categorization. However, it has limitations such as high computing needs, a large model size, restricted interpretability, data-intensive fine-tuning requirements, and lengthier training durations, all of which must be considered when deciding between BERT and typical machine learning models.

Chapter 4: Evaluation

In this section, the experimental setup, Hardware/ software requirements, and evaluation parameters used to assess the performance of the classification models are detailed.

4.1. Experimental Setup

The dataset is divided using k-fold cross-validation into 5 folds. The data is divided into k equal pieces. One part is left out for testing while the model is tested on k-1 pieces (Yadav, and Shukla, 2016). This indicates that the dataset is split into 5 roughly equal-sized sections. It repeats this process for each fold, utilizing 4 of them for training and 1 for validation. This 80-20 split (80% training, 20% validation) is the default behavior of k-fold cross-validation in scikit-learns.

4.2. Hardware Requirements

An Intel Core i7 CPU, 16GB of RAM, and an NVIDIA GeForce GTX 1080 GPU were used in the experimental configuration. The training and evaluation phases were accelerated using the GPU, increasing the experiment's computational efficiency.

4.3. Software Requirements

The investigation was conducted using the computer language Python. Several frameworks and tools were utilized, including Scikit-learn for machine learning techniques, NLTK for text preprocessing, and Matplotlib for data visualization. For NLP tasks, the Hugging Face Transformers library's BERT (Bidirectional Encoder Representations from Transformers) model was also used.

4.4. Evaluation Parameter

Classification measuring parameters like confusion matrix, Recall, Accuracy, Precision, and F-score are employed in this Study since many researchers use these metrics often for classification.

a. Confusion Matrix

In classification tasks, a confusion matrix is used to explain how well a classification model performs on a collection of data for which the real values are known. It allows you to see and comprehend how successfully a model categorizes occurrences into various classes. One dimension of a confusion matrix is indexed by the actual class of an item, while the other is indexed by the predicted class of the classifier (Deng et al., 2016).

The model's predictions for each class are shown visually by the confusion matrix and the corresponding heatmap. This aids in identifying potential error-prone areas for the model and classes that provide the most difficulty for proper classification. Figure 4.1 shows the confusion matrix of all the categories for Decision Tress with count Vectorizer.

Here's a breakdown of the four main components of a confusion matrix:

True Positives (TP): Instances correctly predicted as positive.

True Negatives (TN): Instances correctly predicted as negative.

False Positives (FP): Instances wrongly predicted as positive (Type I errors).

False Negatives (FN): Instances wrongly predicted as negative (Type II errors).

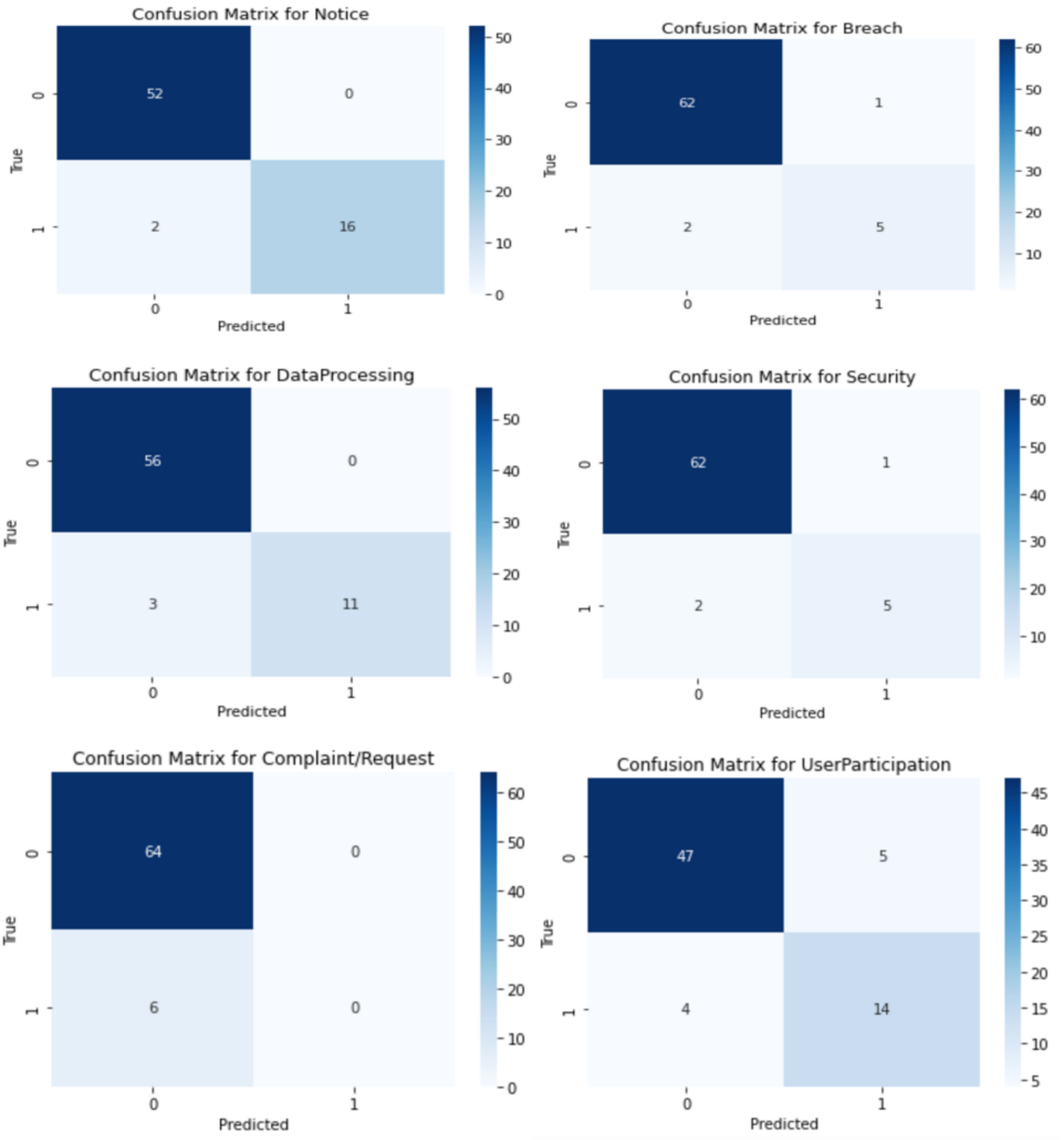


Figure 4.1 Confusion Matrix of Decision Tree with Count Vectorizer.

b. Average Accuracy

The average accuracy number acquired from each fold of cross-validation is known as average accuracy. The percentage of cases out of all instances that were properly predicted is known as accuracy. Figure 4.2 shows the accuracy for all the categories in the Decision Tree with the Count Vectorizer. It is calculated as below:

$$Average\ Accuracy = \frac{Sum\ of\ accuracies\ from\ all\ folds}{Number\ of\ folds} \tag{4.1}$$

$$Accuracy = \frac{\text{TruePositive + TrueNegative}}{\text{TruePositive+TrueNegative+FalsePositive+FalseNegative}} \tag{4.2}$$

Where:

Accuracies from all folds: The accuracy values were found during cross-validation on each fold.

Number of folds: The total number of folds utilized in cross-validation.

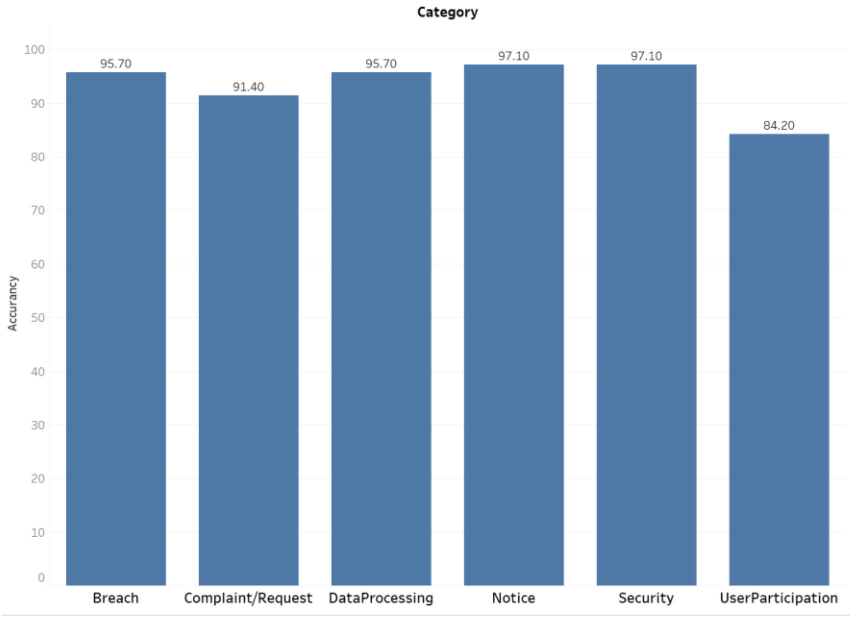


Figure 4.2 Accuracy - Decision Tree with the Count Vectorizer

c. Average Precision

The average of the precision values collected from each fold during cross-validation is known as average precision. The degree to which a model can correctly forecast positive outcomes is referred to as precision. Figure 4.3 shows the average Precision for all the categories in the Decision Tree with the Count Vectorizer. It is calculated as below:

$$Average\ Precision = \frac{Sum\ of\ precision\ from\ all\ folds}{Number\ of\ folds} \tag{4.3}$$

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{4.4}$$

Where:

Precisions from all Folds: The precision values discovered during cross-validation from each fold.

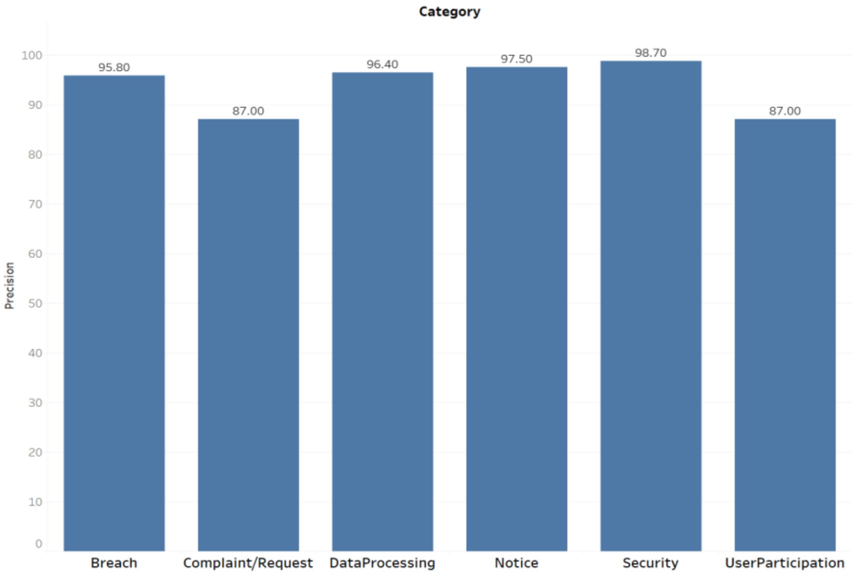


Figure 4.3 Precision - Decision Tree with the Count Vectorizer

d. Average Recall

Recall is a statistic that measures the proportion of accurate positive predictions among all possible positive predictions. The average of the recall values collected from each fold during cross-validation is known as average recall. It is calculated as below:

$$Average\ Recall = \frac{Sum\ of\ recall\ from\ all\ folds}{Number\ of\ folds} \tag{4.5}$$

$$Recall = \frac{TruePositives}{TruePositives+FalseNegatives} \tag{4.6}$$

Where:

Recalls from all Folds: The recall values were obtained during cross-validation from each fold. Figure 4.4 shows the average recall for all the categories in the Decision Tree with the Count Vectorizer.

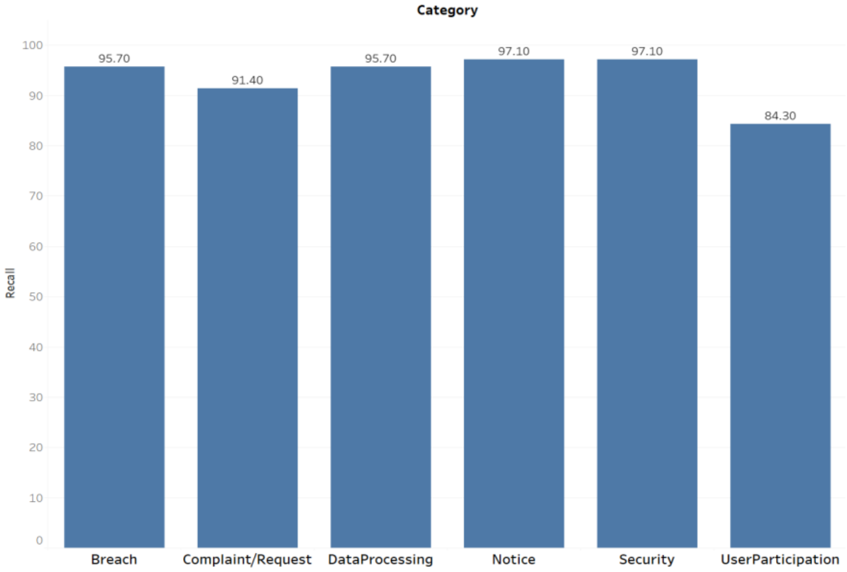


Figure 4.4 Recall - Decision Tree with the Count Vectorizer

e. Average F1-Score

A balanced statistic for model performance is provided by the average F1-score, which is the harmonic mean of accuracy and recall. In cross-validation, the average F1-score is the sum of the F1-score values derived from each fold. Figure 4.5 shows the average F1 Score for all the categories in the Decision Tree with the Count Vectorizer. It is calculated as below:

$$Average\ F1\ Score = \frac{Sum\ of\ F1\ Score\ from\ all\ folds}{Number\ of\ folds} \tag{4.7}$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision+Recall} \tag{4.8}$$

Where:

F1-Scores from all Folds: The F1-score values obtained from each fold during cross-validation.

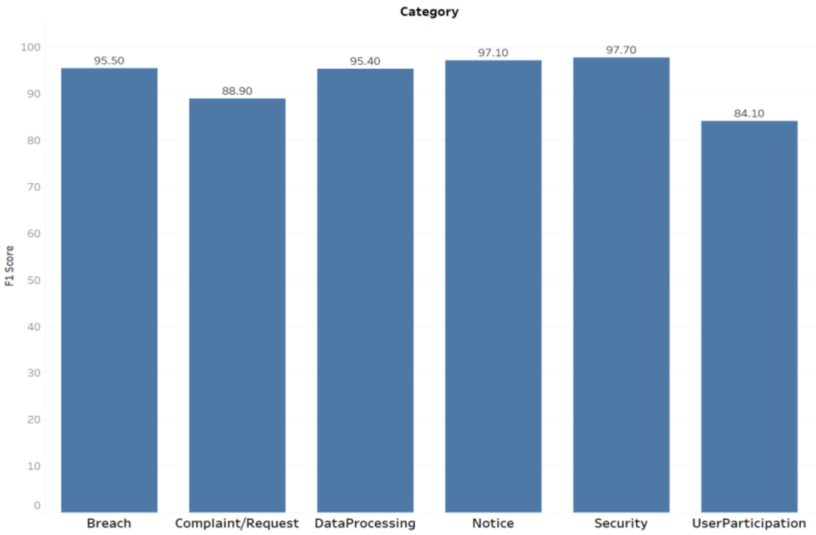


Figure 4.5 F1 Score - Decision Tree with the Count Vectorizer

Chapter 5: Results and Analysis

In this section, the outcomes of the analysis, comparing the performance of machine learning models using different vectorization techniques and the BERT model for categorizing GDPR into privacy requirements are presented.

5.1. Machine Learning Models with Count Vectorizer

The Decision Tree model consistently earns the greatest values across all categories, for accuracy, precision, recall, and F1 score (Accuracy: Notice - 97.1%, Breach - 95.7%, DweataProcessing - 95.7%, Security - 97.1%, Complaint/Request - 91.4%, and UserParticipation - 84.2%). This indicates that the Decision Tree model has been successful in accurately capturing the underlying patterns in the data and making predictions on this specific dataset using the available features (Count Vectorizer). Table 5.1 shows the comparisons of machine learning model performance for count vectorizer.

The "UserParticipation" category shows significantly lower scores across all measures, notably when employing SVM (F1 score - 67.2%) and Random Forest (F1 score - 69.7%). The text patterns in this category are harder to categorize and there aren't as many distinguishing characteristics that let the models separate instances properly. The general performance of all models (Logistic Regression, Random Forest, SVM, Decision Tree) is good for categories like "Notice," "Breach," "DataProcessing," "Security," and "Complaint/Request". Table 1 displays the performance metrics of various machine learning models to categorize GDPR rules into categories for privacy needs using Count Vectorizer. For each category of privacy requirements, measures such as Accuracy, Precision, Recall, and F1 Score are used.

Count Vectorizer					
Category	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Machine Learning Model**
Notice	92.8	94.4	92.9	92.7	**Logistic Regression**
Breach	92.8	89.5	92.9	90.3	
DataProcessing	94.2	94.8	94.3	94.1	
Security	95.7	94.6	95.7	94.8	
Complaint/Request	91.4	84.3	91.4	87.5	
UserParticipation	78.5	84.3	78.6	77.5	
Notice	92.8	94.4	92.9	92.7	**Random Forest**
Breach	90	81.5	90	85.4	
DataProcessing	94.2	94.8	94.3	94.1	
Security	95.7	94.6	95.7	94.8	
Complaint/Request	91.4	84.3	91.4	87.5	
UserParticipation	75.7	67.9	75.7	69.7	
Notice	92.8	94.4	92.9	92.7	**Support Vector Machine**
Breach	90	81.5	90	85.4	
DataProcessing	94.2	95	94.3	93.8	
Security	95.7	94.6	95.7	94.8	
Complaint/Request	91.4	84.3	91.4	87.5	
UserParticipation	74.2	63.9	74.3	67.2	
Notice	**97.1**	**97.5**	**97.1**	**97.1**	**Decision Tree**
Breach	**95.7**	**95.8**	**95.7**	**95.5**	
DataProcessing	**95.7**	**96.4**	**95.7**	**95.4**	
Security	**97.1**	**98.7**	**97.1**	**97.7**	
Complaint/Request	**91.4**	**87**	**91.4**	**88.9**	
UserParticipation	**84.2**	**87**	**84.3**	**84.1**	

Table 5.1 Comparison of machine learning models' performance using Count Vectorizer.

5.2. Machine Learning Models with TF-IDF Vectorizer

In the "Security" category, the Random Forest and Decision Tree models both perform well, attaining the greatest accuracy, precision, recall and F1 score compared to the other two models. This implies that the strengths of these models are well-matched with the features recovered by TF-IDF and the underlying properties of this category. The Random Forest model obtains the best accuracy (95.7%) for the "DataProcessing" category, demonstrating that this particular category benefits more from Random Forest's ensemble method. The "UserParticipation" category shows much lower scores, especially for SVM (F1 score - 63.2%) and Random Forest models (F1 score - 65.9%), much like the outcomes with Count Vectorizer. This shows that these models have difficulty correctly categorizing cases under this category due to the complexity or the lack of clearly defined textual patterns. Overall, the Decision Tree model routinely performs better than alternative models across a range of assessment criteria and categories (Accuracy: Notice - 95.7%, Breach - 94.2%, DataProcessing - 91.4%, Security - 95.7%, Complaint/Request - 91.4%, and UserParticipation - 82.8%). In conjunction with TF-IDF Vectorizer, the Decision Tree approach is extremely efficient at identifying the text patterns. Table 1.2 displays the performance metrics of various machine learning models to categorize GDPR rules into categories for privacy needs using TF-IDF Vectorizer. For each category of privacy need, measures such as Accuracy, Precision, Recall, and F1 Score are used. Table 5.2 shows the comparisons of machine learning model performance for the TF IDF vectorizer.

TF-IDF Vectorizer					
Category	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Machine Learning Model**
Notice	80	70.6	80	73.5	**Logistic Regression**
Breach	90	81.5	90	85.4	
DataProcessing	81.4	67.6	81.4	73.5	
Security	90	81.9	90	85.5	
Complaint/Request	91.4	84.2	91.4	87.5	
UserParticipation	74.2	56.7	74.2	63.9	
Notice	92.8	94.4	92.9	92.7	**Random Forest**
Breach	90	81.5	90	85.4	
DataProcessing	**94.2**	**94.8**	**94.3**	**94.1**	
Security	**95.7**	**94.6**	**95.7**	**94.8**	
Complaint/Request	91.4	84.3	91.4	87.5	
UserParticipation	72.8	62	72.9	65.9	
Notice	88.5	92.4	88.6	87.4	**Support Vector Machine**
Breach	90	81.5	90	85.4	
DataProcessing	91.4	89.9	91.4	89.3	
Security	91.4	87.2	91.4	88.3	
Complaint/Request	91.4	84.3	91.4	87.5	
UserParticipation	72.8	56.5	72.9	63.2	
Notice	**95.7**	**96.4**	**95.7**	**95.7**	**Decision Tree**
Breach	**94.2**	**95.8**	**94.3**	**94.8**	
DataProcessing	91.4	93.1	91.4	91.4	
Security	**95.7**	**94.6**	**95.7**	**94.8**	
Complaint/Request	**91.4**	**87**	**91.4**	**88.9**	
UserParticipation	**82.8**	**85.8**	**82.9**	**82.5**	

Table 5.2 Comparison of machine learning models' performance using TF IDF Vectorizer.

5.3. BERT Model Results

The outcomes of using BERT to categorize GDPR legislation into several privacy-required categories show that performance varies across the various categories. It should be noted that BERT displays stronger accuracy and favorable performance metrics in several categories, including "Breach," "Security", and "Complaint/Request". On the other hand, its performance seems to be substantially lowered in areas like "Notice" "DataProcessing" and "UserParticipation". Several variables might contribute to these variances in BERT's efficiency. First, each category's content's inherent linguistic complexity may be a key factor. Accurate categorization may be difficult for some categories due to the more complex or specialized language usage. The availability and quality of training data also play a role in these discrepancies. In comparison to categories with little or confusing data, those with a large number of well-labeled training samples are likely to provide greater performance.

Additionally, changes might be brought about by the nature of the privacy needs itself. A language model like BERT may have difficulty correctly capturing subtle differences in some categories. Therefore, these complexities can potentially have an impact on BERT's predictions, changing performance. Table 1.3 displays the performance metrics of BERT models to categorize GDPR rules into categories for privacy requirements. For each category of privacy need, measures such as Accuracy, Precision, Recall, and F1 Score are used.

	BERT			
	Accuracy	**Precision**	**Recall**	**F1 Score**
Notice	78.6	83.5	78.6	73.5
Breach	92.8	86.2	92.8	89.4
DataProcessing	78.6	61.7	78.6	69.1
Security	92.9	86.2	92.9	89.4
Complaint/Request	92.9	86.2	92.9	89.4
UserParticipation	71.4	67.9	71.4	67.9

Table 5.3 BERT model performance.

5.4. Research Findings and Discussions

This section addresses two main research questions (RQ1 and RQ2) with a focus on the impact of various Vectorizer techniques and machine learning algorithms on GDPR classification into privacy requirements, as well as a comparison between conventional machine learning algorithms and pre-trained language models like BERT. To summarise:

RQ1: How does the combination of different vectorization methods and machine learning algorithms impact the overall accuracy of classifying GDPR into privacy requirements?

The objective of this research question is to determine the impact of integrating various vectorization techniques with various machine learning algorithms on the precision and effectiveness of separating GDPR laws into Privacy Requirements. The study investigates how the connection between these two crucial elements affects how well the categorization job is performed as a whole.

To better understand the synergistic impacts of feature representation and predictive models, the pairing of vectorization techniques like TF-IDF and Count Vectorizer with machine learning algorithms including SVM, decision trees, logistic regression, and random forest is studied.

The appropriate NLP data pre-processing is performed after extracting the required data. The data were converted into appropriate feature representations using count Vectorizer and TF-IDF pivotal Vectorizer techniques. However, the limited dataset size posed a challenge throughout the study. k-fold cross-validation is used to thoroughly assess model performance. With this method, the dataset was separated into subsets for training and validation, providing reliable findings and reducing the danger of overfitting.

Remarkably, the outcomes repeatedly showed that Count Vectorizer performed better than TF-IDF across all privacy requirement categories (Notice, Breach, DataProcessing, Security, Complaint/Request, and UserParticipation). This phenomenon is due to the nature of Count Vectorizer, which takes word frequency in a document into account. This approach, which considers the precise occurrence of terms and their associations inside each text, can be particularly useful when working with smaller datasets. As a result, it preserves complex language patterns and context, which are essential for comprehending the fundamental design of GDPR rules.

With more investigation, it appears that the decision tree model consistently outperformed other machine learning models when paired with Count vectorization (Accuracy: Notice - 97.1%, Breach - 95.7%, DataProcessing - 95.7%, Security - 97.1%, Complaint/Request - 91.4%, and UserParticipation - 84.2%). The decision tree's capacity to recursively divide the feature space depending on data properties may be responsible for its efficacy in this situation. The decision tree is skilled at locating relevant words and their combinations in the context of Count Vectorizer, which collects word occurrences, to provide unique categorization rules. Since the data can now be efficiently divided into many privacy requirement groups, the model may be effectively used.

RQ2: Do the traditional machine learning algorithms, such as SVM, decision trees, logistic regression, and random forest demonstrate superior performance compared to pre-trained language models like BERT in classifying GDPR into privacy requirements?

Overall, BERT's performance displays great results, but when compared to conventional machine learning methods, it has somewhat poorer accuracy in several areas. Category-wise evaluations is done for each model to offer a thorough comparison.

The Decision Tree model with Count Vectorizer often beats BERT for the categories "Notice," "Breach," "DataProcessing," "Security," and "UserParticipation." BERT, on the other hand, obtains a somewhat higher accuracy (92.9%) than the Decision Tree (91.4%) in the "Complaint/Request" category. But when compared with Random Forest and SVM, BERT gives better results for Breach (SVM accuracy=90%, Random Forest accuracy=90%) as well as Complaint/Request (SVM accuracy=91.4%, Random Forest accuracy=91.4%).

There are several reasons for the performance differences between BERT and other models.

1. The performance of BERT may be impacted by the underlying linguistic complexity within each category. Accurate classification is more difficult for categories that use more complex vocabulary.
2. These differences may be influenced by the volume and caliber of training data. The model's ability to predict outcomes may be impacted if some categories lack enough and properly labeled training data.
3. The privacy requirements themselves may be difficult due to their nature. BERT may have difficulty detecting subtle variations within categories, which might impact its predictions.

The differences in each model's performance across various circumstances are influenced by its unique advantages and disadvantages as well as the detailed features of its privacy requirements categories.

Chapter 6: Threats of Validity

It is crucial to take into account the threats to validity, which are possible sources of bias or inaccuracy that might affect the reliability of the findings. To ensure the accuracy and legitimacy of the study findings, the risks must be addressed. Here are some typical validity risks to the project:

6.1. Internal Validity

To ensure the correctness and reliability of the study findings, it was crucial for the Study to preserve internal validity. To do this, numerous parts of the study were meticulously regulated and standardized. Keeping outside effects to a minimum, the experimental variables are tightly controlled. Randomization and counterbalancing strategies are used to distribute biases and systematic mistakes uniformly. Data preparation was uniform, including the systematic handling of missing data and outliers through the use of Count Vectorizer and TF-IDF Vectorizer. To test machine learning model performance rigorously and prevent overfitting, k-fold cross-validation is used. By lowering the possibility of confounding variables and guaranteeing that the outcomes were unaffected by chance factors, these measures jointly helped to maintain internal validity. This dedication to internal validity strengthened the research's dependability and trustworthiness, enabling us to draw more confident inferences about how various vectorization approaches and machine learning algorithms affect the categorization of GDPR law into privacy requirements.

6.2. External Validity

In the Study, the external validity is ensured by using a diverse dataset representative of GDPR. Rather than being restricted to particular situations, this method allowed the findings to be generalizable across a variety of related circumstances. By being open and transparent about the study processes and methodologies, It is also made possible for other researchers to conduct similar studies using various conditions.

The external validity of the findings was also strengthened by the careful analysis of a range of machine-learning models and vectorization methods. With the help of this thorough methodology, practitioners and academics may use the findings to inform their choice of model and strategy for categorizing GDPR legislation into privacy needs, hence increasing the relevance and application of the study to a wide range of scenarios. Overall, the external validity of the Study was greatly enhanced by the dedication to diverse datasets, open methodology, and thorough technique investigation.

Chapter 7: Conclusion

The key points of my Study are discussed in this final part, including the solutions to the research questions, the contributions of the study, personal reflections, and potential future research areas

7.1. Study summary

The main goal of the Study is to study how NLP and machine learning, including cutting-edge models like BERT, help in classifying GDPR laws into privacy requirements.

The literature research shed light on the GDPR's underpinnings and introduced key NLP and machine learning techniques. The methodology chapter described the study methodology, data collecting, preprocessing with sophisticated AI tools such as ChatGPT, and assessment approaches.

The models are extensively evaluated in the evaluation chapter. To assess model efficacy, it incorporates a variety of performance indicators such as accuracy, recall, precision, F1 score, and confusion matrices. The comparison analysis identifies the most effective strategies.

We revealed surprising insights in the data analysis and findings chapter. The results of this analysis were scrutinized, revealing insight into the intricacies of each model's performance. Count vectorization outperforms TF-IDF, especially when used in conjunction with the decision tree model. BERT, while strong, performed inconsistently when compared to older models across GDPR areas.

The threads of validity chapter identifies and mitigates potential threats to research validity. It addresses issues like data quality and choice of evaluation metrics to ensure robust and credible results. Our contributions were summarised in the conclusion, which directed future research towards improved AI adaptation for GDPR compliance. Personal perspectives emphasized the research's interdisciplinary aspect, ethical issues, and the learning of vital skills.

7.2. Research Contributions

The study contributes significantly to the fields of data privacy and machine learning in several ways.

Model Selection Guidance: The Study provides suggestions for choosing machine learning models to categorize GDPR obligations into privacy requirements. The Decision Tree model performed best when combined with the Count Vectorizer in all categories. Making educated selections when selecting models for comparable text classification tasks is aided by this information.

AI-Driven Data Extraction: Demonstrated the value of advanced artificial intelligence models, such as GPT, for extracting and analyzing legal material, highlighting the potential for AI-driven automation in the legal and regulatory realms.

Benchmarking BERT: By contrasting BERT, a cutting-edge pre-trained language model, with conventional machine learning techniques, its advantages and disadvantages in this situation is identified. For academics and organizations contemplating the implementation of cutting-edge language models for privacy compliance duties, this benchmarking offers useful information.

Practical Implications: The conclusions of the Study have applications for businesses and organizations dealing with GDPR compliance. Identification and implementation of privacy requirements may be sped up by knowing which methods and models produce the most accurate classification of legislation.

Future Research Directions: The work contributes by highlighting potential directions for investigation in the future, such as looking into larger datasets, fine-tuning language models for certain domains, and expanding the research to additional NLP jobs in data privacy. These recommendations can direct future study in this important field.

Optimal Feature Representation: To assist academics and practitioners working with text data in data privacy scenarios, The study offers insights into the efficacy of various feature representation techniques.

7.3. Future research and development

The conclusions and consequences of this Study open up several promising directions for future study and development, including the application of cutting-edge AI techniques. The size of the dataset should first be increased as a top priority. A larger and more varied dataset, along with sophisticated AI tools like GPT for extracting input data, will allow for a more thorough assessment of machine learning models and language models like BERT. It would be intriguing to investigate how these models function when applied to a wider variety of GDPR laws, and for different non-functional requirements particularly those that are more complex and legally sophisticated.

Furthermore, optimising language models like BERT and GPT particularly for GDPR compliance may be a promising course of action. Such models' effectiveness and accessibility to organizations dealing with GDPR obligations may be improved by adapting them to the legal and regulatory language of data protection.

Furthermore, expanding this study to additional NLP activities in the context of data privacy and legal compliance, such as automated policy formulation and compliance evaluation, may provide insightful results. The potential of AI models like GPT for these jobs is enormous.

To comply with GDPR, it is critical to look into the interpretability of machine learning models, especially those based on GPT. In the legal and ethical implications of AI, there is increased interest in understanding how these models make their predictions and guarantee openness in the decision-making procedures.

The models, especially those based on GPT, must be adjusted by the constantly changing environment of regulations governing data protection. NLP models will need to be continuously monitored and modified to guarantee compliance with new legislation and revisions to old ones.

In conclusion, future research should aim to improve the performance, applicability, and ethical aspects of machine learning and NLP models, including advanced AI technologies like GPT, in the area of data privacy and legal compliance.

7.4. Practical Applications

This Study has broad implications for daily life. The model selection information offered, particularly in the context of categorizing legal material, is extremely valuable for enterprises and organizations aiming for GDPR compliance. Automation of labor-intensive data extraction and analysis processes, enabled by advanced AI technologies like GPT, has the potential to improve the efficiency and accuracy of compliance efforts.

Furthermore, the comparison of complex language models such as BERT and classic machine learning approaches provides the path for enterprises looking to use cutting-edge technology for privacy compliance. These insights can help companies make educated judgments about which tactics are most suited to their unique compliance requirements.

The proposals for future research areas, aimed at both academics and industry, give a clear roadmap for breakthroughs in data privacy and legal compliance by integrating AI and NLP. This study paves the way for novel ideas and practises, ultimately leading to more effective and ethical approaches to data protection and regulation.

7.5. Personal Reflections

This study journey has not only increased my academic comprehension but has also had a significant impact on my personal progress. It has reinforced the values of flexibility and determination in me, building in me an unflinching dedication to overcoming the complicated obstacles given by the combination of machine learning and data protection regulation.

The multidisciplinary approach used, combining cutting-edge AI technologies such as GPT with the intimidating GDPR framework, has greatly broadened my views. It has shed light on the enormous potential of interdisciplinary collaboration in addressing difficult societal concerns.

Furthermore, my analytical and critical thinking abilities have undergone significant development during my research trip, talents that will surely be put to use in my future research endeavors. I've also been increasingly aware of the ethical implications of artificial intelligence and data privacy, emphasising the critical importance of ethical and transparent AI research.

In essence, this Study has not only provided me with practical skills, but it has also sparked a strong interest in transdisciplinary research. It has strengthened my resolve to uphold ethical norms in the ever-changing universe of technology and legislation.

REFERENCES

Biswas, S. (2023). Prospective Role of Chat GPT in the Military: According to ChatGPT. *Qeios*.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), pp.5–32.

Casillo, F., Deufemia, V. and Gravino, C. (2022). Detecting privacy requirements from User Stories with NLP transfer learning models. *Information and Software Technology*, 146, p.106853.

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L. and Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, [online] 408(1), pp.189–215.

Charbuty, B. and Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, 2(01), pp.20–28.

Chirasmayee, B.V.S. (2022). Song Recommendation System using TF-IDF Vectorization and Sentimental Analysis. *International Journal for Research in Applied Science and Engineering Technology*, 10(6), pp.2475–2483.

Cleland-Huang, J., Settimi, R., Zou, X. and Solc, P. (2007). Automated classification of non-functional requirements. *Requirements Engineering*, 12(2), pp.103–120.

Das, M., Kamalanathan, S. and Alphonse, P. (2021). *A Comparative Study on TF-IDF feature Weighting Method and its Analysis using Unstructured Dataset*.

Deng, L., Hinton, G. and Kingsbury, B. (2013). New types of deep neural network learning for speech recognition and related applications: an overview. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*.

Deng, X., Liu, Q., Deng, Y. and Mahadevan, S. (2016). An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Information Sciences*, 340-341, pp.250–261.

Devlin, J., Chang, M.-W., Lee, K., Google, K. and Language, A. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. pp.4171–4186.

Elluri, L., Chukkapalli, S.S.L., Joshi, K.P., Finin, T. and Joshi, A. (2021). A BERT Based Approach to Measure Web Services Policies Compliance With GDPR. *IEEE Access*, 9, pp.148004–148016.

Hu, X. and Zhang, R. (2022). *Text classification based on machine learning*. [online] IEEE Xplore.

Intersoft Consulting (2018). *General Data Protection Regulation (GDPR)*. [online] General Data Protection Regulation (GDPR). Available at: https://gdpr-info.eu/.

Jindal, R., Malhotra, R. and Jain, A. (2016). Automated classification of security requirements. *Advances in Computing and Communications*.

Liaw, A. and Wiener, M. (2002). Classification and Regression by randomForest. 2(3).

Liu, S., Zhao, B., Guo, R., Meng, G., Zhang, F. and Zhang, M. (2021). Have You been Properly Notified? Automatic Compliance Analysis of Privacy Policy Text with GDPR Article 13.

Munaiah, N., Meneely, A. and Murukannaiah, P.K. (2017). A Domain-Independent Model for Identifying Security Requirements. *2017 IEEE 25th International Requirements Engineering Conference (RE)*.

Sangaroonsilp, P., Choetkiertikul, M., Dam, H.K. and Ghose, A. (2023a). An empirical study of automated privacy requirements classification in issue reports. *Automated Software Engineering*, 30(2).

Sangaroonsilp, P., Dam, H.K., Choetkiertikul, M., Ragkhitwetsagul, C. and Ghose, A. (2023b). A taxonomy for mining and classifying privacy requirements in issue reports. *Information and Software Technology*, 157, p.107162.

Sharma, S. (2019). *Data Privacy and GDPR Handbook*. John Wiley & Sons.

Štarchoň, P. and Pikulík, T. (2019). GDPR principles in Data protection encourage pseudonymization through most popular and full-personalized devices - mobile phones. *Procedia Computer Science*, [online] 151, pp.303–312.

Suraj Kumar Parhi and Sanjaya Kumar Patro (2023). Compressive strength prediction of PET fiber-reinforced concrete using Dolphin echolocation optimized decision tree-based machine learning algorithms. *Asian Journal of Civil Engineering*.

Thabtah, F., Abdelhamid, N. and Peebles, D. (2019). A machine learning autism classification based on logistic regression analysis. *Health Information Science and Systems*, 7(1).

Tri Julianto, I., Kurniadi, D., Septiana, Y. and Sutedi, A. (2023). Alternative Text Pre-Processing using Chat GPT Open AI. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 12(1), pp.67–77.

Van Hofslot, M., Salah, A., Gatt, A. and Santos, C. (2022). *Automatic Classification of Legal Violations in Cookie Banner Texts*. pp.287–295.

Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F. and Mueller, A. (2015). Scikit-learn. *GetMobile: Mobile Computing and Communications*, 19(1), pp.29–33.

Veluri, R.K., Patra, I., Naved, M., Prasad, V.V., Arcinas, M.M., Beram, S.M. and Raghuvanshi, A. (2022). Learning analytics using deep learning techniques for efficiently managing educational institutes. *Materials Today: Proceedings*, 51, pp.2317–2320.

Venkatesh, B. and Anuradha, J. (2019). A Review of Feature Selection and Its Methods. *Cybernetics and Information Technologies*, [online] 19(1), pp.3–26.

Voigt, P. and von dem Bussche, A. (2017). *The EU General Data Protection Regulation (GDPR)*. [online] Cham: Springer International Publishing.

Yadav, S. and Shukla, S. (2016). Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. *2016 IEEE 6th International Conference on Advanced Computing (IACC)*.

Zheng, Y., Gao, Z., Shen, J. and Zhai, X. (2022). Optimising Automatic Text Classification Approach in Adaptive Online Collaborative Discussion - A perspective of Attention Mechanism-Based Bi-LSTM. *IEEE Transactions on Learning Technologies*, pp.1–14.

Zhou, F., Zhang, Q., Sornette, D. and Jiang, L. (2019). Cascading logistic regression onto gradient boosted decision trees for forecasting and trading stock indices. *Applied Soft Computing*, 84, p.105747.

Zou, X., Hu, Y., Tian, Z. and Shen, K. (2019). *Logistic Regression Model Optimization and Case Analysis*. [online] IEEE Xplore.