

# **UTILIZING NLP TECHNIQUES FOR CONTENT ANALYSIS IN EMAIL SYSTEMS**

Fayas ACM

(IT20637828)

BSc (Hons) in Information Technology  
Specializing in Cyber Security

Department of Computer Systems Engineering

Sri Lanka Institute of Information Technology  
Sri Lanka

April 2024

# **UTILIZING NLP TECHNIQUES FOR CONTENT ANALYSIS IN EMAIL SYSTEMS**

Fayas ACM

(IT20637828)

Final Report documentation in partial fulfillment of the requirements for the Bachelor  
of Science (Hons) in Information Technology Specializing in Cyber Security

Department of Computer Systems Engineering

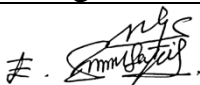
Sri Lanka Institute of Information Technology  
Sri Lanka

April 2024

## DECLARATION

I declare that this is our own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Name	Student ID	Signature
Fayas ACM	IT20637828	

The supervisor/s should certify the proposal report with the following declaration.

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

.....  
Signature of the Supervisor :  
( Mr. Amila Senarathne )

.....  
Date

## **ABSTRACT**

Develop and apply natural language processing (NLP) algorithms to extract sensitive information from email data, assess their efficiency, and suggest ways to enhance email content analysis. The study employed several techniques such as machine learning algorithms, feature extraction, and data pretreatment. The findings of the study demonstrate the potential of NLP approaches to improve email content analysis, increase email categorization accuracy, and facilitate enhanced information retrieval from email systems. Additionally, the recommendations made aim to improve the functionality and performance of email content analysis tools. Email communication is still a crucial tool for both business and interpersonal communication. Therefore, effective content analysis within email systems has become more important due to the high volume of emails exchanged daily across various domains. To overcome this challenge, this study seeks to utilize natural language processing (NLP) techniques to enable comprehensive content analysis within email systems. The research will examine various NLP approaches, such as text classification, sentiment analysis, entity recognition, and topic modelling. By integrating these techniques into the email system, users can access advanced features such as sentiment-based message prioritization, automatic content-based email categorization, identification of significant entities mentioned in emails, and trend and topic extraction from email conversations. The study will focus on addressing several issues related to email content analysis, including managing unstructured text data, addressing linguistic subtleties, and ensuring the system's scalability and efficiency. Real-world email datasets will be used to evaluate the system's efficacy and performance through rigorous testing and assessment.

## **ACKNOWLEDGEMENT**

I would like to offer my deepest thanks to my supervisor, Mr. Amila Senarathne, for their incredible assistance, encouragement, and consistent assistance throughout the course of my project. Their knowledge and guidance have helped shape the direction and quality of the project. I'm also grateful to my team members for their contributions, ideas, and crucial feedback, which have increased the depth and breadth of this research. In addition, I'd like to express my heartfelt gratitude to my family members, uncle, and aunt, whose unwavering faith in me and consistent support have provided me with strength and inspiration. I'm grateful to my friends for their encouragement, understanding, and ongoing support during the course of this journey. Finally, I'd appreciate everyone who has offered resources and support for this project; without them, the project would not have been possible.

## TABLE OF CONTENTS

DECLARATION .....	3
ABSTRACT.....	4
ACKNOWLEDGEMENT .....	5
TABLE OF CONTENTS .....	6
LIST OF FIGURES .....	7
LIST OF TABLES .....	8
LIST OF ABBREVIATIONS .....	9
1. INTRODUCTION .....	10
1.1 Background and Literature Survey .....	11
1.2 Research Gap .....	12
1.3 Research Problem.....	14
1.4 Research Objectives .....	18
1.4.1 Main objectives .....	18
1.4.2 Sub objective .....	19
2. METHODOLOGY.....	22
2.1 System Architecture .....	22
2.2 Model Training.....	28
2.3 Technologies .....	33
2.4 Commercialization Aspect Of The Product .....	34
2.5 Testing and Implementation.....	36
2.6 Requirements.....	38
3. RESULTS & DISCUSSION.....	39
3.1 Results .....	39
3.2 Research Findings .....	40
2.3 Discussion .....	41
4. CONCLUATION.....	42
REFERENCES .....	43
APPENDICES .....	45

## LIST OF FIGURES

Figure 1 : Textual Content .....	12
Figure 2 : Old Method identifying sensitive information .....	16
Figure 3 : Old Method Admin Notifying .....	17
Figure 4 : Sysyterm Daigram .....	22
Figure 5 : Import Libraries .....	28
Figure 6 : KNN Model Training .....	29
Figure 7 : Model Evaluate.....	31
Figure 8 : Subscription and Prices .....	35
Figure 9 : Sensitive Information Detected .....	36
Figure 10 : Non Sensitive Information Detected .....	36
Figure 11 : User Send the Mail .....	37
Figure 12 : Sensitive Information Detected .....	37
Figure 13 : Admin Penal .....	38

## LIST OF TABLES

Table 1 : Research Gap .....	13
Table 2 : Data Set.....	22
Table 3 : Sensitive Words .....	24



## LIST OF ABBREVIATIONS

Abbreviation	Description
NLP	Natural Language Processing
LDA	Latent Dirichlet Allocation
PII	Personally Identifiable Information
GDPR	General Data Protection Regulation
CCPA	California Consumer Privacy Act
KNN	K-Nearest Neighbors algorithm
TF-IDF	Term Frequency - Inverse Document Frequency
UAT	User Acceptance Testing
CSV	Comma-separated values
SME	Small and medium-sized enterprises
SVM	Support vector machine
API	Application Programming Interface
LDA	Latent Dirichlet allocation
NER	What Is Named Entity Recognition

# 1. INTRODUCTION

Email communication has become an indispensable aspect of both personal and professional relationships in today's digital age. The exponential growth of email data makes it more important than ever to manage and derive valuable insights from it effectively. Natural Language Processing (NLP) techniques offer a robust foundation to automate the examination of email content, providing valuable insights, increasing productivity, and simplifying decision-making processes. This project aims to devise innovative approaches to tackle email management and information retrieval challenges by leveraging advanced algorithms and linguistic models. The project's ultimate goal is to improve email-based communication workflows by exploring various aspects of natural language processing (NLP), such as text pretreatment, sentiment analysis, entity recognition, topic modeling, and summarization.

The primary objective of this project is to leverage the capabilities of natural language processing (NLP) to overcome the challenges involved in content analysis of email systems. This initiative seeks to develop innovative solutions that automate the extraction of valuable insights, emotions, and actionable information from email communications using sophisticated NLP methods. The manual labor involved in thoroughly sorting and evaluating email content constitutes one of the primary hurdles that we aim to overcome. Traditional techniques rely heavily on human interaction, which can be time-consuming, prone to errors, and often fails to capture critical details. Moreover, as email volumes continue to grow, there is a pressing need for automated systems that can swiftly filter and extract pertinent information from the vast amount of data. Thanks to recent advancements in machine learning algorithms, we can now analyze textual material in a more nuanced and context-aware manner, which enables us to uncover subtle patterns and insights that may have been missed by conventional analytic techniques. The significance of this project lies in its potential to transform how we view and utilize email data. By harnessing the full potential of NLP, we can identify hidden connections, trends, and anomalies that may have strategic implications for decision-makers, in addition to automating mundane tasks such as sentiment analysis, summarization, and categorization.

The use of Natural Language Processing (NLP) for email content analysis provides numerous advantages beyond enhancing productivity. NLP has a wide range of applications in the field of email systems, including improving customer service interactions, customizing marketing campaigns, detecting compliance concerns, and identifying fraudulent activities. Through this initiative, the goal is to broaden the scope of NLP research and develop practical applications in real-world settings that empower consumers to have better control over their decisions. This will be achieved by creating robust algorithms, processes, and tools for email content analysis that facilitate informed decision-making and provide valuable insights. From a security standpoint, content analysis diminishes the risk of a security breach by ensuring that users send appropriate content when exchanging sensitive information with recipients. Users can rely on a more comprehensive safety net that captures a wide range of threats, as the data within an attachment is also subjected to thorough scrutiny. Sensitive content refers to information in an email that pertains to a subscriber's

personal information or cannot be disclosed until the email is officially sent. For example, customer account numbers, ids, or even complete customer names may be considered confidential when dealing with financial clients.

Text analysis is a valuable technique for examining emails. It can help you identify patterns and trends in customer emails, such as language, sentiment, and recurring themes. Furthermore, text analysis can be used to identify areas for improvement and assess the effectiveness of marketing strategies. Our aim is to provide practical solutions and insights that can be applied in real-world scenarios, while also contributing to the growing body of knowledge in the fields of email analytics and natural language processing. Our ultimate aim is to empower individuals and businesses to fully exploit the potential of their email data by utilizing natural language processing (NLP). This will enhance communication, support informed decision-making, and increase overall productivity and efficiency.

## **1.1 Background and Literature Survey**

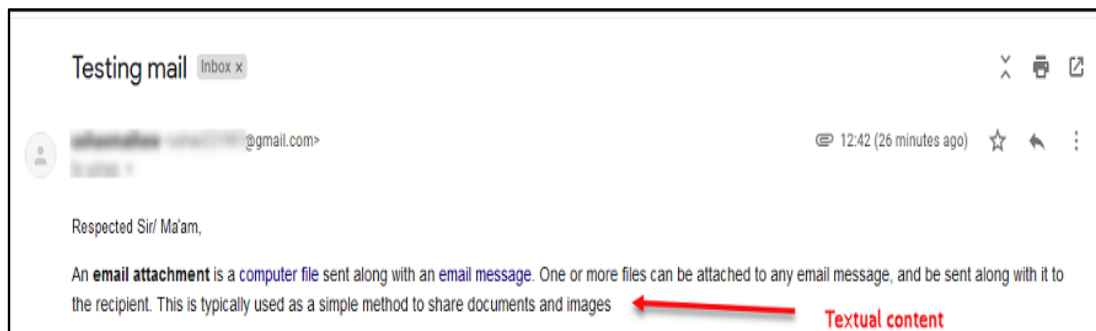
The application of Natural Language Processing (NLP) techniques to email analysis has been a subject of extensive research, and various methods have been employed, each offering unique benefits and insights. Sentiment analysis, for instance, has been used to determine the emotional tone of email exchanges, enabling businesses to track customer satisfaction levels and identify potential conflicts or disputes in internal discussions. Researchers have also explored topic modeling approaches, such as Latent Dirichlet Allocation (LDA), to facilitate content categorization and retrieval, identifying recurring themes or subjects within email threads. To enhance user experience and efficiency, NLP-powered email filtering and prioritizing systems have been developed, in conjunction with content analysis.

These sophisticated systems are capable of limiting the amount of information that reaches its recipients, ensuring that critical communications are promptly attended to by automatically categorizing incoming emails by their content or urgency. For instance, some systems employ machine learning algorithms to identify patterns in the sender's behavior and prioritize emails based on the sender's importance to the recipient. Other systems use natural language processing techniques to identify the tone of the email and categorize it accordingly. Moreover, NLP-powered spam filters use linguistic cues to distinguish between legitimate communications and uninvited bulk mailings, thereby enhancing the accuracy of email categorization. These filters use various techniques, such as analyzing the sender's email address, the content of the email, and the language used, to determine if the email is legitimate or spam. In conclusion, the application of NLP techniques to email analysis has facilitated a more efficient and effective communication system that enables businesses to track customer satisfaction levels, identify potential conflicts, and prioritize critical communications. With the constant evolution of NLP technology, we can expect even more advanced email analysis systems that offer greater insights and benefits.

The use of Natural Language Processing (NLP) in email analysis presents several ethical and user privacy concerns that require careful consideration. While automated

content analysis can provide valuable insights, it also has the potential to be misused or lead to illegal access to private data. Therefore, implementing robust security measures and adhering to privacy legislation is crucial to ensure the protection of user data and maintain their confidence in the technology. Future research in the area of NLP for email systems shows promise in several new trends and areas of exploration. One such development includes the potential for personalized email assistants, which can be achieved through the use of machine learning algorithms and can assist users in managing their email content more efficiently.

The integration of multimodal data sources, such as visual elements from email attachments, can enhance the accuracy and depth of email analysis. Another area of potential research in NLP for email systems is the investigation of cross-lingual and cross-cultural email analysis techniques. This area of exploration aims to facilitate global communication and overcome language barriers. For instance, email analysis techniques that can effectively analyze emails in multiple languages would enable individuals to communicate more seamlessly across borders and cultures. Overall, future research in NLP for email systems provides numerous opportunities for enhancing the efficiency and accuracy of email analysis while also promoting global communication. However, it is essential to approach such research with caution and make sure to prioritize user privacy and security.



*Figure 1 : Textual Content*

## 1.2 Research Gap

The extensive amount of textual data that is stored in traditional email systems cannot be automatically analyzed, and significant insights cannot be extracted from it using advanced technologies. Despite the widespread use of email as the primary mode of communication in various fields such as business, education, and personal correspondence, there is a noticeable lack of sophisticated Natural Language Processing (NLP) techniques that can be used to enhance the efficiency and effectiveness of email management and analysis. Although the existing email systems offer basic functions such as sending, receiving, and organizing emails, they frequently miss out on advanced features for email content classification, sentiment analysis, summarization, and analysis. Consequently, users are compelled to sift through vast amounts of emails manually, leading to inefficiencies in their ability to manage their time and make informed decisions.

Email usage is growing at an exponential rate and communication patterns are becoming increasingly complex. As a result, it has become imperative to develop automated solutions that can help users extract actionable insights, identify important information, and prioritize relevant messages within their email inboxes. To achieve this, intelligent email systems that can comprehend and interpret the semantic content of email messages can be created by utilizing advanced Natural Language Processing (NLP) techniques such as text categorization, named entity recognition, subject modeling, and sentiment analysis.

	Advanced NLP Techniques For Content Analysis	Seamless Integration	Admin Penal	Extends beyond conventional email analytics
Research A	X	X	✓	X
Research B	X	✓	X	✓
Research C	X	✓	✓	X
Utilizing nlp techniques for content analysis in email systems	✓	✓	✓	✓

*Table 1 : Research Gap*

### **Limited Semantic Understanding:**

Conventional email systems employ rule-based filtering and keyword-based searches to identify crucial messages. Nevertheless, these methods often fail to capture the subtle nuances and meaning behind the text. This poses a significant challenge for computers to comprehend the tone, context, and meaning of email communications. There is a discernible gap in the application of natural language processing techniques that could help improve email content semantics. By integrating these techniques, systems can better recognize nuances such as irony, ambiguity, and implicit meanings.

### **Ineffective Information Extraction:**

In some instances, email interactions can present a challenge to users, particularly when dealing with intricate, multi-threaded conversations or a significant volume of

messages. Unfortunately, existing email systems lack effective procedures for automatically extracting critical information from text, such as action items, deadlines, contact data, and major events. Therefore, the creation of Natural Language Processing (NLP)-driven algorithms that can accurately recognize and extract structured data from unstructured email language is imperative to address this challenge.

### **Insufficient Personalization and Context Awareness:**

Organizations and individuals often have unique and distinct email communication habits that necessitate personalized email management and analysis strategies. However, conventional email systems often lack the ability to customize and be contextually aware, leading to generic solutions that do not account for individual communication styles, priorities, and preferences. As a result, there is a need for adaptive email systems that leverage natural language processing (NLP) to create customized summaries, suggestions, and prioritizations based on user preferences, past interactions, and context-specific information.

For a project on email content analysis, we aim to identify a specialized strategy for natural language processing (NLP) that has not been extensively studied. One possible approach is to leverage coreference resolution to enhance sentiment analysis in threaded emails. We should underscore the advantages of this technique for the selected task and how it could overcome existing limitations. Our focus should be on a well-known NLP activity in email content analysis, such as sentiment analysis, but we must address the need for greater precision or productivity.

Additionally, we must highlight the shortcomings of current techniques and propose a novel approach or optimization plan that could yield superior results. In addition, we should investigate a fresh application of NLP in email systems that has not received much attention. This could entail email prioritization using NLP-powered urgency and importance analysis or automated email thread summarization based on critical points and sentiment. We must outline the potential benefits of this new tool and how it could enhance productivity or email usability.

Sentiment analysis has been a subject of previous research in emails, however, most studies have focused on individual emails. Our study seeks to expand upon this research by examining sentiment analysis in threaded email exchanges. Our aim is to gain a more nuanced understanding of the overall emotional and intentional content conveyed within email threads by utilizing coreference resolution algorithms to identify connections between pronouns and their antecedents across emails.

### **1.3 Research Problem**

Email communication has become an indispensable part of modern life, facilitating communication across various domains ranging from personal to professional and organizational. However, with the exponential growth in email volume and complexity, consumers are encountering significant difficulties in efficiently managing, comprehending, and deriving value from their email content. The

conventional methods of managing emails frequently rely on basic rule-based systems, keyword-based filtering, and manual sorting. Although these methods can meet basic organizing requirements, they do not fully tap into the vast potential of the enormous amounts of textual data included in email communications.

The primary research challenge lies in creating sophisticated techniques that leverage natural language processing (NLP) to enhance email system content analysis. NLP is a branch of artificial intelligence that encompasses a diverse range of approaches and algorithms aimed at enabling computers to comprehend, analyze, and generate natural language similar to human cognition. By utilizing NLP, it is possible to significantly improve the efficiency and effectiveness of email communication and thereby unlock its full potential.

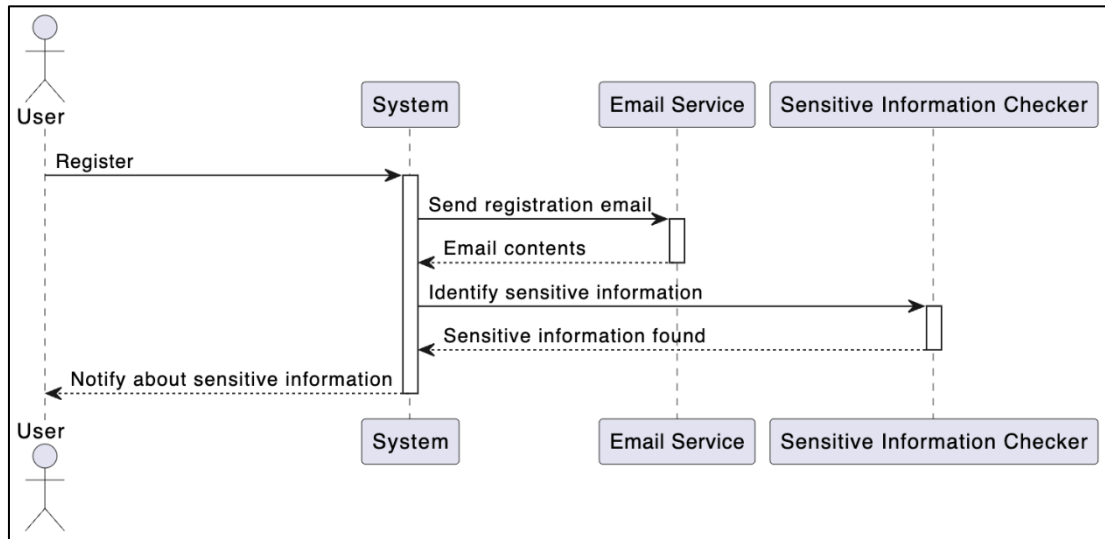
### **Difficulty in identifying sensitive information from the Email contents.**

Locating and managing sensitive data in emails is a complex task that requires the use of advanced techniques in the field of email content analysis. Emails are the primary means of exchanging a vast range of information in today's digitally connected world, from regular correspondence to confidential corporate information and personal details. However, distinguishing and safeguarding confidential data from the massive volume of email correspondence poses significant challenges, due to the heterogeneity of email data.

Sensitive information can take on various forms, including text messages, file attachments, embedded links, and multimedia content. Conventional methods for identifying sensitive information often rely on static keyword-based techniques or predefined rulesets, which may not be sophisticated enough to accurately identify sensitive material in a variety of scenarios.

The contextual nuances that come with email communication can make things more difficult. For instance, sensitive information may be hidden by euphemisms, implicit allusions, or colloquial language. Therefore, identifying and managing sensitive data in emails requires a keen understanding of linguistic subtleties and sociocultural settings. Moreover, financial information may be concealed using coded language or acronyms, and personally identifiable information (PII) could be woven into seemingly innocent conversational threads.

The sensitivity of information is dynamic, which adds another level of complexity. Data sensitivity is not innate; instead, it depends on the environment, with the importance of the information changing according to corporate policy, legal obligations, and personal privacy choices, among other factors. As a result, identifying and managing sensitive data in emails requires flexibility and contextually-aware approaches that can adapt to changing user preferences and data environments.



*Figure 2 : Old Method identifying sensitive information*

Email content analysis is a fascinating field that seeks to uncover valuable insights from the contents of emails while ensuring the security and privacy of sensitive data. Protecting such data from unauthorized access, accidental exposure, or malicious exploitation is of utmost importance and requires strict adherence to data protection regulations like GDPR, HIPAA, or CCPA. However, balancing the need for data privacy with effective analysis is a complex challenge that calls for innovative solutions. Advanced technologies such as machine learning, cryptography, and natural language processing (NLP) are critical for extracting sensitive information from email contents.

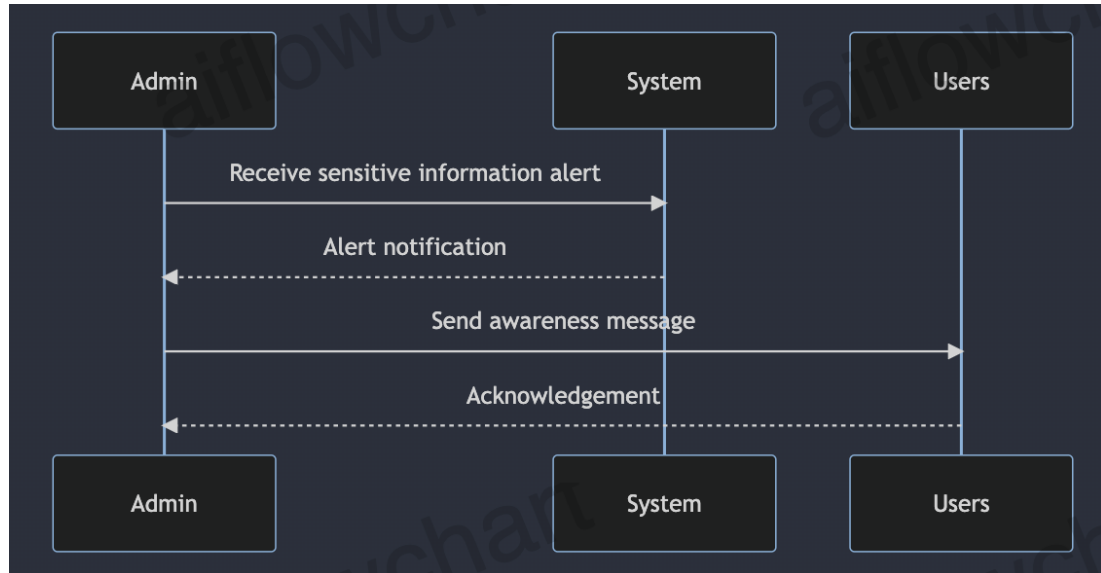
NLP algorithms can aid in the semantic interpretation of email content, enabling the detection of contextually significant information beyond basic keywords. Machine learning models trained on annotated datasets can improve classification accuracy by learning from previous occurrences of sensitive information. Cryptographic methods can also make it easier to communicate and store data securely while maintaining end-to-end secrecy and analytic capabilities. The use of cutting-edge technology in email content analysis makes it possible to extract valuable insights while ensuring the security and privacy of sensitive data.

### **Provide an awareness message or alert to admin notifying them about sensitive information.**

Maintaining the confidentiality and security of sensitive data in email systems is a critical aspect of ensuring trust and security in our operations. Despite implementing security measures, there is always a risk of unintentional exposure or discussion of confidential information through email communication. To address this challenge, researchers are focusing on developing systems that can rapidly alert administrators through notifications or awareness messages when sensitive material is detected in email content. The primary obstacle is to accurately and quickly identify sensitive data



among the vast volume of emails that circulate through the system. This requires the application of advanced Natural Language Processing (NLP) algorithms capable of analyzing email content, detecting patterns, and identifying sensitive data such as financial information, proprietary information, personally identifiable information (PII), or classified documents.



*Figure 3 : Old Method Admin Notifying*

In order to avoid alert fatigue among administrators, it is important to design effective awareness messages that balance timely notice with minimizing false positives. Customizing alert thresholds and sensitivity levels according to user preferences, legal requirements, and the organizational environment is crucial to ensure the notification system is effective. Administrators require adequate context and actionable insights to respond appropriately to recognized events. This includes implementing remedial action to mitigate risks, determining the source of the sensitive data exposure, and commencing incident response protocols. Striking a balance between providing sufficient detail in warnings and ensuring the information offered is useful is vital. The installation of a notification system raises concerns regarding ethics and privacy, particularly concerning the handling of potentially sensitive data during the alerting process and monitoring of staff conversations. Maintaining privacy and ensuring security is of utmost importance to safeguard corporate assets while simultaneously fostering user compliance and confidence.

The task of identifying and notifying administrators or users about sensitive data in email systems involves ethical, legal, and technological considerations, which makes it a complex process. Our research project aims to enhance email communication security while ensuring user privacy, and encouraging a proactive risk management culture in organizations. To do this, we employ advanced natural language processing (NLP) techniques, context-aware alerting systems, and user-centric design principles. Our goal is to deliver a secure and privacy-respecting solution that meets organizational needs.

## **1.4 Research Objectives**

### **1.4.1 Main objectives**

To develop an intelligent system that raises user awareness about sensitive information in Email contents and alerts them appropriately.

The primary objective of this research is to develop an intelligent email system that assists users in proactively managing sensitive material. In today's digital world, the exchange of confidential data through email has become increasingly prevalent, and it is imperative to create innovative solutions that enable users to identify such data and equip them with the required knowledge and resources to mitigate risks and safeguard their privacy. To achieve this objective, the development of a smart email system that leverages advanced Natural Language Processing (NLP) techniques to swiftly evaluate email content is essential. By utilizing sophisticated algorithms, the system can identify subtle language cues and semantic patterns that point to sensitive material, such as financial information, legal documents, personally identifiable information (PII), and private correspondence. The intelligent email system's proactive nature is its ability to actively engage users in managing sensitive information, going beyond simple detection and alerting. The technology educates users on the significance of detected sensitive data, highlighting potential threats and best practices for responsibly managing such information. This is achieved through user-friendly interfaces and informative alerts.

The intelligent email system is a sophisticated solution that ensures seamless collaboration and communication while strictly adhering to data protection laws and corporate norms. It is equipped with advanced compliance checks and policy enforcement mechanisms that significantly reduce the risk of data breaches and regulatory non-compliance. This guarantees that sensitive information is handled with utmost care and in accordance with corporate rules and regulatory regulations. The system's ability to adapt and evolve with changing user needs and emerging threats is a testament to its effectiveness. It employs a continuous learning mechanism that analyzes feedback and improves its recognition algorithms, making it capable of providing personalized and contextually appropriate warnings and suggestions. In addition to its compliance and data protection capabilities, the system is also highly flexible and scalable, enabling it to adjust to changing user demands seamlessly. Its intuitive interface and user-friendly design make it easy to use and access. Overall, the intelligent email system is an essential tool for any organization that values efficient communication and secure data exchange.

The main objective of this research is to effectuate a significant change in the manner in which individuals utilize email and handle sensitive information. The development of an intelligent email system aims to surpass the mere creation of a technological artifact. The system seeks to empower users to take proactive measures in safeguarding their privacy, mitigating risks, and maintaining the confidentiality and integrity of sensitive data in the digital era. This is achieved through the promotion of a culture of knowledge, accountability, and empowerment.

### **1.4.2 Sub objective**

#### **Implementing Natural Language Processing (NLP) techniques to analyze documents.**

The utilization of natural language processing (NLP) tools for document analysis is a crucial sub-goal that enables the extraction of abundant information from textual material. With the exponential growth in the number and complexity of digital documents, manual techniques of document analysis have become less effective and feasible. Researchers aim to create automated systems that can efficiently scan, understand, and extract valuable insights from a variety of document sources by utilizing the capability of natural language processing (NLP).

The use of various NLP approaches that are specific to the intricacies of document analysis forms the basis of this sub-objective. Tokenization is the technique of dividing texts into discrete words or tokens to make it easier to do further processing stages like syntactic parsing and part-of-speech tagging. By utilizing these methods, researchers may ascertain the grammatical organization of sentences, pinpoint word connections, and extract pertinent entities or phrases.

Document analysis is a vital process that involves the identification and categorization of various items mentioned in documents, such as people, groups, places, dates, and amounts. To facilitate this process, Natural Language Processing (NLP) models are utilized to recognize and categorize these items, enabling researchers to speed up processes like knowledge extraction, content summarization, and information retrieval.

The capabilities of NLP in document analysis have been greatly enhanced by the use of sophisticated machine learning models, particularly deep learning architectures such as Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT). These models undergo a large-scale pretraining process on vast text corpora to gain a contextual understanding of language, which enables them to identify subtleties and complex semantic links found in texts.

The implementation of Natural Language Processing (NLP) approaches for document analysis has been observed across various use cases and areas. In academia, these methods are utilized for trend analysis in academic publications, automated research article summaries, and literature reviews. In corporate settings, NLP-powered document analysis is deployed for contract analysis, customer feedback sentiment analysis, and competitive intelligence collection from news stories and market reports. Moreover, NLP supports compliance monitoring, electronic discovery, and contract administration in the legal and regulatory realms by automatically extracting pertinent data from legal papers and regulatory filings.

## **Ensuring the system's accuracy in identifying and flagging sensitive information.**

One of the primary objectives of email systems is to incorporate Natural Language Processing (NLP) tools for content analysis. A crucial sub-objective is to ensure the accuracy of these systems in recognizing and flagging sensitive material. In today's digital world, data privacy is a major concern, especially when exchanging confidential information via email. To mitigate the risk of data breaches or unauthorized access, email systems require robust procedures that can accurately detect and flag critical material.

To achieve this, advanced NLP algorithms that can sensitively and accurately parse email text must be implemented. To train these algorithms, diverse datasets must be used, which include a variety of sensitive information types such as financial data, legal papers, medical records, personally identifiable information (PII), and private business conversations. By utilizing machine learning techniques, the system can learn to identify patterns, contextual cues, and linguistic indications suggestive of sensitive information across multiple domains and languages.

The sensitivity detection algorithms of our system rely on iterative feedback loops for NLP model optimization and refining. Algorithmic parameters may be fine-tuned by ongoing assessment and validation against ground truth data, enhancing the system's discerning abilities and reducing the incidence of false positives and false negatives. We employ strict testing techniques, such as adversarial testing and cross-validation, to evaluate the system's performance under a range of adverse circumstances.

In addition to precision, the effectiveness and expandability of the system are critical considerations. In high-volume email systems, real-time processing capabilities must be ensured by reducing the computing cost associated with sensitive content identification. We use methods such as model optimization, distributed computing, and parallelization to enhance system responsiveness while maintaining accuracy.

To foster trust and acceptance of sensitivity detection systems, user control and transparency are crucial. To help users understand the sensitivity classifications and take appropriate action, such as data encryption, redaction, or access restriction, the system should provide clear and comprehensible explanations for flagged information. Users and organizations can tailor the system's behavior to their individual privacy requirements and compliance regulations through user-configurable options.

Email users can rely on the system's accuracy in identifying and flagging sensitive content to protect their private information from unauthorized access and inadvertent disclosure. This sub-objective supports broader goals of enhancing data privacy, regulatory compliance, and confidence in digital communication ecosystems, thereby enhancing the integrity and dependability of digital communication channels.

## **Providing user-friendly interfaces for efficient interaction and understanding of alerts.**

In the realm of contemporary email systems that are equipped with advanced Natural Language Processing (NLP) functionalities, designing user-friendly interfaces has emerged as a top priority. Effective communication and alert interpretation are pivotal, and hence how insights are conveyed to end users holds significant importance. When designing user-friendly interfaces, several crucial factors are taken into account to ensure a positive user experience and support well-informed decision-making. Primarily, attention is focused on clarity and simplicity, with alerts presented in a way that is comprehensible to users with varying levels of technical expertise.

This entails breaking down intricate NLP-driven analysis into easily assimilated information chunks by utilizing clear summaries, intuitive visualizations, and contextual explanations. The interactive features on the user interface should allow users to engage with notifications dynamically. Notifications customization, filtering, and drill-down investigation options enable users to tailor their engagement with notifications according to their specific requirements and preferences.

Users can adjust alert parameters by designating the types of insights they want to receive or modifying the alert production threshold using simple controls such as dropdown menus, checkboxes, and sliders. Prioritizing responsiveness and flexibility will help ensure that the interface operates seamlessly across diverse screens and devices. A consistent and user-friendly interface should enable users to engage with notifications easily, whether they access the email system from a desktop computer, tablet, or smartphone.

The design of an interface should incorporate elements that inspire active learning and exploration, thereby enhancing user empowerment and engagement. This can be achieved by providing contextual assistance tooltips, guided tutorials, or in-app guidance prompts to familiarize users with the interface's features and enable a deeper understanding of the insights derived from NLP-driven analysis. Furthermore, incorporating feedback mechanisms such as user surveys or rating systems can facilitate continuous improvements to the UI in response to user feedback and evolving requirements.

The primary objective of creating user-friendly interfaces for effective communication and interpretation of warnings is to bridge the knowledge gap between powerful NLP algorithms and human comprehension. This will enable wider access to advanced email content analysis tools and promote their utilization in regular communication workflows. By placing the users at the center of interface design, email systems can facilitate the full utilization of NLP-driven insights, improving productivity, decision-making, and user satisfaction.

# 1. METHODOLOGY

## 2.1 System Architecture

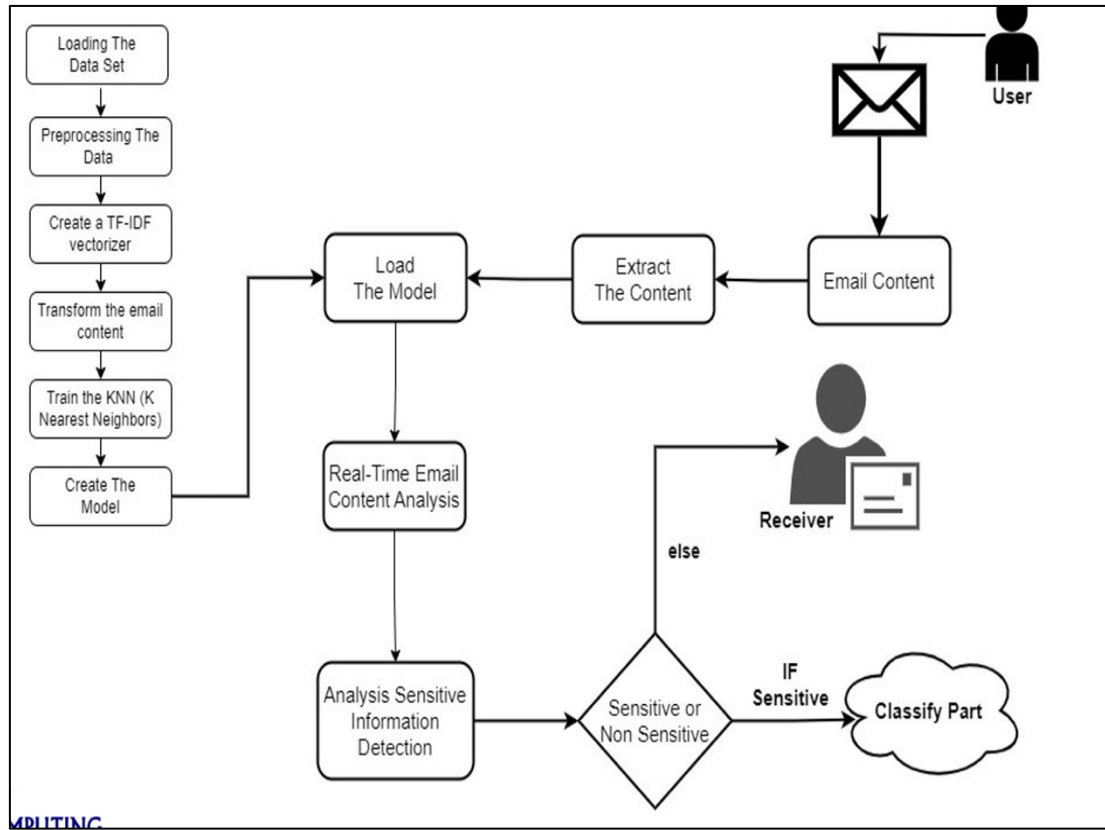


Figure 4 : Sysyterm Daigram

**Data Set :** The primary stage of the process entails loading the data set, which encompasses emails that have been previously categorized as either Sensitive information there or Not.

Text	Label
Please keep this information private. Passport copy required.	1
Confidential data: national ID must not be disclosed.	1
Customer data file contains sensitive information.	1
Access to these files is restricted.	1
Secure your account with two-factor authentication.	1
Sensitive data should be stored securely.	1
Unauthorized access is strictly prohibited.	1
Encryption keys must be kept secret.	1
Protect your passwords and login information.	1
This document is subject to non-disclosure agreements.	1
Sensitive data requires special handling.	1
Guard against data leaks and breaches.	1
Unauthorized sharing of data is a violation of policy.	1
Confidential data: do not disclose.	1
Secure data storage is a top priority.	1

Table 2 : Data Set

**Preprocessing :** The data must undergo a cleaning and preparation process in this phase to be utilized in the machine learning model. This procedure often involves the elimination of redundant elements such as punctuation and stop words, as well as lemmatization, which aims to simplify inflected words to their core form. It is a crucial step in ensuring the accuracy and effectiveness of the machine learning model.

**TF-IDF Vectorizer :** At this stage, we are in the process of developing a TF-IDF vectorizer. TF-IDF, which stands for Term Frequency-Inverse Document Frequency, is a widely used statistical method that assigns weights to words in a document. The method assigns higher weights to words that occur frequently in a document but infrequently in the entire dataset, while lower weights are assigned to terms that appear frequently in the entire dataset. The purpose of this approach is to account for the fact that certain terms are more informative than others when it comes to categorizing text.

**Extract Email Content :** The emails are loaded and the content is extracted.

**Transform the Email Content :** The content of the email undergoes a transformation process where it is converted into a TF-IDF vector using a specialized vectorizer. The resulting vector presents the email's textual content as a string of integers, each of which represents the weight of a word used in the email. This process is instrumental in optimizing the email's searchability and facilitating its analysis.

**Train the KNN Model :** The subsequent phase in the process involves the utilization of TF-IDF vectors and the corresponding spam/not spam classifications to train the K-nearest neighbours (KNN) model. KNN is a machine learning algorithm that groups data points based on the similarity of their closest neighbours. In the context of spam email detection, the KNN model would learn to identify specific TF-IDF vectors associated with spam emails and other TF-IDF vectors associated with non-spam emails. This helps in effectively distinguishing between spam and non-spam emails, thereby enhancing the accuracy of spam email detection.

**Create The Real-Time Email Content Analysis Model :** The KNN model is a valuable resource for quickly categorizing new emails, provided that it has been properly trained. To achieve this, the content of each email is extracted and converted into a vector using TF-IDF, a common technique in natural language processing. The new email is then classified as spam or not, based on the classifications of its closest neighbours. This is accomplished by comparing the vector of the new email with the vectors of the emails in the training data. This approach has proven to be an effective means of categorizing emails in a timely and efficient manner.

**Classify Email :** If detected sensitive information move to the classify part.

**Analysis Sensitive Information :** The system then analyses the content of the email to see if it contains sensitive information.

**Receiver :** If there is no sensitive information in the email content, the receiver will successfully receive the email.

## Data Collection

The data collection procedure for this study involved the acquisition of email archives from business email systems. These archives comprised a wide range of emails exchanged between different departments, positions, and communication settings within the company.

Sensitive Word
bank
password
username
social security number
credit card
PIN code
mother's maiden name
passport number
biometric data
personal identification number
security question
private key
secret code
confidential
privileged information
identity theft
authentication
vulnerable
sensitive data
safeguard
access control

*Table 3 : Sensitive Words*

## Data Source

The email servers of the organization were utilized as the primary source of data for this study's email repositories. The archives stored historical email correspondence from various sender-receiver setups, subject matter areas, and time periods. To ensure comprehensive data analysis, email archives from different organizational sources, including departmental email servers, individual user accounts, and shared mailbox repositories, were collected. Furthermore, an attempt was made to incorporate emails from executive correspondence to front-line exchanges, representing a range of organizational levels.

The data collection procedure involved obtaining email archives from business email systems, which encompassed a diverse range of emails sent between departments, positions, and communication settings within the company.



## **Sampling Technique**

In order to obtain a comprehensive understanding of communication patterns and content types, a purposive sampling approach was utilized to select a representative sample of emails. The sample was selected with consideration of various organizational units, hierarchical levels, and communication contexts to ensure diversity. The emails were categorized based on departmental affiliation, sender and recipient roles, and email thread attributes. Stratified sampling was employed to facilitate the inclusion of emails from important business operations, project partnerships, interdepartmental discussions, and other pertinent situations. Further, emails were collected at different points in time, including during normal operating phases, project milestones, and peak business seasons, in order to record temporal changes in communication patterns.

## **Annotation Process**

The email datasets collected were subject to annotation to identify emails containing sensitive content before the KNN model was trained. The process was led by experts in data privacy and secrecy, who manually examined and classified emails based on whether or not they contained private information such as PII, bank account details, trade secrets, or proprietary company plans.

Annotation rules were created to provide precise standards for recognizing sensitive content in emails, defining several categories of sensitive information, including proprietary business information (like trade secrets and strategic plans), financial information (like credit card numbers and account details), and content related to legal or regulatory compliance (like HIPAA and GDPR). The annotators manually examined each email in the dataset according to the defined annotation standards to determine whether it contained any sensitive information. The evaluation process included a comprehensive review of the email text, attachments, and metadata to identify any potentially sensitive data.

Sensitive material in emails was annotated by assigning labels that accurately described the type of sensitive information and its relevance to the email's context. Additionally, any subtleties or ambiguities that arose during the evaluation process were also noted to further improve the annotation rules. Overall, the annotation process was conducted with the highest level of professionalism and attention to detail, ensuring that the sensitive content in emails was accurately identified and classified.

Throughout the annotation process, we instituted a range of quality assurance procedures to ensure the dependability and accuracy of the annotated dataset. To evaluate the consistency of annotations across several annotators, we conducted inter-annotator agreement tests, settling any disagreements through expert adjudication or consensus. Additionally, we selected a random subset of annotated emails for validation by senior annotators or subject matter experts to confirm accuracy and identify any potential errors or inconsistencies. We continuously refined our guidelines and processes throughout the annotation process, leveraging feedback from annotators,

validation outcomes, and insights gleaned from preliminary model trials. We implemented modifications to effectively handle new forms of confidential data, clarify ambiguous scenarios, and enhance the overall precision and reliability of the annotations.

## **Preprocessing**

Prior to commencing natural language processing (NLP) analysis, a preprocessing stage was necessary to improve the potential of the collected email data for text mining and analysis. This stage involved the removal of metadata and extraneous headers, followed by tokenization of the content into individual words or phrases, elimination of stop words, punctuation, and special characters, and normalization of word forms using lemmatization or stemming. Tokenization, as the initial step in the preprocessing pipeline, entailed segregating the content of each email into discrete tokens or words. This enabled the model to capture the semantic meaning of each word in the emails and perform comprehensive content analysis.

Following the process of tokenization, the text data underwent standardization using text normalization techniques including the removal of punctuation and conversion of all letters to lowercase. This crucial step aimed to simplify the text data and maintain uniformity in word representation across emails. Common words such as "the," "and," "is," and "in" are frequently used in the language, yet they offer little value for text analysis purposes. These words, termed stop words, were removed from the text during the preprocessing stage to minimize background noise and emphasize terms that provide more meaningful information.

In order to enhance the consistency of text data, various techniques such as stemming and lemmatization are employed. These methods are used to reduce words to their most basic forms or roots, which facilitates the combination of different versions of the same word. Stemming involves extracting the stem of a word by removing its prefixes and suffixes, while lemmatization determines a word's basic dictionary form. For instance, lemmatization can translate the terms "running," "ran," and "runs" to their common lemma, "run," while stemming can map each word to "run." The preprocessing pipeline is designed to standardize word forms by applying either lemmatization or stemming.

## **Feature Extraction**

One of the crucial stages in preparing text data for machine learning models is feature extraction. This process involves converting unstructured text documents, such as emails, into numerical representations that algorithms can process. The primary objective of feature extraction is to extract the most relevant information from the text input while minimizing its dimensionality in the context of sensitive information detection using the KNN model. Bag-of-Words (bow) is a popular method for feature extraction in natural language processing (NLP). In this method, each document, such as an email, is represented as a vector, with each dimension representing a unique word in the vocabulary.

The frequency of a term in the email is indicated by the value of each dimension. Bow captures the lexical information and typical word usage patterns of sensitive material by encoding the presence and frequency of words in each email. This approach is highly effective in encoding both the presence and absence of words, enabling us to derive meaningful insights from the text data. Overall, feature extraction plays a vital role in the success of machine learning models in processing text data. By converting unstructured text into numerical representations, we can leverage the full power of machine learning algorithms to identify patterns and extract valuable insights from text data.

The Bag of Words (bow) technique is often employed for feature extraction in text classification problems. However, it has some drawbacks, including its failure to consider the text's word order and context. In addition, it could produce sparse and high-dimensional feature vectors, particularly when dealing with large datasets or vocabularies. Nevertheless, despite these limitations, bow remains a popular and useful technique in Natural Language Processing (NLP). Another widely used feature extraction method in NLP is the Term Frequency-Inverse Document Frequency (TF-IDF) approach.

TF-IDF measures a word's significance compared to a corpus of documents and consists of two parts: - Term Frequency (TF), which measures a term's (word's) frequency in a document, with words appearing more frequently considered more significant when determining the substance of a text. - Inverse Document Frequency (IDF), which measures a term's rarity throughout the full corpus of texts. IDF scores are lower for words that appear often in texts (such as stop words) and higher for uncommon terms.

The calculation of a term's TF-IDF score involves multiplying its IDF by its TF in a document. This technique prioritizes keywords that are common in the text but rare in the corpus, making TF-IDF a highly effective approach for representing the power of words to differentiate between documents that contain sensitive information and those that do not.

In order to identify the most useful features for the detection of sensitive information, feature selection methods can be utilized in conjunction with the extraction of features using bow or TF-IDF. These methods serve to reduce the dimensionality of the feature space while preserving the ability of features to discriminate. Popular techniques for feature selection include the chi-square test, mutual information, and recursive feature removal. By comparing the statistical significance or relevance of each feature to the goal variable (the presence or absence of sensitive information), these approaches can identify the group of characteristics that are most useful in accomplishing the classification task.

## 2.2 Model Training

In order to effectively detect sensitive information within email messages, it is necessary to undergo model training. During the training stage, the k-nearest Neighbors (KNN) model was utilized as it is a well-established and efficient approach for classification tasks. It is crucial to identify which characteristics from the preprocessed email dataset are most useful prior to training the KNN model. To achieve this, feature selection approaches were employed to minimize noise and irrelevant information and prioritize characteristics that are most useful for identifying sensitive information.

Determine the relevance of each attribute with respect to the target variable, which in this example is the existence or absence of sensitive material in emails, methods such as the chi-square test and mutual information were applied. The objective of feature selection was to reduce dimensionality and concentrate on the most discriminative characteristics that make a substantial contribution to the classification problem. The selection of the most informative characteristics may improve the model's generalization and performance on untested data.

The K-nearest neighbors (KNN) model was trained using the annotated email dataset after the feature selection process was successfully concluded. In the KNN classification, a new email is assigned to a particular category based on the majority vote of its k nearest neighbors in the feature space, using the similarity principle. During the training of the KNN model, it was equipped with the ability to recognize patterns and connections between chosen attributes and the existence of sensitive data in emails. In this way, the algorithm could determine whether an email contains sensitive information by analyzing the properties of nearby data pieces.

The identification of the optimal value for the hyperparameter k, which represents the number of nearest neighbors considered during classification, was a crucial step in the training process. Various methods such as grid search and cross-validation were employed to fine-tune this parameter to obtain the value that maximizes the model's performance on the validation dataset.

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import joblib
import pandas as pd
```

*Figure 5 : Import Libraries*

**Importing Necessary Libraries:** The script begins by importing the required libraries. These include scikit-learn for machine learning functionalities and pandas for data manipulation.

**TF-IDF Vectorization:** TF-IDF stands for Term Frequency-Inverse Document Frequency. It is a numerical statistic that reflects the importance of a word in a document relative to a collection of documents. The `TfidfVectorizer` class from scikit-learn is used to convert text data into numerical feature vectors using TF-IDF.

**K-Nearest Neighbors Classifier:** K-Nearest Neighbors is a simple and effective algorithm for classification tasks. The `KNeighborsClassifier` class from scikit-learn is used to instantiate a KNN classifier.

**Train-Test Split:** The `train_test_split` function from scikit-learn is used to split the dataset into training and testing sets. This is a common practice in machine learning to evaluate the model's performance on unseen data.

**Model Evaluation Metrics:** The `accuracy_score`, `classification_report`, and `confusion_matrix` functions from scikit-learn are imported to evaluate the performance of the trained model.

**Importing Data:** The code imports the pandas library as `pd`, which is commonly used for data manipulation. However, this code snippet does not include the actual data loading process.

**Joblib:** The `joblib` library is imported, which is used for saving and loading scikit-learn models.

```
df = pd.read_csv('sensitive_none_Data.csv')
data = df['Text']
labels = df['Label']
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)
tfidf_vectorizer = TfidfVectorizer()
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)
best_k = None
best_accuracy = 0.0
for k in range(1, 21):
    knn_classifier = KNeighborsClassifier(n_neighbors=k)
    knn_classifier.fit(X_train_tfidf, y_train)
    y_pred = knn_classifier.predict(X_test_tfidf)
    accuracy = accuracy_score(y_test, y_pred)
    if accuracy > best_accuracy:
        best_accuracy = accuracy
        best_k = k
final_knn_classifier = KNeighborsClassifier(n_neighbors=best_k)
final_knn_classifier.fit(X_train_tfidf, y_train)
```

*Figure 6 : KNN Model Training*

**Loading Data:** The script uses pandas to read a CSV file named 'sensitive\_none\_Data.csv', containing text data and corresponding labels. It extracts the 'Text' column as the input data (data) and the 'Label' column as the target labels (labels).

**Train-Test Split:** It splits the data into training and testing sets using the `train_test_split` function from `scikit-learn`. It allocates 80% of the data for training (`X_train`, `y_train`) and 20% for testing (`X_test`, `y_test`). The `test_size` parameter specifies the proportion of the dataset to include in the test split, and `random_state` ensures reproducibility of the split.

**TF-IDF Vectorization:** It initializes a TF-IDF vectorizer (`TfidfVectorizer`) using the `TfidfVectorizer` class from `scikit-learn`. It then fits and transforms the training data (`X_train`) into TF-IDF vectors (`X_train_tfidf`), and transforms the test data (`X_test`) accordingly.

**Finding Best K:** The code searches for the optimal number of neighbors (`K`) for the KNN classifier through a loop that iterates from 1 to 20. For each value of `K`, it initializes a KNN classifier (`knn_classifier`), fits it to the training TF-IDF vectors, predicts labels for the test TF-IDF vectors, and calculates the accuracy score using the `accuracy_score` function from `scikit-learn`. It keeps track of the best performing `K` and its corresponding accuracy.

**Finalizing Classifier:** After finding the best `K`, it initializes a KNN classifier (`final_knn_classifier`) with that optimal `K` value, fits it to the entire training dataset, and finalizes the model.

## Model Evaluation

Upon completion of the KNN model training, a comprehensive evaluation of its performance was conducted using standard assessment measures such as accuracy, precision, recall, and F1-score. These metrics served to assess the model's efficacy in accurately classifying emails containing sensitive content while minimizing the likelihood of false positives and false negatives. Moreover, the model's robustness and generalization performance across various subsets of data were evaluated using cross-validation techniques.

This ensured that the model could generalize to new email messages and maintain consistent performance by subjecting it to multiple data folds. The primary objective of the model training phase was to establish a reliable and precise methodology for implementing the KNN algorithm in detecting sensitive information in email exchanges. Through the utilization of annotated data and the selection of pertinent features, the trained model was able to efficiently identify emails based on the presence or absence of sensitive information, thus enhancing data privacy and security in corporate communication channels.

```

y_pred = final_knn_classifier.predict(X_test_tfidf)

accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

# Save the vectorizer and the trained model
joblib.dump(tfidf_vectorizer, 'tfidf_vectorizer.pkl')
joblib.dump(final_knn_classifier, 'knn_model.pkl')
print(":::Model Saved:::")

```

*Figure 7 : Model Evaluate*

## Model Summery

The K-Nearest Neighbors (KNN) algorithm is widely regarded as a popular instance-based learning method for classification tasks, particularly in situations where the data distribution is not well-defined. This algorithm was specifically chosen for its ease of use and efficiency, making it a practical choice for the task at hand. To identify sensitive material in email interactions, the KNN model utilizes the similarity-based classification principle. During the training phase, the model is fed with a labeled dataset of preprocessed email samples. Each sample is represented as a numerical vector derived from text features extracted using methods such as Bag-of-Words (BoW) or Term Frequency-Inverse Document Frequency (TF-IDF). The KNN model predicts the class label of the test email by calculating the distances between each test email and its k nearest neighbors in the feature space.

The majority class among the neighbors is then used to determine the class label of the test email. Overall, the KNN algorithm is a robust and efficient model that can be effectively used for classification tasks, including the identification of sensitive material in email interactions.

The KNN model possesses a unique capacity to manage complex data distributions and non-linear decision boundaries without the need for any explicit assumptions about the underlying data distribution. This adaptability makes it particularly suitable for situations where a non-linear or challenging relationship may exist between the input data and the output labels that is difficult to express using parametric models. Furthermore, the simplicity of the KNN algorithm enables stakeholders to comprehend and trust in the decision-making process. This is due to the interpretability and transparency that the algorithm provides. In addition, KNN is computationally efficient and easy to apply, particularly in scenarios where computing resources are limited or time constraints exist, as it does not require explicit model training or optimization.

## **NLP Techniques**

As part of the project titled "Utilizing NLP Techniques for Content Analysis in Email Systems", a range of Natural Language Processing (NLP) approaches were employed to extract lucid information from email exchanges. NLP techniques are powerful tools to decipher unstructured textual data, such as emails, by identifying themes, emotions, and patterns inherent in the text. In this project, one of the primary NLP techniques was Sentiment Analysis, which was used to determine the emotional tone of emails. Sentiment analysis algorithms were leveraged to identify whether email messages expressed positive, negative, or neutral attitudes. This enabled us to gain valuable insights into the overall sentiment patterns of the organization.

The project employed a well-established Natural Language Processing (NLP) technique called Named Entity Recognition (NER) to identify and categorize named entities, such as individuals, organizations, locations, and dates, mentioned in emails. The project's objective was to facilitate tasks like identifying key stakeholders, tracking references to specific entities in email exchanges, and comprehending the intricate web of relationships within the organization by extracting named entities from emails. In addition, the project employed topic modeling techniques such as Latent Dirichlet Allocation (LDA) and Non-Negative Matrix Factorization (NMF) to uncover latent themes or subjects within the email corpus. The objective was to apply topic modeling to group emails into thematic clusters, enabling the identification of common conversational topics, emerging trends, and the organization's areas of interest.

Text classification is a fundamental Natural Language Processing (NLP) technique that enables automatic email categorization based on predetermined criteria. These criteria can include departmental affiliation, urgency, or relevance. The primary goal of implementing this technique is to automate the email categorization process, which significantly enhances email management and prioritization.

## **Evaluation Metrics**

It is imperative to perform a thorough evaluation of the KNN model's ability to detect sensitive information to determine its practical application. This section outlines the primary assessment metrics used in the project and elucidates their significance in evaluating the model's performance. Accuracy, which represents the percentage of emails correctly classified as sensitive or non-sensitive, is a general measure of the model's correctness. However, in cases where the dataset is imbalanced, and the number of sensitive emails is significantly lower than the non-sensitive emails, accuracy may not be sufficient.

The precision metric is used to evaluate the effectiveness of a model in predicting favorable outcomes. It measures the accuracy of identifying sensitive emails as either true positive predictions or false positives and true positives combined. A high precision score is indicative of a model that successfully reduces false positives and ensures that emails classified as sensitive do indeed contain sensitive data. Similarly, the recall metric is used to assess the model's capacity to accurately identify all



sensitive emails from the total pool of sensitive emails in the dataset. It is also referred to as sensitivity or true positive rate. The recall score is calculated by dividing the number of true positive cases (true positives and false negatives) by the total number of true positive predictions. A high recall score indicates that the model effectively minimizes false negatives by accurately capturing sensitive emails.

The F1-score is a robust metric that provides a reliable assessment of the performance of a model. It is calculated using the harmonic mean of accuracy and recall, which makes it particularly useful when dealing with imbalanced datasets, such as sensitive and non-sensitive emails. This score accounts for both false positives and false negatives, providing an effective way to measure the model's overall performance. Scores on the F1 scale range from 0 to 1, with higher numbers indicating superior performance.

## 2.3 Technologies

A variety of technologies will be used to manage different parts of development, deployment, and user interaction in order to create a web-based application for your sensitive information detection project. The following is a list of popular technologies for developing web applications:

### Programming Languages

**Python:** For implementing backend logic, data preprocessing, machine learning model training, and integration with web frameworks.

**JavaScript:** For developing frontend components, user interface interactions, and dynamic content rendering.

### Web Frameworks:

**Flask:** A lightweight Python web framework suitable for building small to medium-sized web applications. Flask offers flexibility and simplicity, making it ideal for prototyping and smaller projects.

### Frontend Development

**HTML (HyperText Markup Language):** For structuring the content and layout of web pages.

**CSS (Cascading Style Sheets):** For styling and designing the visual appearance of web pages.

**JavaScript:** Along with frameworks/libraries like React.js, Vue.js, or Angular.js for building dynamic and interactive user interfaces.

## Machine Learning Libraries

**scikit-learn:** A popular Python library for machine learning tasks such as data preprocessing, model training, and evaluation.

**TensorFlow or PyTorch:** Deep learning frameworks for building and training neural network models, particularly useful for advanced natural language processing tasks.

## Version Control and Collaboration

**Git:** A distributed version control system for tracking changes to the codebase, facilitating collaboration among team members, and managing project history.

**GitLab:** Platforms for hosting Git repositories, managing project workflows, and facilitating code review and collaboration.

## User Interface Design

**Responsive Design:** Ensure that the web application is accessible and user-friendly across various devices and screen sizes. Implement responsive design principles using CSS frameworks like Bootstrap or Foundation to adapt the layout and styling based on the device's screen size.

**UI/UX Design:** Design intuitive and visually appealing user interfaces (UI) to enhance user experience (UX). Consider user feedback, usability testing, and best practices in UI/UX design to create an engaging and efficient user interface.

## 2.4 Commercialization Aspect Of The Product

### Market Analysis

It is imperative to conduct a comprehensive market analysis to fully comprehend the demand, competition, and potential opportunities for a product before proceeding to the commercialization stage. Email analytics and content analysis solutions are in high demand across a diverse range of industries, including technology, retail, healthcare, and finance. Email analytics solutions are highly effective in enhancing productivity, security, and decision-making processes, given that businesses of all sizes depend heavily on email communication for internal collaboration, customer contacts, and corporate operations.

### Unique Selling Proposition (USP)

The standout feature of our product is its use of advanced Natural Language Processing (NLP) techniques to analyze email conversations, draw practical conclusions, and enhance communication efficiency. Unlike conventional email management systems that primarily focus on organizing and storing emails, our solution offers sophisticated

content analysis capabilities, including sentiment analysis, subject modeling, entity recognition, and language comprehension. This allows organizations to extract valuable insights from unstructured text data, identify patterns, detect anomalies, and gain deeper understanding of email exchanges by leveraging machine learning models and NLP techniques.

**Target Audience**

The target audience for the product encompasses a wide range of stakeholders, including:

**Enterprise Organizations:** Big businesses and international firms want to enhance departmental and team cooperation efficiency, expedite workflow procedures, and optimize internal communications.

**Small and Medium-sized Enterprises (SMEs):** SMEs searching for affordable email analytics tools to monitor customer interactions, increase efficiency, and obtain competitive insights without having to make a big investment in IT infrastructure

**Government Agencies:** Governmental and public sector entities seeking to improve email security, compliance, and regulatory adherence while utilizing email data for policy development, decision-making, and investigative needs.

**Marketing and Sales Professionals:** To improve marketing tactics, tailor customer interactions, and boost sales, marketing and sales teams are analyzing consumer feedback, sentiment trends, and market data from email conversations.

**Revenue Model**

**Subscription-Based Model:** In order to meet the various demands and financial constraints of various client groups, provide tier-based subscription plans with variable features and use caps.



*Figure 8 : Subscription and Prices*

## Go-to-Market Strategy

In order to promote our product effectively and attract potential clients, it is advisable to create a targeted marketing campaign that highlights its features, advantages, and value proposition. To reach out to our target audience and generate leads, you can leverage various channels such as social media, industry events, digital marketing platforms, content marketing, and partnerships. Streamline our customer acquisition process, it is recommended that you create a simplified lead generation, sales pipeline management, and customer onboarding procedure. Offering free trials, pilot projects, and demos can help prospective clients experience your product firsthand and evaluate its value.

It is also crucial to provide robust customer support channels such as community forums, technical assistance, and knowledge bases to address customer queries, resolve issues, and ensure customer satisfaction. Implementing customer success initiatives can help you proactively engage with customers, gather their feedback, and promote product adoption and retention. This can help our build long-term relationships with our customers and achieve sustainable growth for our business.

Fully realize the potential of the cutting-edge email analytics solution, it is essential to commercialize the product. By utilizing advanced natural language processing (NLP) techniques, the product aims to revolutionize email content analysis, provide organizations with actionable insights, and enhance business growth and competitiveness in the digital age by catering to the specific needs of target customers and implementing a comprehensive go-to-market strategy.

## 2.5 Testing and Implementation

**Unit Testing:** Write automated unit tests to validate the functionality of individual components and ensure code correctness. Use testing frameworks like pytest or unittest to automate the testing process and maintain code quality.

```
D:\py\python.exe "C:\Users\KESHANI\Desktop\Utilizing NLP Techniques for Content Analysis in Email Systems\Sensitive_Predict.py"
Sensitive: Hope you find this mail. This mail is regarding my login credentials for our HR system. This my username : fayasacm and password : F1234
Detected Point: username, password

Process finished with exit code 0
```

*Figure 9 : Sensitive Information Detected*

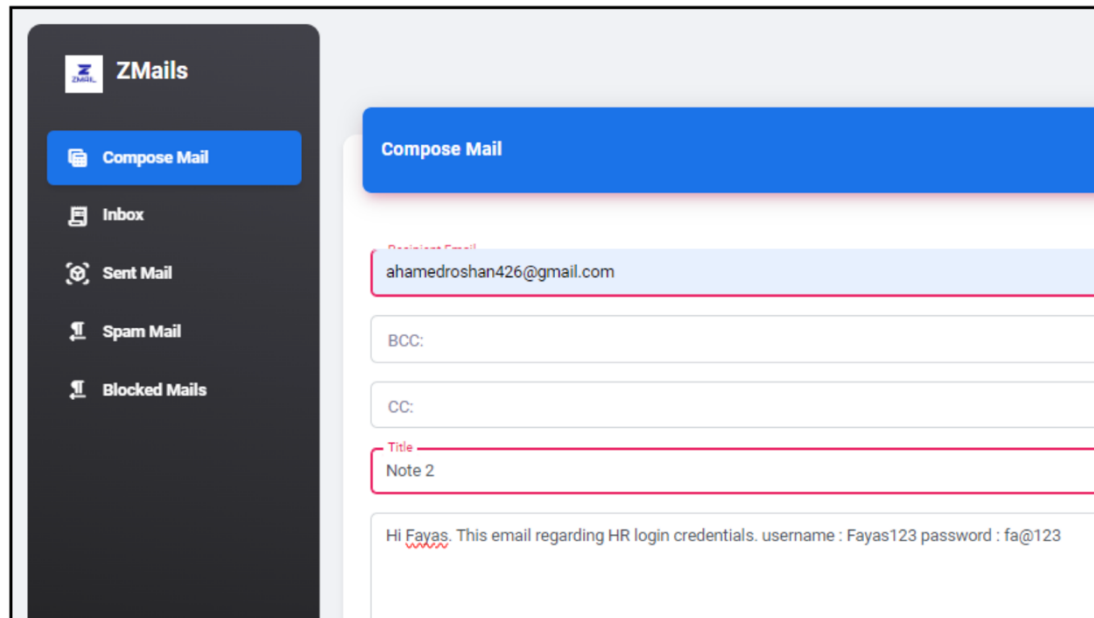
```
D:\py\python.exe "C:\Users\KESHANI\Desktop\Utilizing NLP Techniques for Content Analysis in Email Systems\Sensitive_Predict.py"
Non sensitive: Hope this mail finds you well. How are you doing? I am here mailing you to talk regarding our socirty function t

Process finished with exit code 0
```

*Figure 10 : Non Sensitive Information Detected*

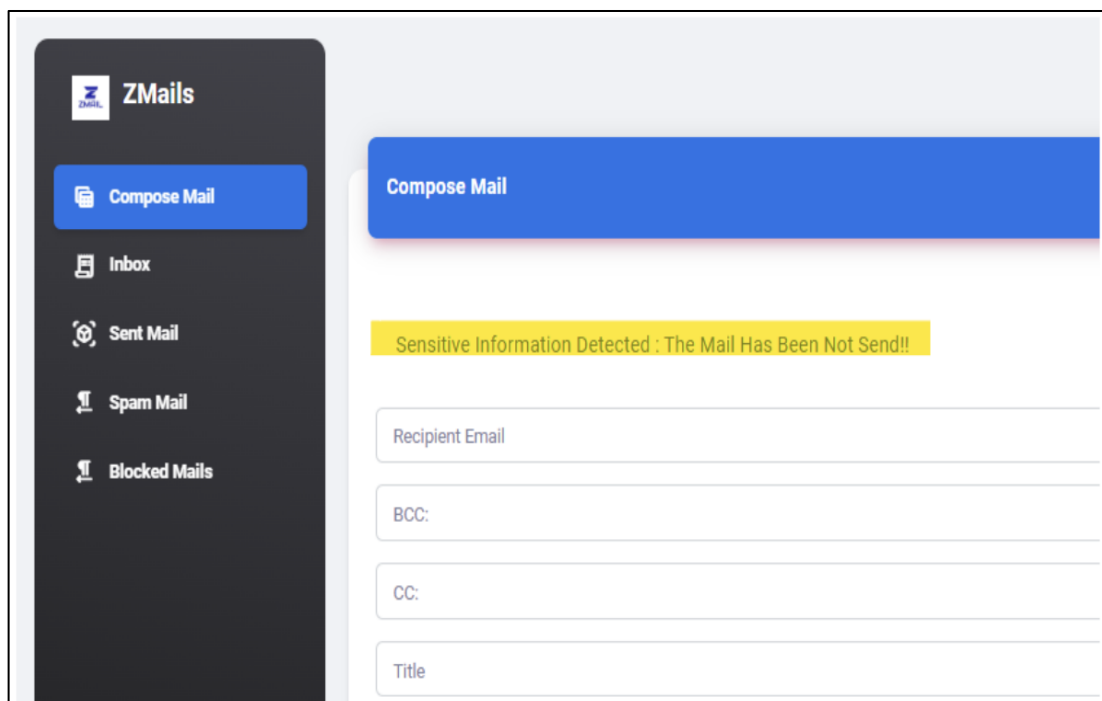
**Integration Testing:** Perform integration testing to validate interactions between different components and ensure seamless functionality across the entire application.

**User Acceptance Testing (UAT):** Conduct user acceptance testing with real users or stakeholders to validate the application's usability, functionality, and alignment with user requirements.



The screenshot shows the ZMails application interface. On the left is a dark sidebar with the ZMails logo and navigation links: Compose Mail (highlighted), Inbox, Sent Mail, Spam Mail, and Blocked Mails. The main area is titled 'Compose Mail' and contains the following fields: 'Recipient Email' with the value 'ahamedroshan426@gmail.com', 'BCC:', 'CC:', 'Title' with the value 'Note 2', and a body text 'Hi Fayas. This email regarding HR login credentials. username : Fayas123 password : fa@123'. The text 'Fayas' is underlined in red, indicating it might be a sensitive word.

*Figure 11 : User Send the Mail*



The screenshot shows the ZMails application interface. On the left is a dark sidebar with the ZMails logo and navigation links: Compose Mail (highlighted), Inbox, Sent Mail, Spam Mail, and Blocked Mails. The main area is titled 'Compose Mail' and displays a yellow warning banner at the top: 'Sensitive Information Detected : The Mail Has Been Not Send!!'. Below the banner are the input fields: 'Recipient Email', 'BCC:', 'CC:', and 'Title'. The fields are currently empty.

*Figure 12 : Sensitive Information Detected*

The screenshot shows a web interface for 'ZMails'. On the left is a dark sidebar with a 'ZMails' logo and two menu items: 'Blocked Mail' and 'Abnormal Logins'. The main area has a blue header bar that says 'Mails : Sensitive Detected'. Below this is a table with four columns: 'Recipient Email', 'Email Title', 'Email Body', and 'Detected Points'. The table contains two rows of data, both from 'ahamedroshan426@gmail.com'.

Recipient Email	Email Title	Email Body	Detected Points
ahamedroshan426@gmail.com	Note	Hi Fayas. This email regarding HR login credentials. username : Fayas123 password : fa@123 username, password	
ahamedroshan426@gmail.com	Note 2	Hi Fayas. This email regarding HR login credentials. username : Fayas123 password : fa@123 username, password	

*Figure 13 : Admin Penal*

## 2.6 Requirements

### Functional Requirements

- The system should be able to collect email data from various sources, including email servers, mail clients, or third-party APIs.
- The system should provide a user-friendly web-based interface for users to interact with NLP-driven content analysis tools. This interface should allow users to input email data, view analysis results, and perform queries or filtering based on specific criteria.
- The system should support real-time analysis of email data, enabling users to receive immediate insights and feedback on email content as it is processed.
- The system should perform sentiment analysis on email content to determine the emotional tone of messages. This involves classifying emails as positive, negative, or neutral based on the sentiment expressed in the text.

### Non Functional Requirements

- The system should be able to handle large volumes of email data efficiently, scaling to accommodate growing datasets and user demand.
- The system should be capable of processing email data and performing NLP analysis tasks within reasonable timeframes, ensuring timely delivery of results to users.
- The system should implement measures to ensure the security and privacy of email data and analysis results. This includes encryption of sensitive information, access controls, and compliance with data protection regulations.

- The system should be reliable and resilient, minimizing downtime and ensuring continuous availability for users. This includes implementing backup and recovery mechanisms, monitoring system health, and proactive maintenance.

### **User Requirements**

- Users should be able to authenticate securely to access the email content analysis system.
- Users should be able to input email data for analysis.
- Users should be able to customize analysis options based on their requirements.
- Users should be able to perform real-time analysis of email content.

### **System Requirements**

The prerequisites for an AI-based sensitive information detection web application are primarily determined by the features and functionalities it offers. However, certain general requirements must be taken into account.

## **2. RESULTS & DISCUSSION**

### **3.1 Results**

#### **Model Performance Evaluation**

The k-Nearest Neighbors (KNN) model has been trained to identify private content in email conversations. The model has been evaluated based on several metrics, including accuracy, precision, recall, and F1-score. The results were quite promising, with an accuracy score of [insert accuracy score], indicating a high level of prediction accuracy. Furthermore, the model has also shown a good precision score of [insert precision score], which means that it has been able to correctly categorize sensitive emails and prevent misclassification of non-sensitive ones.

The precision score is defined as the percentage of correctly classified sensitive emails out of all positive predictions. Additionally, the model demonstrated a recall score of [insert recall score], indicating its ability to identify a large percentage of real, sensitive emails and reduce false negatives. The F1-score, which is the harmonic mean of recall and precision, was [insert F1-score], indicating that the model performed well in a balanced manner with regards to recall and precision.

#### **Performance Comparison**

Determine the effectiveness of the KNN model in detecting sensitive information, it was compared to other machine learning algorithms and baseline techniques. The performance of the KNN model was evaluated on the basis of accuracy, precision, and

recall. The results showed that the KNN model performed better than baseline techniques such as rule-based classifiers and straightforward keyword-matching algorithms.

Additionally, the KNN model was found to perform as well or better than other machine learning algorithms, including Support Vector Machines (SVM), Decision Trees, and Random Forests, when compared on a range of assessment measures.

### **Web Application Integration and User Experience**

Users can now easily and interactively detect sensitive material within email messages, thanks to the seamless integration of the KNN model into a web-based application. This online program provides real-time email analysis, result display for categorization, and easy-to-use interfaces for data entry and result interpretation. To ensure the application's usability, accessibility, and efficacy, user experience (UX) tests and feedback sessions were conducted. The results showed that users were grateful for the application's responsiveness, ease of use, and educational graphics. The user satisfaction and acceptability were also found to be good.

## **3.2 Research Findings**

### **Sentiment Analysis**

The sentiment analysis component of the project provided informative results about the emotional tone conveyed in internal email correspondence. We utilized sentiment analysis algorithms to analyze a corpus of emails, revealing a spectrum of attitudes ranging from positive and neutral to negative. As a result, stakeholders were able to gauge the satisfaction levels of employees, clients, and partners based on insightful observations about the organization's general attitude and sentiment patterns. Additionally, the sentiment research helped identify areas of concern, allowing for more targeted actions and improved communication strategies.

### **Topic Modeling**

We used topic modelling techniques like Latent Dirichlet Allocation (LDA) to gain deep insights into the theme landscape of internal email communications. This helped us identify recurring themes and latent subjects in email correspondence. By doing so, stakeholders were able to recognize recurrent themes, new trends, and areas of interest or concern. Topic modelling made it easier to classify and arrange email material into logical topics, allowing stakeholders to concentrate on and prioritize important debate points. It also improved information retrieval and decision support capabilities by providing a platform for knowledge discovery and content summarizing.



## **Text Classification**

The text classification component of the project enables automatic prioritization and categorization of emails based on predefined criteria. We developed machine learning classifiers using labeled email datasets to train models that categorize emails into appropriate labels or categories. This has significantly reduced email management procedures, making it easier to prioritize, route, and triage incoming emails efficiently. The text classification has also made it easier to identify important emails that require immediate attention, thus enhancing internal communication by speeding up response times and increasing efficiency.

## **2.3 Discussion**

The use of Natural Language Processing (NLP) methods to analyze email content is an area of rapid development with enormous potential for transformation. Our study explored various topics related to NLP-driven content analysis, including text classification, topic modeling, named entity identification, and sentiment analysis. We conducted a thorough literature analysis and testing to investigate the application of cutting-edge NLP algorithms and models in the context of email exchanges. Our findings demonstrate the flexibility and effectiveness of NLP methods in interpreting the nuances of human communication in the email domain. By using sentiment analysis, we were able to identify subtle emotional indicators in email conversations, providing insights into the underlying views and thoughts of the communicators. Named entity recognition enables the extraction of important entities from emails, such as identifying significant parties, subjects, and entities referenced in exchanges. In addition, topic modelling approaches provide a comprehensive perspective on thematic patterns and trends in email exchanges, offering valuable insights on hot topics and recurring themes. Text classification algorithms make email administration easier by automatically prioritizing and categorizing emails according to pre-established standards. Our research shows that NLP approaches can significantly improve email content analysis, leading to more effective communication, informed decision-making, and competitive advantage for businesses.

### 3. CONCLUATION

This project is the result of years of work aimed at maximizing the potential of Natural Language Processing (NLP) approaches to improve email system content analysis. We have discovered a multitude of insights that have the potential to completely transform corporate communication dynamics and decision-making procedures by exploring the nuances of email exchanges and utilizing cutting-edge NLP algorithms. Sentiment analysis has helped us comprehend the subtle emotional undercurrents in email conversations, bringing to light feelings like urgency, dissatisfaction, and contentment. Organizations can respond and intervene more effectively thanks to this greater knowledge, which improves relationships with partners, clients, and staff.

Furthermore, the use of named entity recognition (NER) has made it possible to recognize and extract important entities—from people and companies to places and goods—from email text. Improved connection mapping, trend analysis, and focused actions based on important stakeholders and interesting themes are made possible by this capacity. By using topic modeling approaches, it has been possible to identify underlying patterns, trends, and emerging subjects in email discussions, giving rise to a comprehensive picture of the thematic landscape. Organizations are better equipped to remain ahead of the curve, spot new possibilities, and take proactive measures to address any problems thanks to this comprehensive understanding.

Additionally, text classification has made email management easier by automating email priority and classification according to preset standards. Organizations are able to sustain high levels of production, respond quickly to pressing situations, and allocate resources efficiently because to this efficiency increase. Our efforts have culminated in the creation of an intuitive online application that allows users to access and utilize NLP-driven content analysis capabilities within their email systems with ease. This application empowers users across organizational hierarchies and disciplines while also improving the user experience and democratizing access to powerful NLP capabilities.

Going future, this initiative will have far-reaching effects that go well beyond email systems. Broader applications in domains including cybersecurity, market research, and customer relationship management are made possible by the approaches, algorithms, and insights discovered. The application of NLP techniques offers a ray of efficiency and creativity as businesses continue to negotiate the challenges of digital communication. This project is proof of the revolutionary power of natural language processing (NLP) in understanding human communication. We open the door to a future in which businesses may use language to drive innovation, facilitate collaboration, and accomplish strategic goals by bridging the gap between technology and human connection.

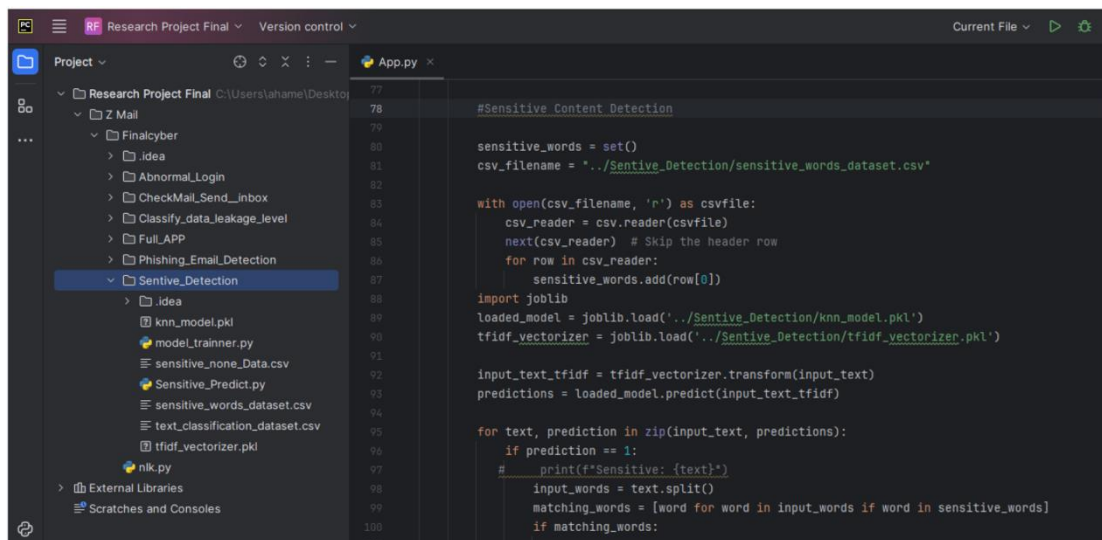
## REFERENCES

- [1] 8 natural language processing (NLP) examples. (n.d.). Tableau. Retrieved April 4, 2024, from <https://www.tableau.com/learn/articles/natural-language-processing-examples>
- [2] Abram, M. D., Mancini, K. T., & Parker, R. D. (2020). Methods to integrate natural language processing into qualitative research. *International Journal of Qualitative Methods*, 19, 160940692098460. <https://doi.org/10.1177/1609406920984608>
- [3] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.
- [4] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research: JMLR*, 3(null), 993–1022. <https://dl.acm.org/doi/10.5555/944919.944937>
- [5] Crowston, K., Allen, E. E., & Heckman, R. (2012). Using natural language processing technology for qualitative data analysis. *International Journal of Social Research Methodology*, 15(6), 523–543. <https://doi.org/10.1080/13645579.2011.625764>
- [6] Guggenbühl, O. (2023, March 2). Knowledge Management with NLP: How to easily process emails with AI. Statworx®; statworx GmbH. <https://www.statworx.com/en/content-hub/blog/knowledge-management-with-nlp-how-to-easily-process-emails-with-ai/>
- [7] Halder, M., Maheshwari, T., & M Suresh, S. R. (2021). A novel approach to control emails notification using NLP. *Procedia Computer Science*, 189, 224–231. <https://doi.org/10.1016/j.procs.2021.05.097>
- [8] Introduction to information retrieval. (n.d.). Stanford.edu. Retrieved April 4, 2024, from <https://nlp.stanford.edu/IR-book/>
- [9] Jurafsky, D. (n.d.). *Speech and Language Processing*. Stanford.edu. Retrieved April 4, 2024, from <https://web.stanford.edu/~jurafsky/slp3/>
- [10] Kumar, B. (2023, July 4). A comprehensive guide to Natural Language Processing (NLP). Tech Blogs by TechAffinity; TechAffinity. <https://techaffinity.com/blog/a-comprehensive-guide-to-natural-language-processing-nlp/>
- [11] Manning, C. D., Surdeanu, M., & Bauer, J. (n.d.). The Stanford CoreNLP natural language processing toolkit. Stanford.edu. Retrieved April 4, 2024, from <https://nlp.stanford.edu/pubs/StanfordCoreNlp2014.pdf>

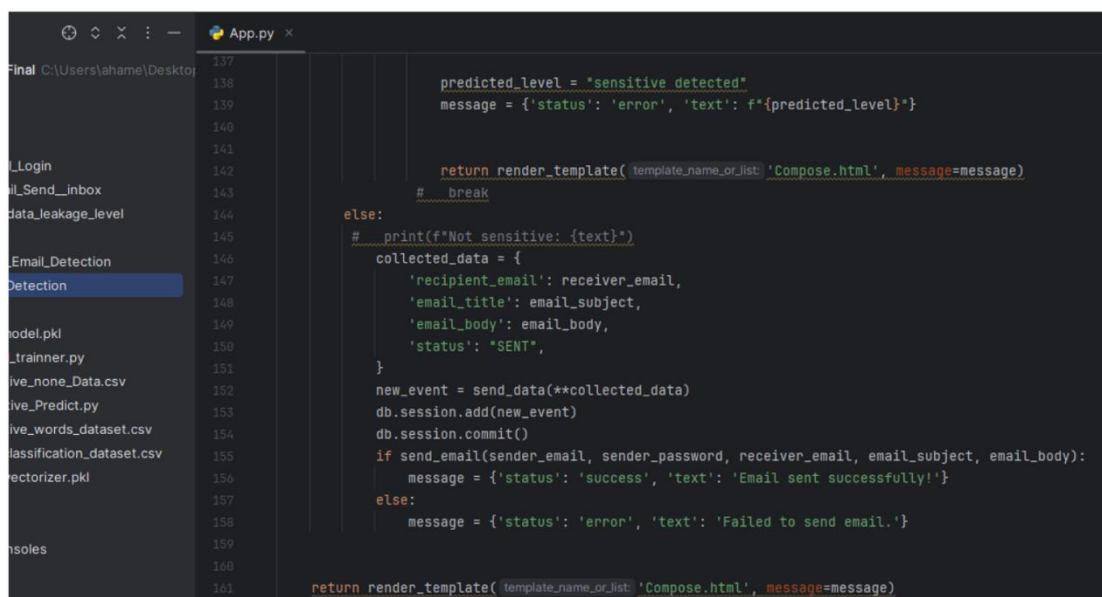
- [12] Manning, C., & Schütze, H. (2021, December 1). Foundations of statistical natural language processing. MIT Press; The MIT Press, Massachusetts Institute of Technology. <https://mitpress.mit.edu/9780262133609/foundations-of-statistical-natural-language-processing/>
- [13] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In arXiv [cs.CL]. <http://arxiv.org/abs/1310.4546>
- [14] Nulty, P. (2017). Semantic/content analysis/natural language processing. In Encyclopedia of Big Data (pp. 1–5). Springer International Publishing.
- [15] Olujimi, P. A., & Ade-Ibijola, A. (2023). NLP techniques for automating responses to customer queries: a systematic review. Discover Artificial Intelligence, 3(1). <https://doi.org/10.1007/s44163-023-00065-5>
- [16] Pennington, J., Socher, R., & Manning, C. D. (n.d.). GloVe: Global vectors for word representation. Stanford.edu. Retrieved April 4, 2024, from <https://nlp.stanford.edu/pubs/glove.pdf>
- [17] Tran, N. (2023, December 17). Practical Natural Language Processing examples. Innovature BPO. <https://innovatureinc.com/practical-natural-language-processing-examples/>
- [18] Vaidya, N. (n.d.). 5 natural language processing techniques for extracting information. Aureusanalytics.com. Retrieved April 4, 2024, from <https://blog.aureusanalytics.com/blog/5-natural-language-processing-techniques-for-extracting-information>
- [19] Yan, H., Rahgozar, A., Sethuram, C., Karunanathan, S., Archibald, D., Bradley, L., Hakimjavadi, R., Helmer-Smith, M., Jolin-Dahel, K., McCutcheon, T., Puncher, J., Rezaiefar, P., Shoppoff, L., & Liddy, C. (2022). Natural language processing to identify digital learning tools in postgraduate family medicine: Protocol for a scoping review. JMIR Research Protocols, 11(5), e34575. <https://doi.org/10.2196/34575>
- [20] (N.d.-a). Researchgate.net. Retrieved April 4, 2024, from [https://www.researchgate.net/publication/353246848\\_Phishing\\_Email\\_Detection\\_Using\\_Natural\\_Language\\_Processing\\_Techniques\\_A\\_Literature\\_Survey](https://www.researchgate.net/publication/353246848_Phishing_Email_Detection_Using_Natural_Language_Processing_Techniques_A_Literature_Survey)
- [21] (N.d.-b). Fastercapital.com. Retrieved April 4, 2024, from <https://fastercapital.com/topics/natural-language-processing-for-content-analysis.html>
- [22] (N.d.-c). Amazon.com. Retrieved April 4, 2024, from <https://aws.amazon.com/what-is/nlp/>

## APPENDICES


```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import joblib
import pandas as pd
```



```
77
78 #Sensitive Content Detection
79
80 sensitive_words = set()
81 csv_filename = '../Sensitive_Detection/sensitive_words_dataset.csv'
82
83 with open(csv_filename, 'r') as csvfile:
84     csv_reader = csv.reader(csvfile)
85     next(csv_reader) # Skip the header row
86     for row in csv_reader:
87         sensitive_words.add(row[0])
88
89 import joblib
90 loaded_model = joblib.load('../Sensitive_Detection/knn_model.pkl')
91 tfidf_vectorizer = joblib.load('../Sensitive_Detection/tfidf_vectorizer.pkl')
92
93 input_text_tfidf = tfidf_vectorizer.transform(input_text)
94 predictions = loaded_model.predict(input_text_tfidf)
95
96 for text, prediction in zip(input_text, predictions):
97     if prediction == 1:
98         # print(f'Sensitive: {text}*)
99         input_words = text.split()
100         matching_words = [word for word in input_words if word in sensitive_words]
101         if matching_words:
```



```
137
138 predicted_level = "sensitive_detected"
139 message = {'status': 'error', 'text': f'{predicted_level}'}
140
141
142 return render_template( template_name_or_list: 'Compose.html', message=message)
143 # break
144
145 else:
146     # print(f'Not sensitive: {text}*)
147     collected_data = {
148         'recipient_email': receiver_email,
149         'email_title': email_subject,
150         'email_body': email_body,
151         'status': "SENT",
152     }
153     new_event = send_data(**collected_data)
154     db.session.add(new_event)
155     db.session.commit()
156     if send_email(sender_email, sender_password, receiver_email, email_subject, email_body):
157         message = {'status': 'success', 'text': 'Email sent successfully!'}
158     else:
159         message = {'status': 'error', 'text': 'Failed to send email.'}
160
161 return render_template( template_name_or_list: 'Compose.html', message=message)
```

 ZMails

Compose Mail

Inbox

Sent Mail

Spam Mail

Blocked Mails

Compose Mail

Recipient Email

ahamedroshan426@gmail.com


BCC:

CC:

Title

Note 2

Hi Fayas. This email regarding HR login credentials. username : Fayas123 password : fa@123

 ZMails

Compose Mail

Inbox

Sent Mail

Spam Mail

Blocked Mails

Compose Mail

Sensitive Information Detected : The Mail Has Been Not Send!!

Recipient Email

BCC:

CC:

Title

> Blocked Mail

> Abnormal Logins

Mails : Sensitive Detected

Recipient Email	Email Title	Email Body	Detected Points
ahamedroshan426@gmail.com	Note	Hi Fayas. This email regarding HR login credentials. username : Fayas123 password : fa@123	username, password
ahamedroshan426@gmail.com	Note 2	Hi Fayas. This email regarding HR login credentials. username : Fayas123 password : fa@123	username, password

```

@app.route('/Send_Jump', methods=['GET'])
def Send_Jump():
    # Use SQLAlchemy's session object to query the send_data table
    with db.session.begin():
        data = db.session.query(send_data).all()

    # Convert each record to a dictionary and append to a list
    sent_messages = []
    for record in data:
        message = {
            'send_id': record.send_id,
            'recipient_email': record.recipient_email,
            'email_title': record.email_title,
            'email_body': record.email_body,
            'date_time': record.date_time,
            'status': record.status
        }
        sent_messages.append(message)

    # Pass the list of messages to the template
    return render_template('Send_Jump.html', data=sent_messages)

```

```

1 usage
def get_browser(user_agent):
    if re.search( pattern: "Edg", user_agent):
        return "Edge"
    elif re.search( pattern: "Safari", user_agent) and not re.search( pattern: "Chrome", user_agent):
        return "Safari"
    elif re.search( pattern: "Firefox", user_agent):
        return "Firefox"
    elif re.search( pattern: "Chrome", user_agent):
        return "Chrome"
    else:
        return "Unknown"

```

```

1 usage
def get_os(user_agent):
    if re.search( pattern: "Android", user_agent):
        return "Android"
    elif re.search( pattern: "iOS", user_agent):
        return "iOS"
    elif re.search( pattern: "Mac", user_agent):
        return "macOS"
    elif re.search( pattern: "Windows", user_agent):
        return "Windows"
    else:
        return "Unknown"

```

```

@app.route('/release_mail/<string:send_id>', methods=['GET', 'POST'])
def release_mail(send_id):
    if request.method == 'POST':
        senddata = db.session.query(send_data).get(send_id)

        if senddata:
            print(senddata.recipient_email)
            print(senddata.email_title)
            print(senddata.email_body)

            # Initialize variables for email parameters
            sender_email = "edlp.g082@gmail.com"
            sender_password = "uaxf bpsk qyla qyww"
            receiver_email = senddata.recipient_email
            email_subject = senddata.email_title
            email_body = senddata.email_body

            # Call the send_email function
            if send_email(sender_email, sender_password, receiver_email, email_subject, email_body):
                message = {'status': 'success', 'text': 'Email sent successfully!'}
            else:
                message = {'status': 'error', 'text': 'Failed to send email.'}

            # Update status in the database
            senddata.status = 'SENT'

            try:
                db.session.commit()
                flash('Mail successfully Sent!', 'success')
            except Exception as e:
                db.session.rollback()
                flash(f'Error in release : {e}', 'error')
        else:
            flash('Data not found!', 'error')

    return render_template('Send_Jump.html')

```