University of East London

Pioneering Futures Since 1898

SCHOOL OF ARCHITECTURE, COMPUTING AND ENGINEERING

Department of Engineering and Computing

Email Classifier to Detect Spams Email Using Machine Learning

[redacted]

[redacted]

A report submitted in part fulfilment of the degree of

BSc (Hons) Cyber Security and Networks

[redacted]

CN6000

Abstract

Purpose: As the world is moving towards the digitalization and the number of internet users are increasing rapidly which leads to the increase of the spam emails. The main purpose of the research is to develop the system which can classify the user emails. The project aim is to build the system with more precise and the accuracy than the past researcher. Furthermore, the email also has become the most convenient way of communication. Due to which the spam emails are increasing either with the good motive like advertising the system or bad motive like getting the access of the user's systems or data, so this project was built to decrease the number of users who are been haunted by the harmful spam emails and making aware the users of those kind of the emails.

Methodology: The project was initially started with the literature review which show the current literature on email classifier that can detect the spam emails, some machine learning research, and the techniques to build the system. I am using Agile approach of software development life cycle. Along with the agile, kanban method was also used due to which the project was completed in the limited time. Furthermore, online survey was conducted to collect the data for the research. Those data were analysed using the quantitative method through the tool like MS Excel which process the data into the graphical forms. Additionally, for the system development platforms like python, Jupyter Notebook, PyCharm and various python libraries were used. Along with that, various machine learning model were built, after comparing the performances of those models with each other, and finally one model was used to make the predictions (i.e., MultinomialNB). The user interface was also built to create the interaction of the users and the systems.

Findings/Result/Outcomes: The outcome of the project mainly depends on the analysis of the data that was collected from the online survey, due to which I was able to find the best techniques to build the systems and makes easy to choose the model to predict the emails. From the analysis of the data, it was found that Naïve Bayes ML model provides the best performance. On the implementation part, it includes the steps of development of system. It also includes the final model of machine learning to predict the emails, before finalizing the model, improvement of model was also tried with the various models and model with the best performance was finalised. The system has its own data to train and test the machine learning models. As a result, the system shows the emails entered by the users are spam or ham.

Acknowledgements

I would like especially to thank my project supervisor Dr Umar Mukhtar Ismail for his expert advice, support and encouragement throughout the project. His knowledge and feedback have a lot for me to accomplish this challenging project.

Furthermore, I would also like to express my appreciation towards the University of East London which helped me to gain the basic and foundation knowledge and skills in the cybersecurity sector. I am also grateful to my family and friends who were supporting me throughout the journey along with thanks to everyone who helped in the completion of the project.

Contents

Tables of Figures:

Chapter 1: **Introduction**

1.1 Introduction:

This is the beginning chapter of the project. This project outlines the aim and objectives of the project that need to be achieved at the end of the project. In this chapter, I will discuss the structure of the research along with its objective and aims.

1.2 Main Theme of the research:

With the increase in the number of internet users, email has become the most significant method of communication between two users. The main theme of the research is to find the best way to detect spam emails. Throughout the research, I would like to know more about the spam email along with their use and impact on the users. Furthermore, I would like to gather more information to improve the accuracy of the automated email classification models with machine learning and with the better outcomes of the model.

1.3 Research Questions:

What are spam emails and are they safe? How can we build the spam emails detection techniques using the machine learning models? Which machine learning model give good performance on spam emails detection?

1.4 Research Aim and Objectives of the Project:

1.4.1 Project Aim:

The aim of the project is to develop a demo of the email classifier which helps to detect whether the email is spam or legitimate.

1.4.2 Project Objectives:

- To research different machine algorithms related to email classification.
- To research the literature and the approaches relating to the classification of email.
- To conduct an online survey with people in the related field through which the feedback can be collected.
- To use quantitative analysis method to find the best conclusion on the project.
- To implement a demo of machine learning techniques and the algorithm to classify the emails.
- To evaluate the outcome of the implemented machine learning techniques measuring their performance and effectiveness.
- To reflect on the results and findings of the project, identifying areas of success and the areas for improvement, which will also suggest future research in this field.

1.5 Research Outlines:

The project is organized with the chapter six. The chapter 2 describes about the academic literature, begins with the recent approach related to the spam emails detection, exploring various concept and considering the potential risk and the drawbacks. It also describes about gaps on the past research. The third chapter focus on the implementation of the project and all the challenges and the limitation of the project. Chapter four focus on the result and outcomes of the research and implementations along with that it also describes about the process used for system development. The chapter five and six focus on the evaluation and the conclusion of the project.

Chapter 2: **Literature Review**

2.1 Introduction:

This chapter is the most important part of the project where I can develop knowledge in the detail of the existing literature related to the email classifier. In this chapter, I will be emphasizing and reviewing the relevant literature and the research about the email classifier, spam emails, and past perspective, I will also be focusing on the machine learning techniques and the approaches to email classification. In this chapter, I will be emphasizing the most recent research and the literature so that this chapter will be the base of my project about machine learning techniques and the implementation of the project. Furthermore, this chapter will also include the gap in the literature about email classification in the detection of spam email.

2.2 What is Email Classification and Spam Email:

2.2.1 Defining Spam Email:

Email has become the most common and convenient way for communication between people with the increase of internet users. The term spam is the data or information that is shared with a large audience without their prior consent, and it may be divided into a broad range of types, including instant messaging (IM) spam, social networking spam, voice over internet protocol (VOIP) spam, online spam, and cell phone-messaging spam among them email spam is the one of the recognized form of the spam in this modern age (Khan et al., 2015). "*Spam email is an unsolicited, unwanted email that was sent indiscriminately, directly or indirectly, by a sender having no current relationship with the recipient.*" (Cormack, 2008, pp 335-455). Furthermore, there are several purposes of sending the bulk email to the unknown recipient. Some of them might be the commercial purposes or some of them might be with bad intentions like getting access to the recipient devices or their data. In some cases, the sender might include malware in the email which will harm the recipient devices, this sort of spam email is more dangerous. However, all the spam is also not dangerous to the recipient. In the business aspect, spam helps them to advertise their business through which they can increase the revenue of their business.

2.2.2 Defining Email Classification:

As the number of people using the Internet expands, email is growing into a widely used and efficient communication tool. Because email is so easy to misuse, email management is becoming a bigger issue for both people and businesses. Email classification is known as the technique and the method of systematically separating email into an already established collection of emails like spam, ham, etc. Mainly, the email classification is performed by reviewing the things mentioned in the email like attachments, the subject of the email, the body of the email, the senders, etc. Cormack (2008) defines a Spam filter or classifier as an automated system or technique for identifying spam and stopping its spread to the users.

The main theme of the email classifier was to manage email traffic systematically and to identify the type of email like unwanted, harmful, etc. Furthermore, email

classification can be done manually or by machine learning (automatically). The manual classification of the email is done by addressing the email that was sent before from the same email address which was identified as spam (Panigrahi, 2012, pp. 506-512). Similarly, while talking about the machine learning email classification the most efficient algorithms which can provide the most efficient results are Neural Network (NN), Support Vector Machine (SVM), and Naive Bayesian (NB) while using the dataset (Youn and McLeod, 2007). Sahami et al. (1998) mentioned that the use of feature classification can help the classification model to improve its performance and the selection of the feature depends on the type of dataset that was used in the development of the algorithm.

2.3 Historical Perspective:

For many years, Email classification has been one of the most researched topics with the increase in the number of use of emails (Mujtaba et al, 2017). Youn and McLeod (2007) have mentioned that among the various techniques and methods, Neural Network (NN), Support Vector Machine (SVM) and Naïve Bayesian (NB) are the most used techniques in the various datasets with good performance outcomes. Additionally, Iqbal and Khan (2022) described that the performance of the machine model could be improved by the use of feature selection techniques. Furthermore, the application of semantic feature space has been proposed as a means of addressing issues related to large feature sets and imprecision in email classification (Yi, Li, Song, 2008, pp. 32-37). Similarly, According to Li and Meng (2015), the main theme of email classification is to separate spam emails from legitimate emails where the classification technique can be divided into the Rule-Based approach and the Content-Based approach. Vyas, Prajapati, Gadhwal (2015) examine supervised machine learning techniques for spam email filtering and find that while SVM and ID3 provide more precision but require more time to build, Naïve Bayes provides faster results and reasonable precision.

2.4 Machine Learning in Email Classification:

IBM (2023) describes Machine learning as a branch of artificial intelligence and computer science, that mimics human learning by using data and algorithms, gradually improving its accuracy. Ahmed et al state that starting with a training dataset, machine learning algorithms are made to generate automated tools for training data, based on user input, this dataset—which may include reviews, examples, feedback, or real-world experiences—helps detect trends and improve decision-making. The main working mechanism of machine learning is first the data will be trained and using the data ML model will be trained, furthermore using the trained model the accuracy of the model will be checked and finally the prediction will be made of the data which can also be seen on the figure below.

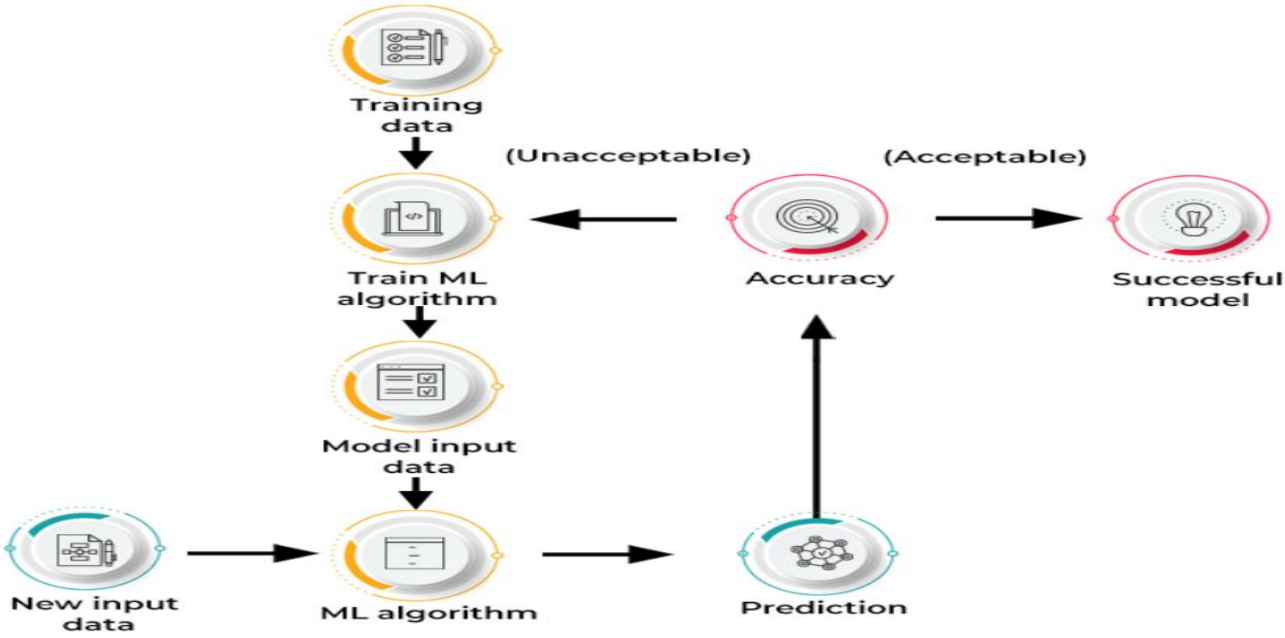

Figure 1. Procedure of Machine Learning (Kanade, 2022).

Furthermore, machine learning has been classified into four parts which are Supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning.

2.4.1 Supervised learning:

Supervised Machine Learning is a mode which was made by using the labelled data where it can predict the outcome of the event based on the training data (Ahmed et al, 2022). The main objective of this type of learning is to map the input variable with the output variable. It has been further divided into two board categories: Classification and Regression. In classification, the algorithms are intended to address classification issues with categorical output variables like yes or no, email filtering etc, while in regression, the algorithms address the regression issue on which the variables have a linear relationship like it is used to analyse the continuous output variables (Kanade, 2022).

Some of the supervised learning models are:

- **Decision Tree Algorithm:**
  The decision tree algorithm is one of the most used supervised machine learning models because it is easy to use and easily describes and visualization (Mansoor, Jayasinghe, Muslam, 2021). It can easily predict numeric data and classify data. According to Verma and Gautam (2019), A decision tree is a hierarchical supervised learning model that uses fewer steps and recursive splits to identify local areas. With interior nodes known as decision nodes and leaf nodes that indicate class values, it resembles a binary tree. A test function with discrete results labelling branches is implemented by each decision node. Based on the results of the tests, the model selects one branch once input is given.

- **Linear Regression:**

Based on the research, the linear regression model is the simple regression supervised model which is used to predict the statistical value by the understanding of the linear relationship between different values (IBM, 2023).

- **Support Vector Machine Algorithm:**
  SVM is a supervised learning classifier that uses labelled examples for training and produces a hyperplane for classifying new data, separated by decision planes for different class memberships (Verma and Gautam, 2019).
- **Naïve Bayes:**
  It is one of the supervised models based on the Bayes rule which uses the probabilistic learning method that classifies features independently of each other's value (Gomes et al, 2017).

- **Logistics Regression:**
  It is the statistical-based supervised model which gives the prediction of the probability of the given data class using regression models (Adewale and Yamazaki, 2023).
- **Random Forest:**
  According to Adewale and Yamazaki (2023), the random forest model is one of the supervised machine learning models where a model is made up of several decision trees with a component of nodes and branches. Based on a question's response, each node chooses which branch to go to next. The destination node is the answer to the preceding question, and the initial node is referred to as the root. An ensemble-leaning model selects the best response from a range of possibilities.
- **K-Nearest Neighbour:**
  According to Kumar and Sonowal (2020), K-nearest neighbours are a supervised classification technique that predicts the categorization of fresh sample points by using data points and vectors divided into many classes. This model didn't make decisions by itself just memorize the process instead of learning the process.

2.4.2 Unsupervised Machine Learning:

Unsupervised machine learning is the machine learning algorithm that can analyse and cluster unlabelled datasets to identify hidden patterns or data grouping without the needs of humans (IBM, 2023). Ahmed et al (2022) mentioned that unsupervised machine learning models are used in those cases when labelled data are lacking and investigate how programmes can infer a feature from unlabelled data to explain a hidden structure. The unsupervised machine has been further divided into 2 methods i.e., clustering and association (Kanade, 2022).

Some of the unsupervised models are:

- **K-Means Clustering model:**
  K-means is a straightforward unsupervised learning technique that addresses the clustering problem by categorising a data set into a predefined number of clusters (Mahesh, 2020, pp). The goal of K-means clustering is to allocate data points to clusters by minimising the sum of squared distances

between data points and their assigned cluster. This technique divides a dataset into discrete, non-overlapping groups or clusters. With a randomly chosen centroid for each cluster, the K clustering data mining process begins with the first randomly chosen group. The optimised position is produced by doing iterative calculations from this centroid (Mansoor, Jayasinghe, Muslam, 2021).

2.4.3 Semi-Supervised Machine Learning:

Semi-supervised machine learning is the combination of both techniques which means the machine learning can be trained with both the labelled and unlabelled data during the testing phase and analysis is done using both techniques (Mansoor, Jayasinghe, Muslam, 2021). A semi-supervised learning model aims to provide superior prediction outcomes compared to using labelled data alone from the model and it has been further categorised into Clustering and Classification (Sarker, 2021).

2.4.4 Reinforcement Machine Learning:

Reinforcement machine learning is also known as an environment-driven technique. It is a kind of machine learning algorithm that allows software agents and computers to automatically assess the best behaviour in a given context or environment to increase its efficiency (Sarker, 2021). Throughout the research, It was found that this tool is not appropriate for tackling simple or fundamental problems, but it is helpful in training AI models and improving automation and operational efficiency in complex systems like robots, autonomous driving, manufacturing, and supply chain logistics. Furthermore, it has been divided into model-based and model-free techniques where model-based reinforcement learning infers optimal behaviour from an environment model, including the subsequent state and immediate reward. Model-free techniques, such as SARSA, Q-learning, Deep Q Network, and Monte Carlo Control, do not make use of the MDP-related reward function and transition probability distribution (Sarker, 2021).
Some of the reinforcements machine learning are:

- **Deep Q-learning:**
  In deep Q-learning, an initial state is fed into a neural network, and the network outputs the Q-value of every action that might be taken. When dealing with more complicated states and actions, it can be utilised as a function approximator, but it performs well in simpler scenarios (Kaelbling, Littman, Moore, 1996).
- **Q-learning:**
  The goal of Q-learning, a model-free reinforcement learning method, is to identify the characteristics of behaviours that influence an agent's decisions in given situations (Kaelbling, Littman, Moore, 1996). The 'Q' in Q-learning stands for quality since the algorithm determines the maximum predicted rewards for a particular behaviour in a given state. Q-learning is a model-free approach that can handle stochastic transitions and rewards without modifications (Saeker,2021).

2.5 Feature Extraction and Selection Methods:

Feature extraction and selection methods are the common steps of feature engineering. According to Patel (2021), Feature engineering points toward the process of modifying data sets to optimise the training of machine learning models, leading to increased accuracy and performance and the main goal of feature engineering is to improve the performance of machine learning models. There are several steps in feature engineering like text data processing, normalization, vector feature, feature selection, feature extraction, etc (Patel, 2021). By carrying out the feature engineering, it helps to understand the pattern of the dataset.

- **Feature Extraction:**
  It the one of the most important parts while talking about email classification which helps to improve the performance of the models. It helps to decrease the complexity of data in the datasets. It converts the original feature of data into smaller, more compact space and replaces the original features with a smaller representative set without deleting them (Zareapoor and Seeja, 2015). Some of the common techniques of feature extraction are:
  - **Principal Component Analysis:**
    It creates new uncorrelated variables that are combinations of the original variables, reducing the dimensionality of the data by reshaping the original attribute space into a smaller space (Zareapoor and Seeja, 2015). It is a statistical technique that focuses on finding patterns and connections among the variables in a dataset.
  - **Bag of Words (BOW):**
    It is a text-based method for classifying and extracting characteristics in Natural Language Processing (NLP) that are based on how often they are used. This method gives machine learning algorithms a vector of word counts (Jason Brownlee, 2019). It is a text presentation that lists the words that appear in a dataset.

- **Feature Selections:**
  The feature selection method eliminates deleted features from calculations and chooses a subset of the original features for classifier testing and training (Zareapoor and Seeja, 2015). Some of the common techniques are:

  - **Chi-Square:**
    The Chi-Square ($\chi 2$) approach is a popular feature selection technique that assesses each feature separately by calculating chi-square statistics and examining term-class dependencies; higher scores yield more information (Zareapoor and Seeja, 2015).
  - **Information Gain:**
    It is a feature selection method that keeps the best attributes and reduces feature size by calculating and rating attribute values to establish a threshold (Zareapoor and Seeja, 2015). The selection of features is done by the score.

2.6 Gaps in the Current Literature:

Throughout the research, it was found that the specific method for the solution for the management of spam email was not addressed. Furthermore, there is a need for further research in the field of proper management of spam email classification. The researchers had mentioned that the email classification model's performances can be improved by feature engineering, but the proper way of feature engineering method was not discussed. Moreover, spam detection was one of the most researched topics to prevent attacks by spam email attack. Additionally, the current solutions of spam management are not considered perfect, therefore, more efficient way of models or techniques need to be implemented which will have a high accuracy in the detection of spam emails. Additionally, the researchers were struggling to find balanced email data and the limited availability of datasets other than English languages. More comprehensive approaches are required to handle the complex functional range of email and offer improved defence against constantly changing spam email attacks.

Chapter 3: **Project Methodology**

3.1 Introduction:

In this chapter, I will discuss the implementing methods of my project. Throughout this chapter, I am going to explain how the research approach and the implementation are made for the project. At last, I will be talking about the challenges and limitations of the research and the ethical considerations of the project.

3.2 Approach:

The project will be implemented by using the agile approach of system development life cycle methodology. Along with the agile methodology, the Kanban method will be used which will help me to finish the system development in a limited period, in a smooth and faster way by splitting the task into the sub-task. Furthermore, for the development of the system, I will be using Python programming languages, and, in Python, I will use libraries like pandas, NumPy, Matplotlib, NLTK, and so on.

The online survey will used for the research in the quantitative way which is one of the most common in the software development field.

3.3 Implementation:

While talking about the development of email classification, one of the first steps is to choose the platform and the machine learning algorithms. I have used Python for both platform and machine learning as Python has various libraries which help me with effective data analysis and machine learning models.

Furthermore, During the development of the system, I used various tools like Jupyter Notebook, PyCharm, etc. where Jupyter Notebook helped me to run the machine learning model, PyCharm helped me to design the user-friendly interface from which the users will be able to classify their entered email is spam or not. The dataset which was used for training and testing the model was taken from the Kaggle website which also includes promotion and chained content. The machine learning model, I have used to design a system is a supervised machine learning model. Throughout the research, it was found that Naïve Bayes is the best machine learning model to create a classification and finally, after testing the performance of other machine learning models with the data set, Multinomial Naïve Bayes was finalised to build the model.

3.3.1 SDLC

The methodology that was used for the completion of the project is agile. It is one of the approaches of the software development life cycle (SDLC) which emphasises frequent interaction and improvement while segmenting the tasks into manageable parts (Atlassian, 2019). One of the main reasons for using the agile methodology is the priority of this project is to develop the working demo of the system which will help to predict whether email is spam or ham. The project has a tight schedule and a fixed period of time on which the project needs to be finished with the higher quality of the system's performance.

With the use of agile methodology in the project, the project could be started with the initial idea and any problem that arises in the problem can be mitigated at any stage without affecting the project's progress. While the use of the traditional approaches of SDLC like a waterfall, doesn't allow to change anything between starting and end stages. Furthermore, Agile supports the quick development of the system with high-performance efficiency and is suitable for the small team.

3.3.2 Kanban Method:

Kanban is one of the methods of agile methodology which was developed in Japan in the late 1940s and was started to use in software development in the early 2000s (Martins, 2022). It is defined as the visual workflow management method which specifically focuses on the continuous improvement of the project and limiting work in progress. The use of the Kanban method on the project, will help me with the continuous improvement of the system and divide the work into a small part which was the main reason for choosing it.

3.4 Research Approach:

3.4.1 Literature Review:

While talking about the literature review, I must use verifiable and credible sources from where all the information related to my project can be collected. I have been using online platforms like Google Scholar and IEEE Xplore. It is also one of the important parts for the completion of the project along with the efficient performance of the system because it will help and guide me to find the ways and methods to achieve the objective of the project and to implement the system. Additionally, The E-library and the physical library of the university (University of East London) were also used to gain the knowledge and the data needed for the project. Furthermore, all the references and the citations of all the information collected from the sources are also provided in the project in the Harvard referencing form.

3.4.2 Quantitative approach using online survey:

Online surveys are a common way of gathering data in which a target sample is provided with a set of survey questions and asked to reply to the appropriate ones via the Internet (Bhat, 2018). It is also one of the popular methods used by the researcher to collect data. When conducting an online survey, a series of questions is usually created and submitted by respondents using a web-based platform where the questionnaires might include rating scales, open-ended inquiries, multiple-choice questions, and more.

In this project, I will create a survey starting with the generic questions and moving into the related to the projects. The survey will be shared on the university team's platform from where I will be able to collect the data and only the final year student will be able to participate in the online surveys. Finally, the quantitative analysis approach will be used to conclude the results of the survey.

3.5 Challenges and Limitations:

3.5.1 Implementation:

During the implementation of the system for the detection of spam email, one of the main challenges was data collection and time for the completion of the project. The dataset that was used on the system was in the proper format and that data was also not balanced which means a number of spam emails and the ham emails was not equal. As the dataset was imbalanced, another challenge that I faced was building the machine learning model with 100% accuracy (i.e., the model was developed with 100% precision, but the accuracy was not 100%). Additionally, one of the other challenges was joining the machine learning model and the user interface with the pipeline which means the data used on the system was not possible to use in it online. So, the dataset needed to be downloaded on the system due to which the system was getting slow, and I was facing a problem in building the environment that was needed to connect the machine learning model and the user interface site.

3.5.2 Research Approach:

As we know quantitative analysis using the online survey is the most common and precisely used research method, but there are several limitations. The quality of data may be low as the respondent may be in a hurry or might be tired of attending different surveys and various

people may be surveyed under various circumstances, so data obtained from these people using the same questionnaire is not accurate. As the data was just collected from the final year students of computing some of the participants may not be comfortable with the questionnaires which may lead to inaccurate responses and may not be able to find the best conclusion on the project. Researchers' comprehension of non-response patterns may be limited by the lack of specific information provided by online surveys regarding the reasons behind certain people's decision to decline participation.

3.6 Ethical Considerations:

When the project has been started, ethical considerations are one of the most important steps that need to be considered either in the implementation approach or research approach. While the quantitative online survey was used in the research approach, none of the participants were asked to provide their details which means all the data were collected anonymously. The survey was created by using the Microsoft Forum and the responses were stored secretly. These data can only be accessed by the user who has created the survey using the form and the supervisor of the users.

Furthermore, In the implementation of the system in the project, the dataset that was used in the development of the system from the public open source. Additionally, the system needs interaction with the user through which the user can check whether their email is either spam or not. In the meantime, users were not asked for their details, the system will just allow the user to check their email, but it will not store any activity of the users. So, the ethical considerations were considered very well in every stage of the project.

Chapter 4: **Results/Findings/Outcomes**

4.1 Implementation Result:

4.1.1 Introduction:

In this section of the chapter, I will display the outcomes of implementing the project. I will also discuss the final machine learning model that can detect the spam email entered by the users. This section consists of all the processes I used to develop the system. As a part of the system, it has its data which was used to develop the system. Along with the machine learning model, the system consists of the user interface which helps to interact the system with the users.

4.1.2 Initial Specification:

One of the main things that must be planned to develop the system is choosing the platform along with the programming languages. When the development of the system was planned, the first step of the system was to implement platforms like Jupyter Notebook, PyCharm etc. to run the code. Along with the platform, another was to collect the data which can be used to develop the machine learning model. Furthermore, the pipeline was also developed to connect the machine learning model and the user interface website.

4.1.3 Dataset:

The dataset which was used for the building of the system was collected from the Kaggle website. Furthermore, the same data were split into train and test data to train and test the machine learning model. The data was loaded on the Jupyter Notebook to develop the system which can also be seen in the figure below.

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN
5	spam	FreeMsg Hey there darling it's been 3 week's n...	NaN	NaN	NaN
6	ham	Even my brother is not like to speak with me. ...	NaN	NaN	NaN
7	ham	As per your request 'Melle Melle (Oru Minnamin...	NaN	NaN	NaN
8	spam	WINNER!! As a valued network customer you have...	NaN	NaN	NaN
9	spam	Had your mobile 11 months or more? U R entitle...	NaN	NaN	NaN
10	ham	I'm gonna be home soon and i don't want to tal...	NaN	NaN	NaN
11	spam	SIX chances to win CASH! From 100 to 20,000 po...	NaN	NaN	NaN
12	spam	URGENT! You have won a 1 week FREE membership ...	NaN	NaN	NaN
13	ham	I've been searching for the right words to tha...	NaN	NaN	NaN
14	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	NaN	NaN	NaN

Figure 2. Sample of data used on the system.

4.1.4 Data Cleaning:

After loading the data in the source code, another step was to check the data and clean the unwanted data. As for the dataset, there were some columns which do not have the data which can be viewed in the figure shown below. Cleaning the data helps to make the machine learning model work properly as well and also makes it easier to analyze the dataset. In this process, all the duplicate data were removed, and the columns were renamed as the target and the text. Additionally, the ham and spam emails were labelled as 0 and 1 respectively.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

Figure 3. Sample of the dataset before cleaning them.

	target	text
3548	ham	I got like $ <#> , I can get some more l...
674	ham	I've got <#> , any way I could pick up?
709	spam	4mths half price Orange line rental & latest c...
4454	ham	Storming msg: Wen u lift d phne, u say \HELLO\...
4301	ham	Aiyar hard 2 type. U later free then tell me t...
4479	ham	Y cant u try new invention to fly..i'm not jok...
1288	ham	Happy new year to u too!
811	ham	So there's a ring that comes with the guys cos...
3813	ham	Can. Dunno wat to get 4 her...
715	ham	When i have stuff to sell i.ll tell you

Figure 4. Sample of the dataset after cleaning and labelling

4.1.5 Exploratory data analysis:

Exploratory data analysis is also another important part of the process after the collection and the data cleaning process. It helps to get to know the condition of data in the dataset. Similarly, It also helps in choosing the machine learning model to give the best result. As per the analysis of the dataset, It was found that the data on the dataset was not balanced which means ham emails were larger in number than spam emails which can also be seen in the pie chart below.

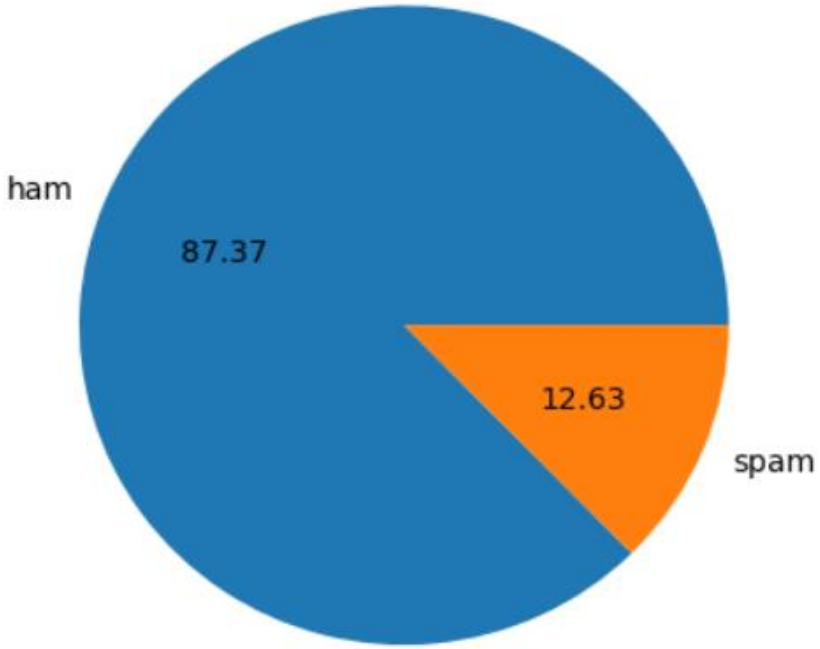

Figure 5. Pie-Chart shows the percentage of ham and spam email on the dataset.

Furthermore, in this process, I have also checked the presence of the number of words, number of sentences, and number of characters on the ham emails and the spam emails to know the relationship between them on both emails.

	target	text	num_characters	num_words	num_sentences
0	0	Go until jurong point, crazy.. Available only ...	111	24	2
1	0	Ok lar... Joking wif u oni...	29	8	2
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2
3	0	U dun say so early hor... U c already then say...	49	13	1
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1
5	1	FreeMsg Hey there darling it's been 3 week's n...	148	39	4
6	0	Even my brother is not like to speak with me. ...	77	18	2
7	0	As per your request 'Melle Melle (Oru Minnamin...	160	31	2
8	1	WINNER!! As a valued network customer you have...	158	32	5
9	1	Had your mobile 11 months or more? U R entitle...	154	31	3

Figure 6. Sample of data showing the number of characters, words, and sentences on each email

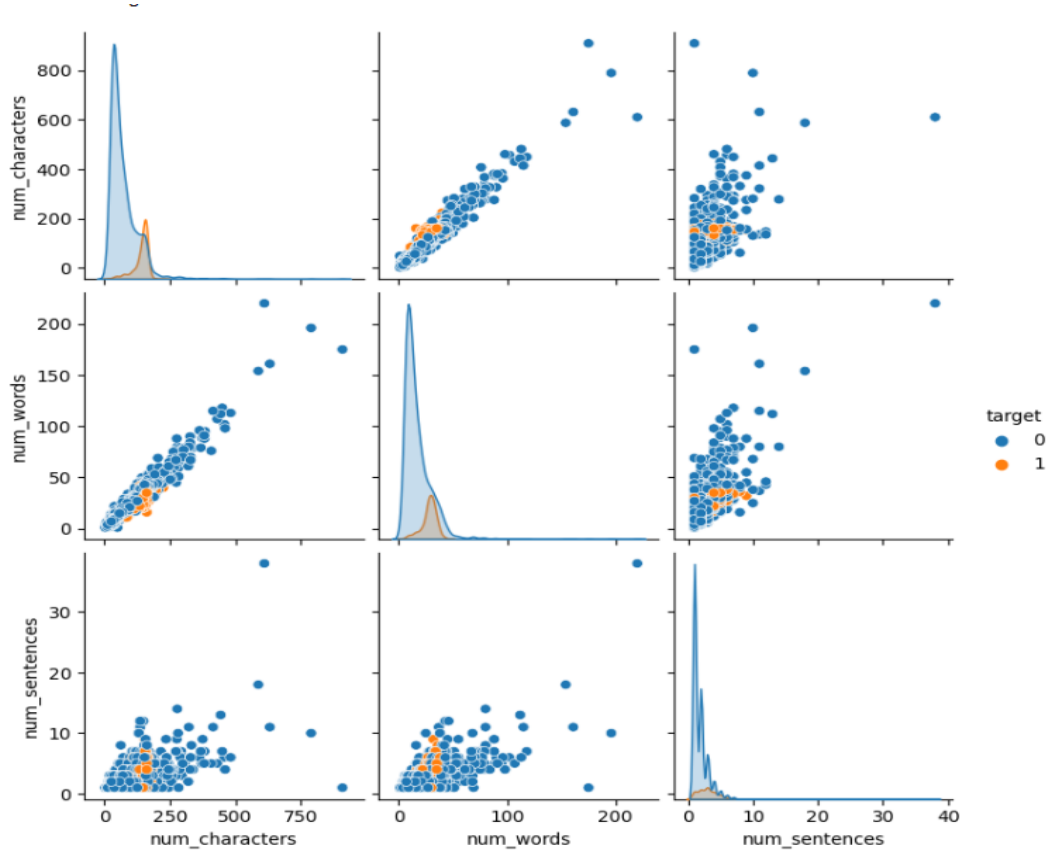

Figure 7. Pair plot showing the relationship between a number of words, characters, and sentences in each ham and spam email.

Figure 8. Heatmap Correlation of the number of characters, words, and sentences

4.1.6 Data preprocessing:

In this process, I am going to pre-process all the data of the dataset which is another step of the system development. This step includes the conversion of all the data into lowercase,

tokenizing the words, removing the special characters, stopping word and punctuation and changing the word to their original form (i.e., converting all the words into their original verbs).

	target	text	num_characters	num_words	num_sentences	transformed_text
0	0	Go until jurong point, crazy.. Available only ...	111	24	2	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	29	8	2	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2	free entri 2 wkli comp win fa cup final tkt 21...
3	0	U dun say so early hor... U c already then say...	49	13	1	u dun say earli hor u c alreadi say
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1	nah think goe usf live around though
5	1	FreeMsg Hey there darling it's been 3 week's n...	148	39	4	freemsg hey darl 3 week word back like fun sti...
6	0	Even my brother is not like to speak with me. ...	77	18	2	even brother like speak treat like aid patent
7	0	As per your request 'Melle Melle (Oru Minnamin...	160	31	2	per request mell oru minnaminungint nurungu ve...
8	1	WINNER!! As a valued network customer you have...	158	32	5	winner valu network custom select receivea pri...
9	1	Had your mobile 11 months or more? U R entitle...	154	31	3	mobil 11 month u r entitl updat latest colour ...

Figure 9. Sample of the dataset after pre-processing them.

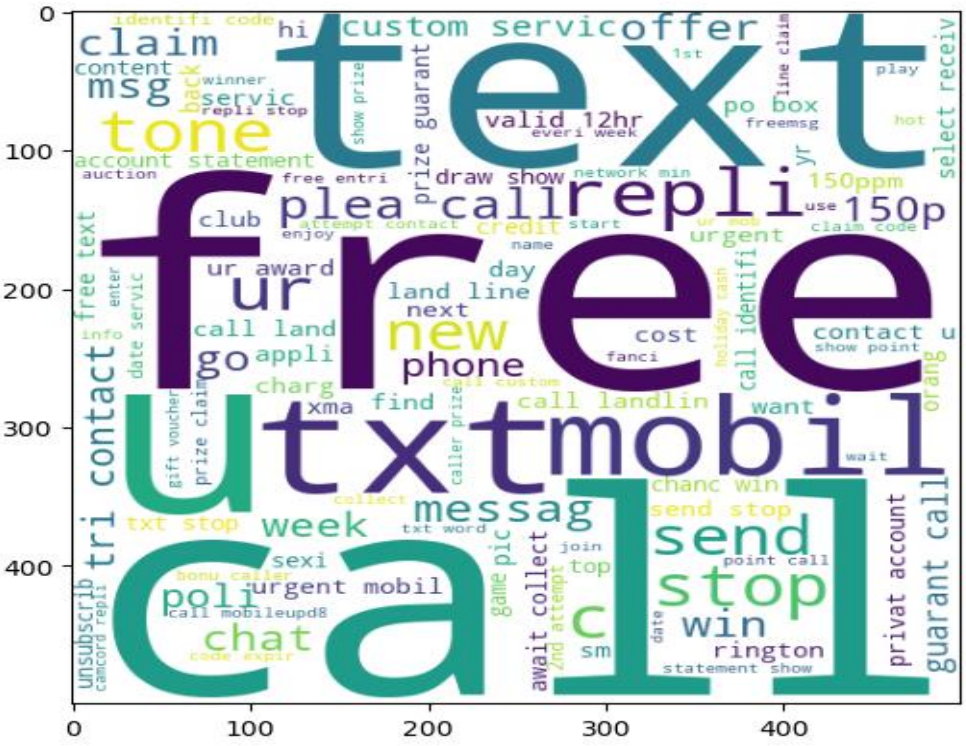

Figure 10. Word cloud showing the content of Spam emails.

Figure 11. Word Cloud showing the content of ham emails.

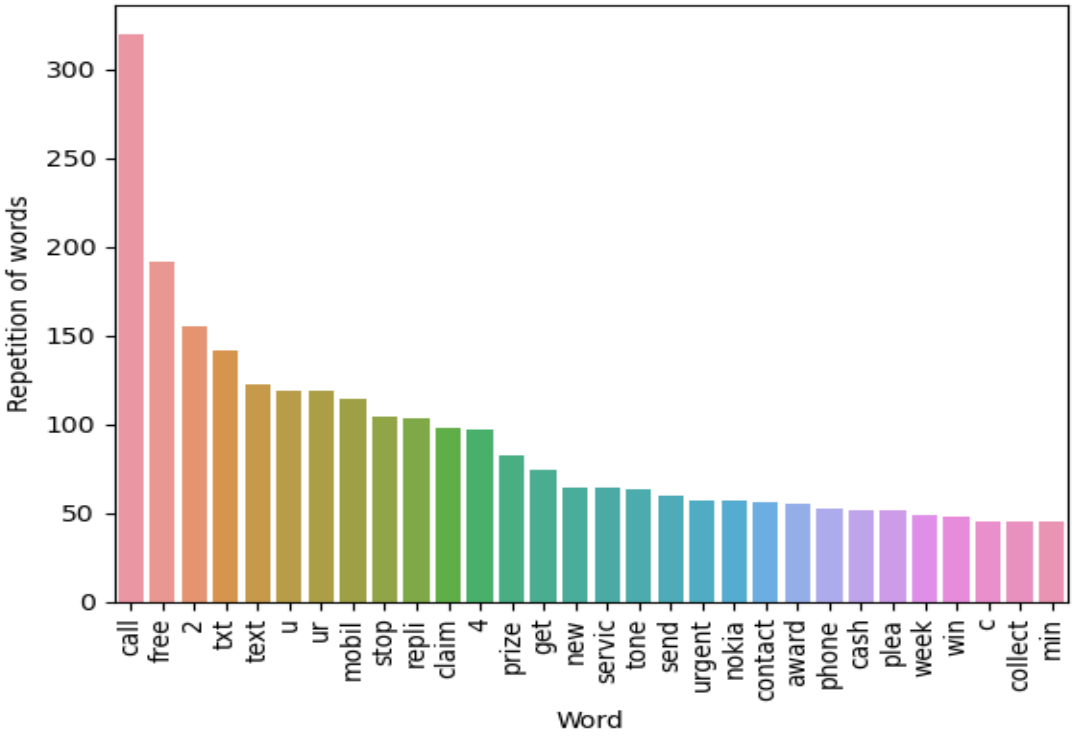

Figure 12. Top 30 word mostly repeated words in spam emails

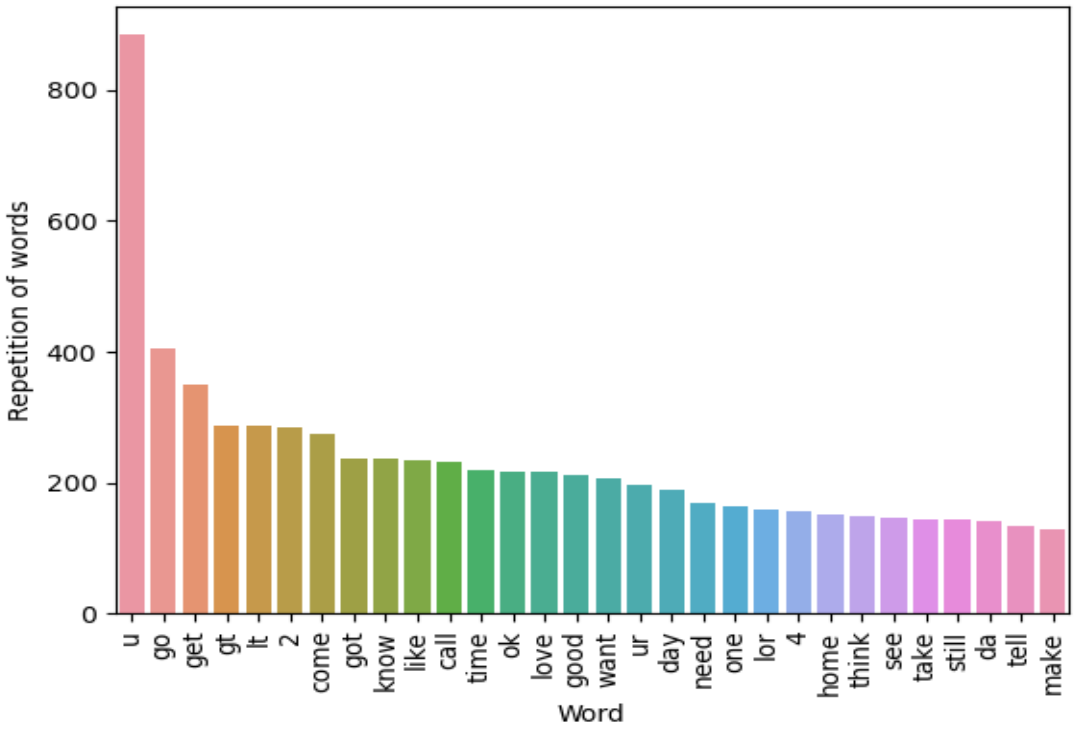

Figure 13. Mostly 30 repeated words in ham emails

4.1.7 Model Building:

This is one of the important steps in which I have built the model which can predict whether the email is spam or not. This step consists of finalizing the machine learning model, Improvement of the model, etc. First, while building the model the main things that need to be done is vectorizing the data (i.e., changing the text into the number) as I am going to build the model using the Naive Bayes and it just accepts the input and output only into the number. For vectoring the text, first I tried with the count vector to build the machine learning model using the Naive Bayes and got the accuracy and precision as shown below.

```
0.8800773694390716
[[792 104]
 [ 20 118]]
0.5315315315315315
```

GaussianNB

```
0.965183752417795
[[872  24]
 [ 12 126]]
0.84
```

MultinomialNB

```
0.9709864603481625
[[893   3]
 [ 27 111]]
0.9736842105263158
```

BernoulliNB

Figure 14. Accuracy score, confusion matrix and precision score of Gaussian, Multinomial and Bernoulli Naïve Bayes using count vectorizer.

Furthermore, the text was vectorised using Tfidf to check whether the performance of Naive Bayes can be improved. It was found that the multinomial naïve Bayes model was giving better performance by using the Tfidf vectorizer than the count vectorizer. Among the three of them, multinomial was performing better. So, I decided to go forward with the Multinominal Naïve Bayes using the Tfidf vectorizer.

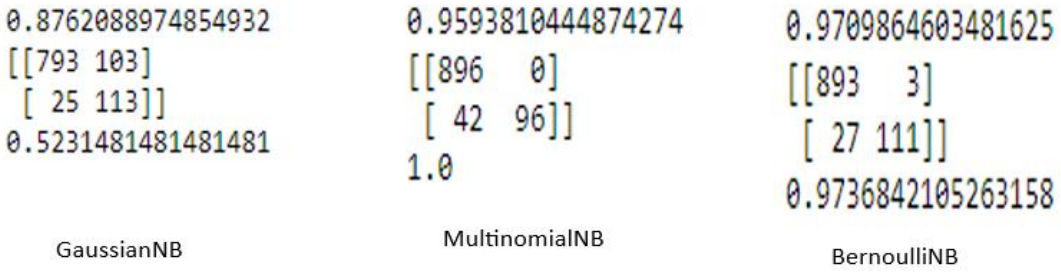

Figure 15. Accuracy score, confusion matrix and precision score of Gaussian, Multinomial and Bernoulli Naïve Bayes using Tfidf Vectorizer.

Although the Naive Bayes model will provide good performance using the dataset, I have also tried other machine learning models like Random Forest, Support Vector, AdaBoost, logistic regression etc. to check how they will be their performance in comparison with the Multinominal Naive Bayes model.

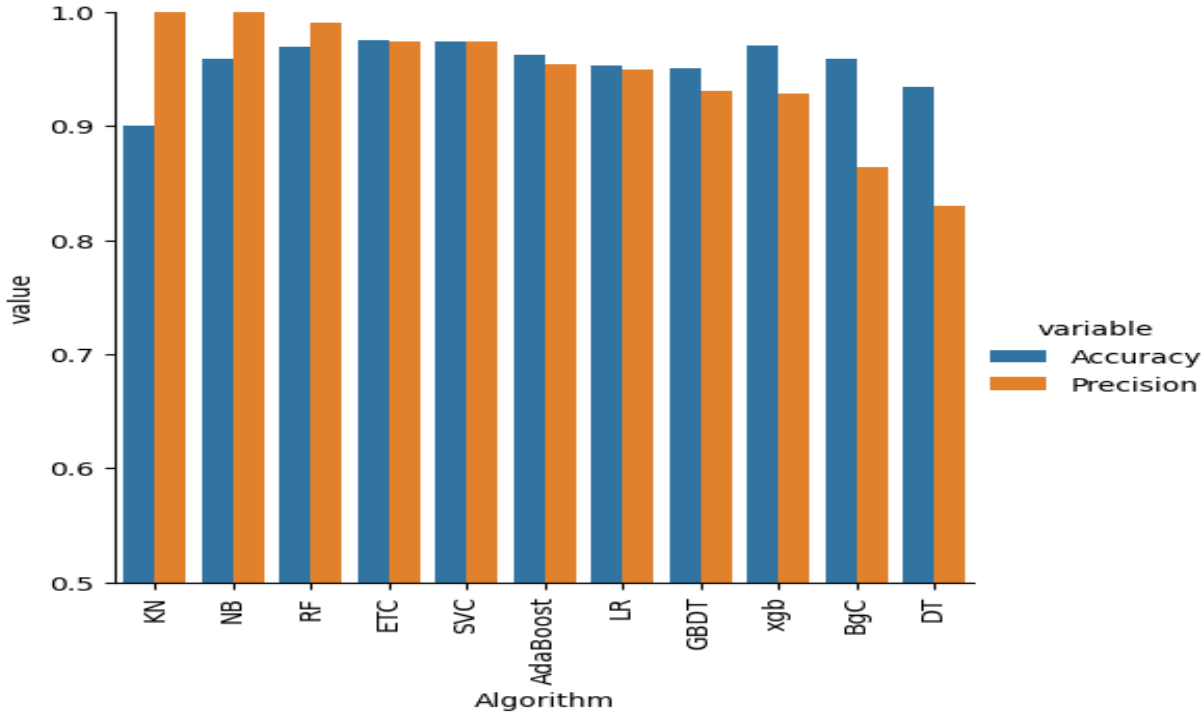

Figure 16. Original Accuracy and precision of Support vector, KNeighbors, MultinominalNB, DecissionTree, Logistic Regression, AdaBoost, Bagging Classifier etc.

Additionally, from the figure above, Multinomial Naïve Bayes, Random forest, ExtraTree Classifier and Support Vector machine learning model are giving good performance. Among four of them, Multinomial naïve Bayes did not give any false positive and the precision score was higher than the other model. The dataset used in this project is unbalanced. So, the multinomial naïve bayes model looks perfect to build the model. However, before finalizing the model I am going to try to improve the performance of the model. While trying to improve the model, I have used the max feature of 2000 in the Tfidf, min-max scaler method and also tried by appending the Num-character column in X which was extracted from exploratory data analysis. After comparing the accuracy and precision which was obtained from applying these improving methods, the Multinomial Naïve Bayes model was improved

in comparison to others using the Max feature selection method of Tfidf. So, the Multinomial Naïve Bayes model was finalized to build the machine learning model with Tfidf max feature selection methods.

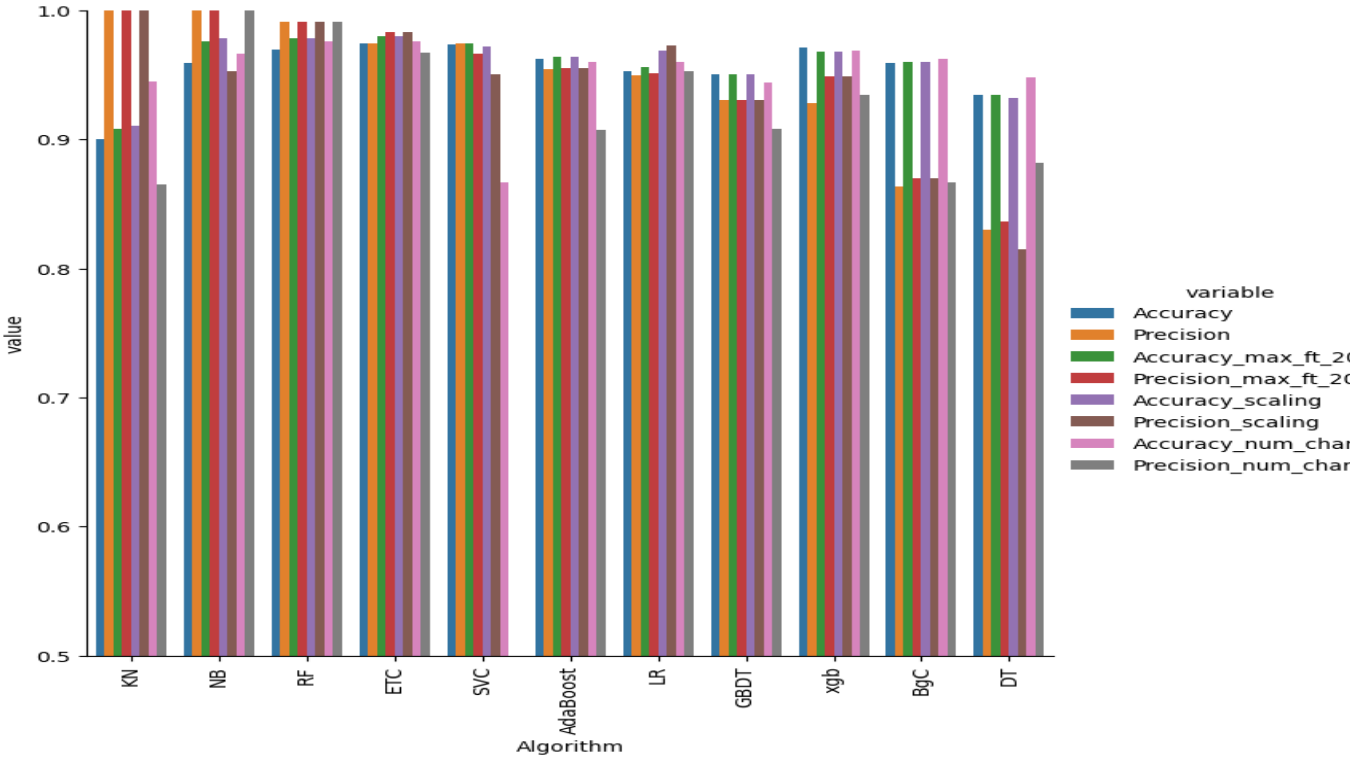

Figure 17. Accuracy and precision score after applying improving model with Original Accuracy and precision score of Support vector, KNeighbors, MultinominalNB, DecissionTree, Logistic Regression, AdaBoost, Bagging Classifier etc.

4.1.8 Dashboard:

This is the final step of the system implementation. In this step, I have used the Python library (i.e., Streamlit) to create the dashboard by which users can interact with the system and check whether their email entered is spam or not. Furthermore, in order to display the result of the email entered as spam or ham, the entered emails need to be pre-processed, vectorized and made the prediction. The dashboard and the trained machine-learning model were connected by using the pipeline (Pickle).

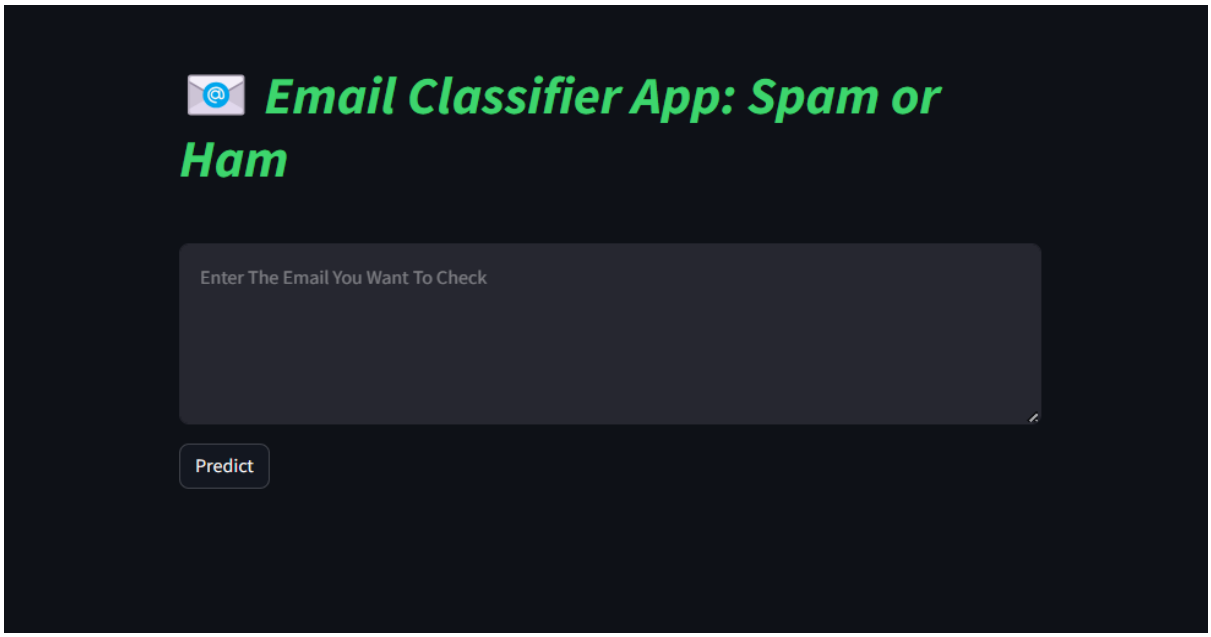

Figure 18. User Interface dashboard created using Streamlit.

Figure 19. Dashboard Showing the email entered is spam and is not safe to use.

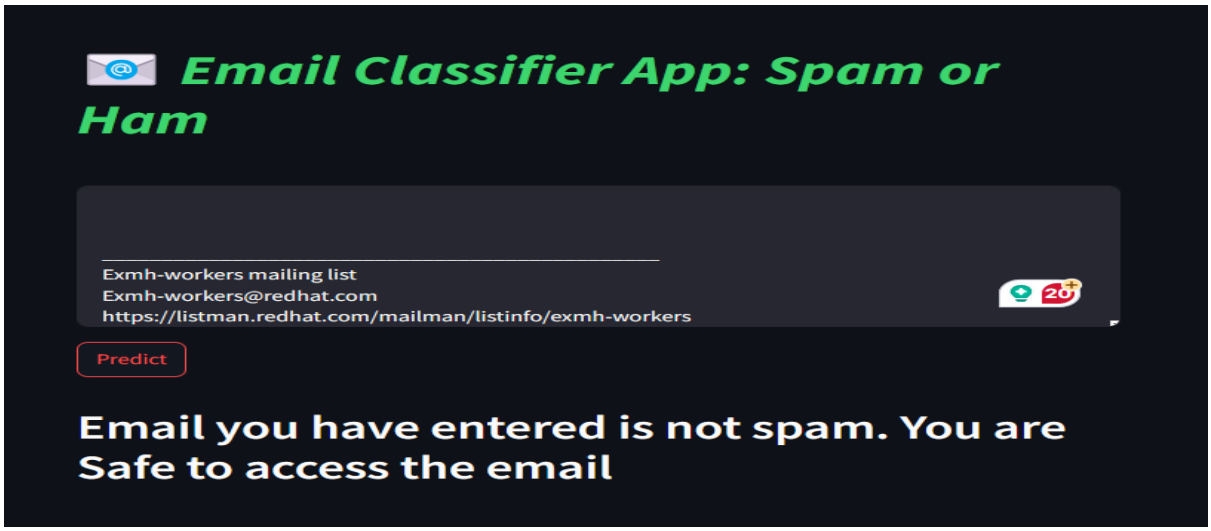

Figure 20. Dashboard showing the email entered is ham and safe to use.

4.2 Online Survey Result:

In this project, the online survey was conducted from where all the data was collected that was needed to do the quantitative analysis for the project. One of the main purposes of conducting the survey is to collect the data through which the prediction can be made to build the system, the importance of the project and to test the system. As mentioned in the methodology chapter of the project, the online survey was distributed to the university final year students of the bachelor's degree and it was posted on the online platform of the university i.e., Microsoft Teams.

The questionnaire on the survey started with the demographic question moving forward with the related to spam email detection questions. The questionnaires were created using Microsoft Forms and the link was shared within the university. Once the survey was shared about 37 responses were collected, and all the responses were exported into the MS Excel sheet which was one of the benefits of Microsoft Form. Finally, the analysis was performed to get the conclusion of the survey.

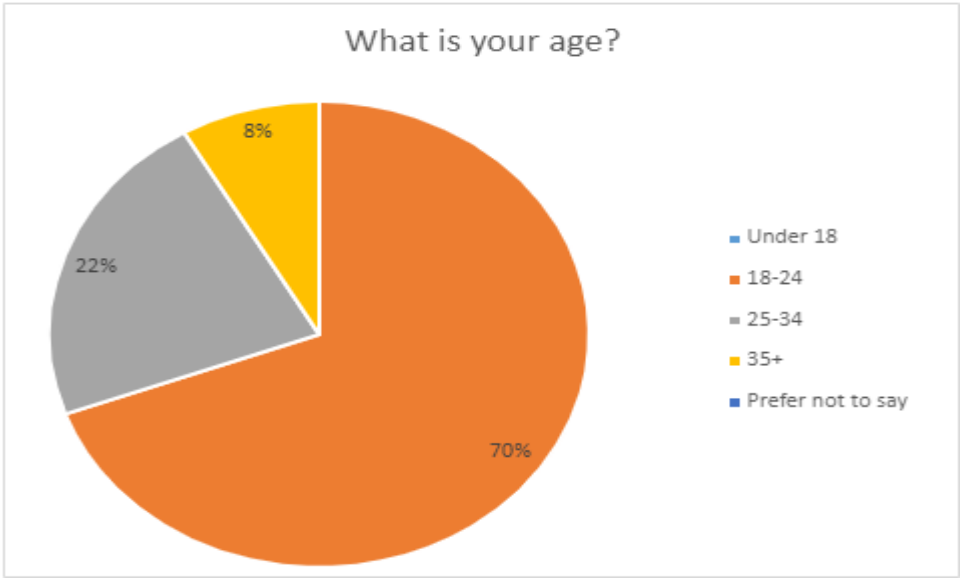

Figure 21. Pie-chart showing the age of the participant.

From the picture above, the maximum number of participants were the age of 18-27 which is 70%. Furthermore, 22% of participants were the age of 25-34 and 8% were of the age of 35+. None of the participants were the age of under 18 and all participants were comfortable to share their age.

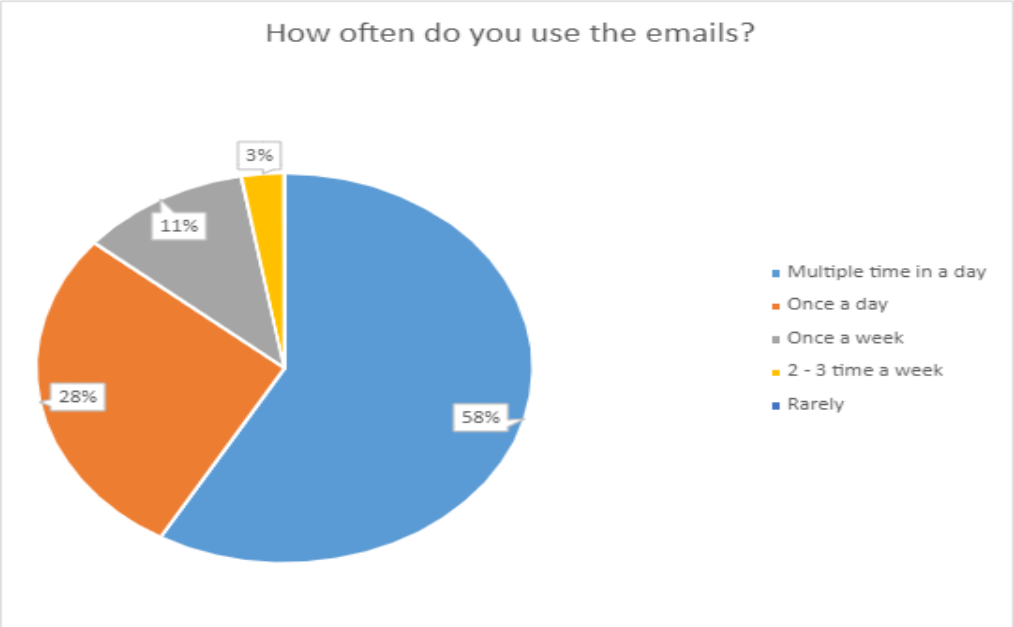

Figure 22. Pie-chart showing frequency of using the emails by the participants.

Among the participants, 58% of the participant use the email multiple times in a day however there were no nay participant who did not use the email (i.e., who use the email rarely). Additionally, from the above pie chart, we can also see that 28%, 11% and 3% of the participant use the email once a day, once a week and 2-3 time a week respectively.

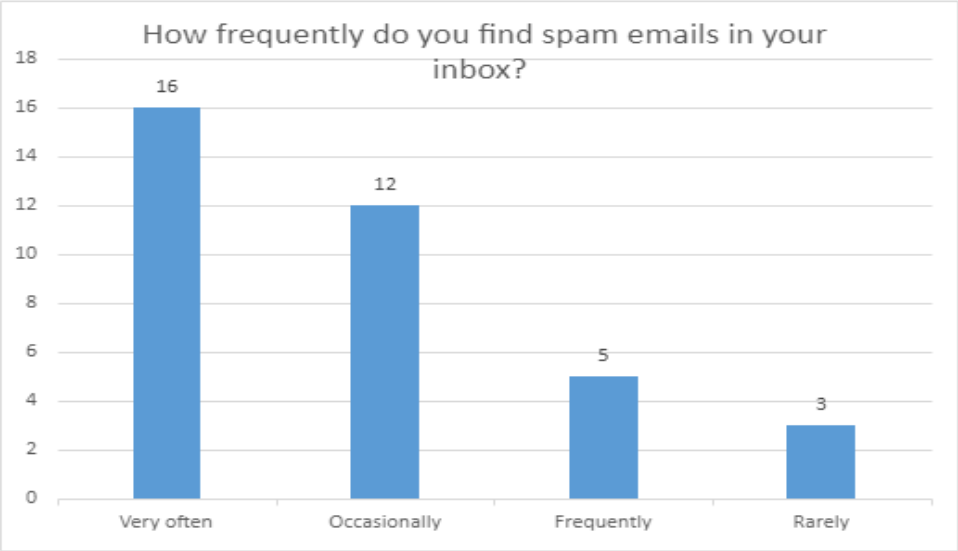

Figure 23. Bar graph shows the frequency of spam email found on the inbox on the participant emails.

From the above bar-graph, we can see that there were maximum number of participants who use to find the spam email in their inbox email (i.e., 16). Furthermore, 12 and 5participants find the spam email occasionally and frequently on their inbox respectively. However, just 3 participants mentioned that they find the spam email rarely on their inbox.

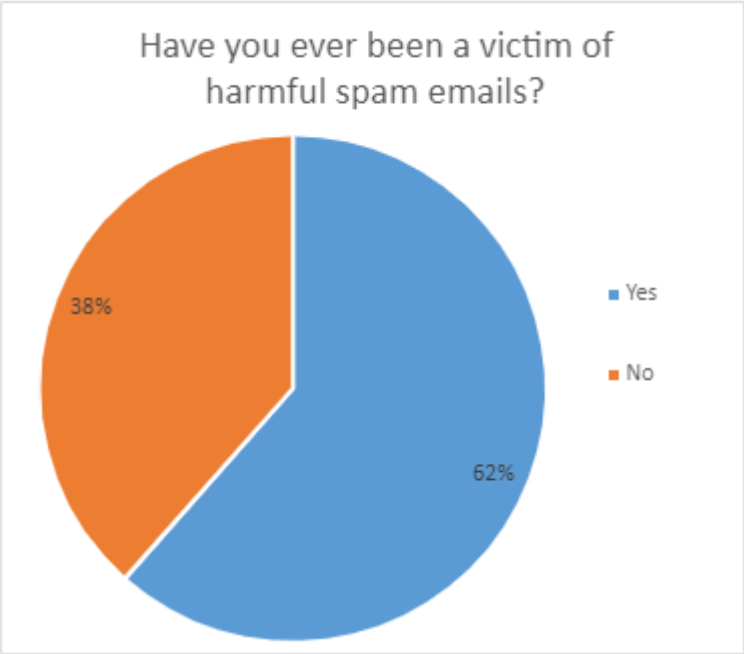

Figure 24. Pie-chart shows the number of participants haunted by harmful spam emails.

Throughout the survey, 62% of the respondents mentioned that they were haunted by the harmful email (i.e., which contain the malicious URL that leads the user to loss of data or financial loss) and 38% of the respondents were safe from the harmful emails.

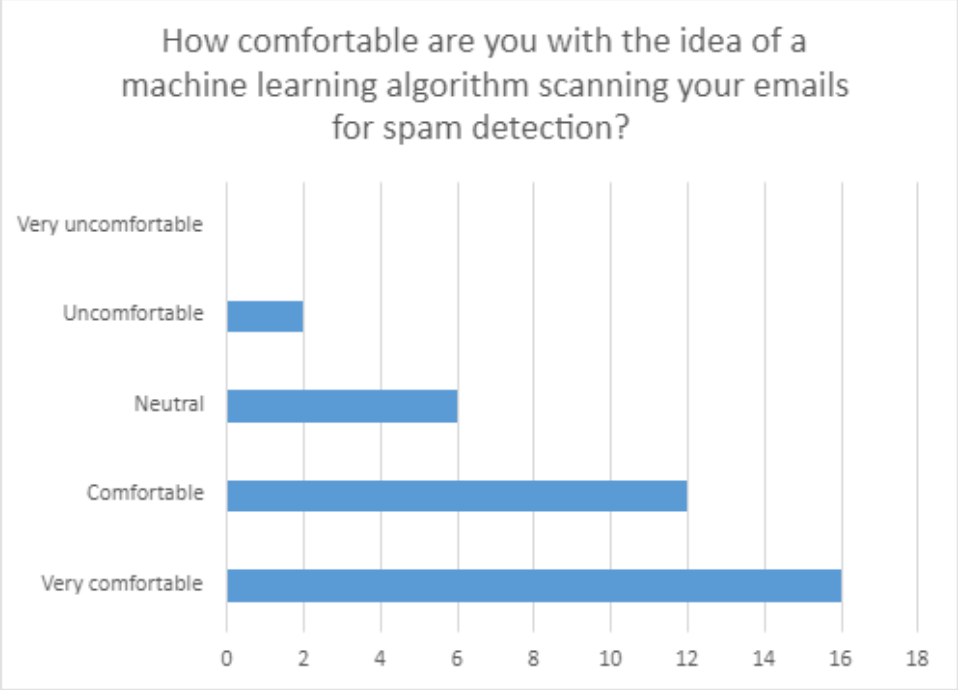

Figure 25. Bar-graph shows the knowledge of the participants in the Machine Learning.

From the above bar-graph, we can see that 16 of them were very comfortable with the machine learning knowledge. However, there were not any participant who are unknown about the machine learning knowledge. There were 12, 6 and 2 of the participants were comfortable, Neutral and uncomfortable with the machine learning idea.

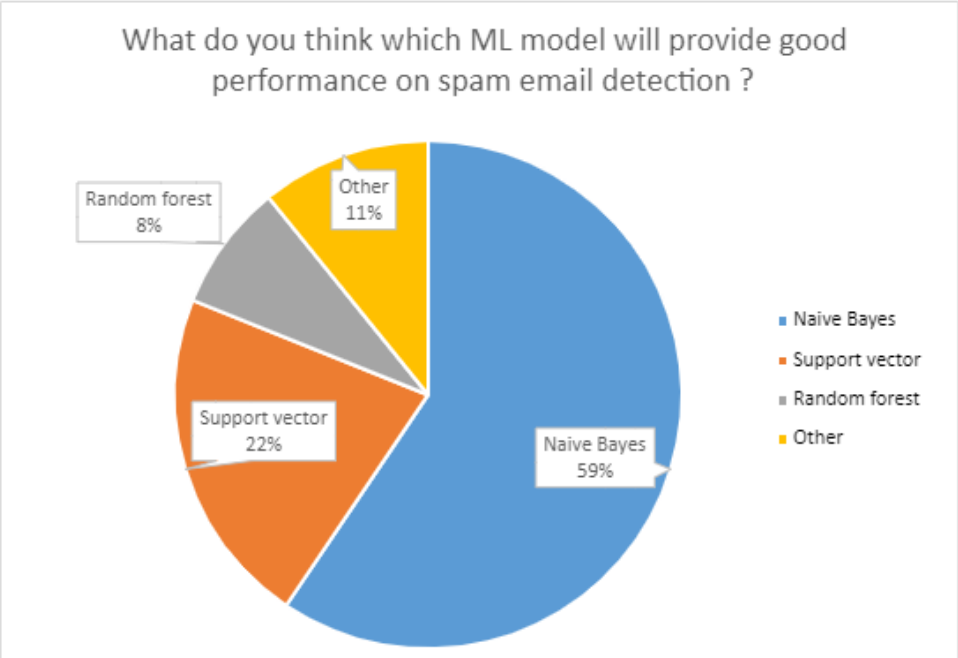

Figure 26. Pie-chart shows the ML model suggestion of the participant that should be used to build the system.

As the result of the survey from the above figure, it was found that 59% participants had suggested to use the naive bayes ML model to build the system while only 8% of participants had suggested to use the random forest. Furthermore, 22% of participants suggested to use the support vector ML models. Similarly, 11% of participants answered to use other model like Logistic regression model and K Neighbour model.

Chapter 5: **Evaluation**

5.1 Introduction:

Throughout this chapter, I will focus on the evaluation and the reflection of the project's research outcomes, result of the implementation and Limitation and the challenges of the project. Furthermore, I will also be comparing the evaluation with the project objectives if the objectives of the project were fulfilled or not along with that I will also focus on the key findings and issues of the project.

5.2 Reflection on the Research:

Email has become one of the common and the most convenient ways of communicating with each other along with that the number of internet users is also increasing rapidly. Throughout the project deep research has been conducted related to the email classifier for the detection of spam email like historical perspectives, machine learning on the email classifiers etc.

While talking about the finding on the literature review, As the number of internet user are increasing rapidly, so the emails are being used from perspective of good and bad motive. It was also found that as number of systems were developed on the past to stop the spam email, but the number of spam email were increasing instead of decreasing as the people are adapting the new techniques of spreading the email so that system cannot detect as the email is spam. Additionally, it was also found that all the spam emails are not harmful to the user. Spam email can be sent to many users with the motive of commercial advertisement or of accessing the user's data and the system by attaching malicious things. As mentioned above email is the convenient way of sharing the information and the number of the internet users are increasing, people can do their business advertisements in the convenient way and the easily with the larger number of people which was also the one of the main reasons for increasing the spam emails.

Furthermore, when it comes to the building of the system for spam email detection using machine learning, supervised machine learning provides the best performance. Among all those machine learning models, it was found that Neural Network (NN), Support Vector Machine (SVM), and Naive Bayesian (NB) provides the best result. In the past, researchers were not able to develop the system with high accuracy which was the one of the key findings of the research along with that they were not able to identify the best way of increasing the accuracy of the systems.

When looking into the online survey analysis, As mentioned on the chapter 3 methodology, the online survey was shared within the university and the data were collected from the final year student of bachelor's degree along with that all the ethical consideration were considered very welly. Also, I would like to thank my supervisor for the help and guidance for the creation of the questioners needed on the research analysis. For the questioner's creation, Microsoft form was used and then after the question's creation, it was shared on Microsoft, making me easy to collect responses for the survey form. Furthermore, all the responses were exported to MS Excel, which was then analysed, which also makes it easy to make the result of the responses of the survey.

According to the survey results, most of the responded were 18 – 24 and almost all participants happily agreed to participate in the survey, but just one of them was not comfortable attending. Similarly, I have also found that more than 50% of participants use email multiple times a day and there was no one who used emails rarely which means email is the most common way of communicating and sharing information between each other. Throughout the survey, it was suggested to use the naive bayes machine learning model to build the ML model to make the prediction. Also, none of the participants were uncomfortable with the idea of machine learning, which means almost all were too comfortable with the machine learning model. Many

participants mentioned that they frequently received spam emails in their inbox of the email where most were harmful to them, from this it can be concluded that the recent system was not enough to detect them.

5.3 Reflection on the implementation:

5.3.1 Evaluation of System development:

This is another important part of the project where I will be developing the system where it can classify the enter email by the user is whether the spam or not, which was also the next part after the completion of the literature review. For the building of the system, I have chosen machine learning which can make predictions and furthermore, I have also developed a dashboard where the user can interact with the system.

As mentioned on the chapter 3 methodology, for the development of system, I have used the platform like Jupyter Notebook, PyCharm and Python programming language. Additionally, In the Python language, I had used various libraries like pandas, Sklearn, nltk etc. Similarly, the dataset containing the ham email and the spam is also used to train and test the machine learning models where the dataset was taken from the Kaggle website which was prepared by the open-source database. As I was a beginner in machine learning, I have used various platforms to build the skill in it.

Throughout the research, I have found the naive bayes ML model provides the good performance, So, firstly, I have used different Naive Bayes ML models to build the system and tried with the other machine learning model to check which will provide good performance. After comparing the performance of different models, I have finally used MultinomialNB model. Additionally, Streamlit open source was used to build the dashboard to make the interaction between the users and the system and two of them were connected using the pickle. Finally, after developing the system, I have tested the system to see whether the result of the system meets the expectation or not. To do so, I have taken some of the ham and spams email from GitHub and I had done the testing. As a result, the system was classifying whether the entered emails are spam or not.

5.3.2 Issues and Challenges in implementation:

While implementing the system, I have faced a lot of issues and challenges. As mentioned above, Machine learning was the beginning for me, so, I must learn the machine learning to build the system. Once after getting the intermediate machine learning knowledge, another issue was to choose the model to make predictions. Although I tried other models as well to compare the performance it took me a long time to learn machine learning.

One of the main challenges was to find the dataset which has the balanced data of ham and spam emails. As the dataset used on test and train the model were unbalanced, another challenge was to decide which model to be used. In the implementation, I had to use many libraries which were complex and needed to be installed on the PC which was another challenge. After finalizing the model and testing, the model was ready, and the dashboard for the user interaction was also ready, but another challenge was to connect them to each other. While connecting them, the issues I was facing was to create the environment but the low power of the laptop. However, all the issues and challenges were fixed one by one, and the final system was built.

5.4 Reflection on the project's objectives:

On this heading, I will be analysing my initial project's objectives from my proposal to check whether I am able to achieve them or not. While looking into the first objective, I was able to research all the machine learning algorithms and the literature related to email classification which were covered in the literature review part of the project. Throughout the project, I have conducted an online survey and used the quantitative analysis method to find the conclusion as

well, so the other objectives were also fulfilled on the project. The implementation objective was also fulfilled, as I have developed machine learning model and the user-interface to interact with the users and system was tested as well. Finally, the system was implemented with good functioning, so it was developed more than the demo. The last objective of the project was also fulfilled throughout the project which was finding the result of the outcomes.

Chapter 6: **Conclusion**

6.1 Introduction:

In this chapter, I will be discussing about the concluding the project. It contains the summary of key finding of the project and limitation of the research and the project. Most importantly, it will be the closing of the project where I will be discussing about the final summary of the project in brief with all the information which will help the future researcher for the implementation of the project.

6.2 Summary of key findings:

While looking toward the literature review of the project, it contains all the research related to the email classifier. The main purpose of literature review is to find the reason of increasing and the historical perspective of spam emails along with the way of solving the issue of spam emails. Throughout the literature review, one main thing I have found is all the spam emails are not harmful to the users but it is also used to advertise the business, as the number of internet users are increasing in this era which makes the businessman to advertise their business in the conveniently and easily, and what was the reason for increasing the spam emails instead of decreasing although the system for spam detection are implemented. Furthermore, as there are the lots of spam detection techniques but those are not sufficient, and number of people are also increasing who are getting haunted by the spam emails that contains the malicious things. As spammers are using the new techniques to create the spam email which can manipulate the techniques which cannot detect the spam emails and they can easily share them to the users which was the one of reason for increasing the number of haunted users. From the survey, I have also found that Naïve bayes machine learning model is the one of the best techniques to build the system. Additionally, it was also noticed the number of harmful emails is dominating over the normal emails and the previous techniques were not able to detect them. Similarly, among all the participants the number of users haunted by spam emails were larger than the other one from which it can be concluded that implementation of these techniques needs to be taken into the consideration.

When we look into the implementation of the project, development of the system was the most important part of the project using the machine learning with the good functionality. Data cleaning, analysis of the dataset, preprocessing, model building, evaluating the model and improvement of the model are the main step needed to implement the system. Throughout the implementation, the dataset was collected from the open source and analysed. Once the dataset was cleaned, pre-processed, I have built the model using naïve bayes first and used the various model from which the comparison of performance of these models were done. Finally, looking at the models after applying the improvement of the model, Multinomial Naïve bayes model was finalized to make the prediction of the emails. As a result, I have found that the Naïve bayes is the best model to build the system depending on the dataset used, as on point the dataset was unbalanced which means the number of ham emails were larger than spam emails which leads to decrease the accuracy of the model, but the precision was 100% that means it was not give the false prediction.

6.3 Project/Research Limitation:

While carrying out the project, there were lots of the limitation in the project. One of the limitations was the time which means the project should be completed on the constraint deadline

with the good functionality of the systems. Durning the quantitative analysis of the data collected from the survey, responses might not be precise, and I might not be able to improve the performance of the system which could be increased by making the interaction of the responded and the system with each other from which I might come with the better conclusion and find the section to increase the quality and the performances of the project. Furthermore, while conducting the literature review, I was not able to find the best explanation which could effect on the project. Similarly, I have also found some of the gaps in the literature, on which the researcher had not explain about the feature selection by which the performance and the accuracy of the system might be improved.

Furthermore, In the implementation of the project, one of the biggest limitations is the dataset (i.e., the dataset used was unbalanced and it was difficult to find the balanced dataset which could leads to the increase of the system performance). Additionally, in this project, I have also planned to make the user interaction where I had to connect the trained ML model and the dashboard on which I have to create the environment between them to do so the performance of the device need to be good. Similarly, for the building of the system using ML, I had to use the complex and the various libraries which was leading to the decrease in the performance of the computer device. Along with all these limitations, finally, I was able to build and implement the project with the good functionality and as a result all the project objectives were also fulfilled.

6.4 Future research and recommendations

With the increase in the use of the technology the number of the internet users are also increasing, and spam email detection has become one of the important topics that need to be research in this rapidly increasing world. As mentioned previously on the project, people are using the new techniques of writing and sending the spam email due to which system of spam detection are not able to detect them, so the researchers need to focus on the new trends and the techniques for spam emails detections. Furthermore, I will also suggest the researchers to make the user interaction for the research analysis in person through which they can also find the way of increasing the performances of the systems.

Additionally, the researchers are suggested to use the balanced dataset by which they can increase the accuracy of the systems. They can also use the same idea and techniques I have used on the project however I would suggest them to use the complex machine learning model algorithm looking into their requirements. As the system I have implemented will only be applicable in the computer, so they can implement the system which can be applicable for the small devices like mobile phones, tablets etc. Furthermore, they can build the dashboard in more good visual way through which the user can interact with the system.

6.5 Significances of the project

This project has a great significance in spam emails detection using the machine learning. As the world is rapidly shifting toward the use of the technology with the increasing of internet users, spam emails are increasing. The one of the main purposes of this project is to develop a system using ML which can detect the spam emails and was to contribute to the development of spam email detection techniques with the high performance and help the future researcher as well. While talking about spam email detection, this project also contributes towards the finding about the way to increase the accuracy of the systems. Along with that, it also shows why the development of email detection techniques need to be considered. From the perspectives of the users, they can check that the emails they had received are safe to use or not.

Reference List

Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K. and Alazab, M., 2019. A comprehensive survey for intelligent spam email detection. IEEE Access, 7, pp.168261-168295.

W. Z. Khan, M. K. Khan, F. T. Bin Muhaya, M. Y. Aalsalem and H. -C. Chao, "A Comprehensive Study of Email Spam Botnet Detection," in IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2271-2295, Fourthquarter 2015, doi: 10.1109/COMST.2015.2459015.

Cormack, G.V., 2008. Email spam filtering: A systematic review. Foundations and Trends® in Information Retrieval, 1(4), pp.335-455.

Alpha, V. (2018). Everything you need to know about email classification. [online] Visible Alpha. Available at: https://visiblealpha.com/blog/email-classification-overview/ [Accessed 21 Nov. 2023].

Mujtaba, Ghulam & Shuib, Liyana & Raj, Ram & Majeed, Nahdia & Al-Garadi, Mohammed. (2017). Email Classification Research Trends: Review and Open Issues. IEEE Access. PP. 1-1. 10.1109/ACCESS.2017.2702187.

Aery, M. and Chakravarthy, S., 2005, November. emailsift: Email classification based on structure and content. In *Fifth IEEE International Conference on Data Mining (ICDM'05)* (pp. 8-pp). IEEE.

Panigrahi, P.K., 2012, November. A comparative study of supervised machine learning techniques for spam e-mail filtering. In *2012 Fourth International Conference on Computational Intelligence and Communication Networks* (pp. 506-512). IEEE.

Youn, S. and McLeod, D., 2007. A comparative study for email classification. In *Advances and innovations in systems, computing sciences and software engineering* (pp. 387-391). Springer Netherlands.

Mujtaba, G., Shuib, L., Raj, R.G., Majeed, N. and Al-Garadi, M.A., 2017. Email classification research trends: review and open issues. *IEEE Access*, *5*, pp.9044-9064.

Iqbal, K. and Khan, M.S., 2022. Email classification analysis using machine learning techniques. *Applied Computing and Informatics*.

Yi, Y.F., Li, C.H. and Song, W., 2008, July. Email Classification Using Semantic Feature Space. In *2008 International Conference on Advanced Language Processing and Web Information Technology* (pp. 32-37). IEEE.

IBM (2023). *What is Machine Learning?* [online] IBM. Available at: https://www.ibm.com/topics/machine-learning.

Li, W. and Meng, W., 2015, June. An empirical study on email classification using supervised machine learning in real environments. In *2015 IEEE International Conference on Communications (ICC)* (pp. 7438-7443). IEEE.

T. Vyas, P. Prajapati, and S. Gadhwal, “A survey and evaluation of supervised machine learning techniques for spam e-mail filtering,” in *Proceedings of the 2015 IEEE international conference on electrical, computer and communication technologies (ICECCT)*, IEEE, Tamil Nadu, India, March 2015.

Ahmed, N., Amin, R., Aldabbas, H., Koundal, D., Alouffi, B. and Shah, T., 2022. Machine learning techniques for spam detection in email and IoT platforms: Analysis and research challenges. *Security and Communication Networks*, *2022*, pp.1-19.

Mansoor, R.A.Z.A., Jayasinghe, N.D. and Muslam, M.M.A., 2021, January. A comprehensive review on email spam classification using machine learning algorithms. In 2021 International Conference on Information Networking (ICOIN) (pp. 327-332). IEEE.

Kanade, V. (2022). *What Is Machine Learning? Definition, Types, Applications, and Trends for 2022*. [online] Spiceworks. Available at: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ml/.

Verma, S. and Gautam, A.K., 2019, September. Machine Learning Techniques for Classification of Spambase Dataset: A Hybrid Approach. In *Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control* (pp. 1-6).

Adewale, A.E. and Yamazaki, T., 2023, February. Fundamental Sentiment Analysis by Natural Language Processing and Machine Learning for Email Classification. In *Proceedings of the 2023 5th Asia Pacific Information Technology Conference* (pp. 103-105).

Gomes, S.R., Saroar, S.G., Mosfaiul, M., Telot, A., Khan, B.N., Chakrabarty, A. and Mostakim, M., 2017, September. A comparative approach to email classification using Naive Bayes classifier and hidden Markov model. In *2017 4th international conference on advances in Electrical Engineering (ICAEE)* (pp. 482-487). IEEE.

Kumar, N. and Sonowal, S., 2020, July. Email spam detection using machine learning algorithms. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 108-113). IEEE.

Mahesh, B., 2020. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, *9*(1), pp.381-386.

Sarker, I.H., 2021. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, *2*(3), p.160.

Kaelbling, L.P., Littman, M.L. and Moore, A.W., 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*, *4*, pp.237-285.

Patel, H. (2021). *What is Feature Engineering — Importance, Tools and Techniques for Machine Learning*. [online] Medium. Available at: https://towardsdatascience.com/what-is-feature-engineering-importance-tools-and-techniques-for-machine-learning-2080b0269f10.

Zareapoor, M. and Seeja, K.R., 2015. Feature extraction or feature selection for text classification: A case study on phishing email detection. *International Journal of Information Engineering and Electronic Business*, *7*(2), p.60.

Jason Brownlee (2019). *A Gentle Introduction to the Bag-of-Words Model*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/gentle-introduction-bag-words-model/.

Atlassian (2019). *Agile Best Practices and Tutorials | Atlassian*. [online] Atlassian. Available at: https://www.atlassian.com/agile.

Martins, J. (2022). *What Is Kanban? A Beginner's Guide for Agile Teams [2023] • Asana*. [online] Asana. Available at: https://asana.com/resources/what-is-kanban.

Bhat, A. (2018). *Online Surveys: Definition, Characteristics, Examples, Advantages and Disadvantages*. [online] QuestionPro. Available at: https://www.questionpro.com/blog/what-are-online-surveys/.

Appendix A - Initial Project Proposal

Project (CN6000) Initial Proposal Form

Programme: BSc (Hons) Cyber Security and Networks Year: 2023/24

Student Number: 2195276

Proposed Title: Email Classifier to detect spam email or legitimate (i.e., ham) email.

Proposed Aim: The aim of the project is to develop the demo of the email classifier which helps to detect if the email is either spam or legitimate (i.e., ham).

Rationale:

This project should be undertaken due to increasing in the volume of spam email which is affecting the individual as well as the organization. Nowadays, the spam emails are wasting the time, and some of them are also can be dangerous to the individuals. Furthermore, accurate email classification is required due to the security dangers provided by spam, which frequently includes malware and phishing efforts. Addressing these problems can significantly improve individual experiences, especially in the business context where the emails management directly impact the business financial resources and their data. Exploring the machine learning and natural language processing in email classifier aims is to provide an efficient and accurate email classifier, empowering the individual to regain their access in their inbox.

Supervisor: Dr. Umar Mukhtar Ismail

Appendix B - Final Project Proposal

Project (CN6000) Final Proposal Form

Programme: BSc (Hons) Cyber Security and Networks Year: 2023/24

Student Number: 2195276

Proposed Title: Email Classifier to detect spam email or legitimate (i.e., ham) email.

Proposed Aim: The aim of the project is to develop the demo of the email classifier which helps to detect if the email is either spam or legitimate (i.e., ham).

Objectives:

1. To research different machine algorithms related to email classification.
2. To research the literature and the approaches relating to the classification of email.
3. To conduct an online survey between people of the related field through which the feedback can be collected.
4. To use quantitative analysis method to find the best conclusion on the project.
5. To implement a demo of machine learning techniques and the algorithm to classify the emails.
6. To evaluate the outcome of the implemented machine learning techniques measuring its performance and effectiveness.
7. To reflect on the results and findings of the project, identifying areas of success and the areas for improvement, which will also suggest future research in this field.

Rationale:

This project should be undertaken due to increasing in the volume of spam email which is affecting the individual as well as the organization. Nowadays, the spam emails are wasting the time, and some of them are also can be dangerous to the individuals. Furthermore, accurate email classification is required due to the security dangers provided by spam, which frequently includes malware and phishing efforts. Addressing these problems can significantly improve individual experiences, especially in the business context where the emails management directly impact the business financial resources and their data. Exploring the machine learning and natural language processing in email classifier aims is to provide an efficient and accurate email classifier, empowering the individual to regain their access in their inbox.

Facilities required:

- Python
- Machine learning libraries
- Jupyter notebook
- Email dataset

Supervisor: Dr. Umar Mukhtar Ismail

Appendix C – Questioners used for the online survey.

CN6000 Dissertation

Questioner for final year project on spam email detection using machine learning

1. Do you agree to take part in the survey, all your answer will be stored and will be used on the project? *

- ◯ Yes, I agree
- ◯ No, I don't

2. What is your age? *

- ◯ Under 18
- ◯ 18-24
- ◯ 25-34
- ◯ 35+
- ◯ Prefer not to say

3. How often do you use emails ? *

- ◯ Multiple time in a day
- ◯ Once a day
- ◯ Once a week
- ◯ 2 - 3 time a week
- ◯ Rarely

4. How frequently do you find spam emails in your inbox? *

- ◯ Very often
- ◯ Occasionally
- ◯ Frequently
- ◯ Rarely

5. Have you ever been a victim of harmful spam emails? *

- ◯ Yes
- ◯ No

6. How comfortable are you with the idea of a machine learning algorithm scanning your emails for spam detection? *

- ◯ Very comfortable
- ◯ Comfortable
- ◯ Neutral
- ◯ Uncomfortable
- ◯ Very uncomfortable

7. What do you think which machine learning models will be provide good performance on spam email detection ? *

- ◯ Naive Bayes
- ◯ Support vector
- ◯ Random forest
- ◯ Other

Source Code

Codes for the Model Building:

In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
# the code is to detect the encoding inorder to red the file
import chardet
def detect_encoding(file_path):
    with open(file_path, 'rb') as f:
        result = chardet.detect(f.read())
    return result['encoding']

detected_encoding = detect_encoding('spam.csv')
```

In [3]:
```
df = pd.read_csv('spam.csv', encoding = (detected_encoding))
```

In [4]:
```
df.head(15)
```

In [5]:
```
df.shape
```

In [6]:
```
#process for the project
# 1. Data cleaning
# 2. EDA
# 3. Text Preprocessing
# 4. Model building
# 5. Evaluation
# 6. Improvement
# 7. Website
# 8. Deploy
```

1.Data cleaning

In [7]:
```
df.info()
```

In [8]:
```
# As the 3 coluums didnot have the more data (i.e, Null value), So i ma deleting those cloumn
#droping the column (Inplace is to fixed the change)
df.drop(columns=['Unnamed: 2','Unnamed: 3', 'Unnamed: 4'], inplace=True)
```

In [9]:
```
df.sample(10)
```

In [10]:
```
# renaming the cols
df.rename(columns={'v1':'target','v2':'text'},inplace=True)
df.sample(10)
```

In [11]:
```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
```

In [12]:
```
# Labeling the number for ham and spam email
encoder.fit_transform(df['target'])
```

In [13]:
```
# Ham Label as 0
# Spam Label as 1
df['target'] = encoder.fit_transform(df['target'])
```

In [14]:
```
df.head()
```

In [15]:
```
# missing values (Checking of the missing value)
df.isnull().sum()
```

In [16]:
```
# check for duplicate values
df.duplicated().sum()
```

In [17]:
```
# remove duplicates (Keeping the 1st value)
df = df.drop_duplicates(keep='first')
```

In [18]:
```
df.shape
```

2. EDA (Exploratory data analysis)

In [19]:
```
df.head()
```

In [20]:
```
# Counting of the value on target column (number of ham and spam me=ail)
df['target'].value_counts()
```

In [21]:
```
## plotting the the above % in pie chart, on pie chat i am showing in % so autopct
import matplotlib.pyplot as plt
plt.pie(df['target'].value_counts(), labels=['ham','spam'],autopct="%0.2f")
plt.show()
```

In [22]:
```
 # Data is imbalanced as per the pie chart.
```

In [23]:
```
import nltk
```

In [24]:
```
!pip install nltk
```

In [25]:
```
nltk.download('punkt')
```

In [26]:
```
## calculating the numv=ber of character on the text for further analysis
#df['num_characters'] = df['text'].apply(lambda x: len(str(x)))
df['num_characters'] = df['text'].apply(len)
```

In [27]:
```
df.head(10)
```

In [28]:
```
# num of words
df['num_words'] = df['text'].apply(lambda x:len(nltk.word_tokenize(x)))
```

In [29]:
```
df.head()
```

In [30]:
```
df['num_sentences'] = df['text'].apply(lambda x:len(nltk.sent_tokenize(x)))
```

In [31]:
```
df.head(10)
```

In [32]:
```
# Inorder to see what happening with these above count like avg
df[['num_characters','num_words','num_sentences']].describe()
```

In [33]:
```
# description of ham like above
df[df['target'] == 0][['num_characters','num_words','num_sentences']].describe()
```

In [34]:
```
# description of ham like above
df[df['target'] == 1][['num_characters','num_words','num_sentences']].describe()
```

In [35]:
```
# plotting the analysis in histogram
import seaborn as sns
```

In [36]:
```
# plotting the number of characetr for comparision of ham and spam
plt.figure(figsize=(10,5))
sns.histplot(df[df['target'] == 0]['num_characters'])
sns.histplot(df[df['target'] == 1]['num_characters'],color='red')
```

In [37]:
```
# plotting the number of words for comparision of ham and spam
plt.figure(figsize=(10,5))
sns.histplot(df[df['target'] == 0]['num_words'])
sns.histplot(df[df['target'] == 1]['num_words'],color='red')
```

In [38]:
```
## to see the relationship between each of them like to see the relation of characeter with word, sentences etc
#(i.e, pair comparision)
sns.pairplot(df,hue='target')
```

In [39]:
```
## checking the correlation in heat map
numeric_columns = df.select_dtypes(include=['number']).columns
df[numeric_columns].corr()
```

In [40]:
```
sns.heatmap(df[numeric_columns].corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.show()
```

3.Data (Text) Pre-Processing

```
#.conversion in lowercase
#.Tokenization
#.Removing special characters
#.Removing stop words and punctuation
#.Stemming (changing of words to the its original form)
```

In [41]:
```
from nltk.corpus import stopwords
#stopwords.words('english')
```

In [42]:
```
import string
#string.punctuation
```

In [43]:
```
## Stemming (changing of words to the its original form)
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
ps.stem('loving')
```

In [44]:
```
def transform_text(text):
    text = text.lower() # Convert into the lowercase
    text = nltk.word_tokenize(text) # break the word in the text

    # running the loop to remove the special character (i.e to just keep alphabet and the numbers)
    A = []
    for i in text:
        if i.isalnum():
            A.append(i)

    text = A[:]
    A.clear()
     # running the loop to remove the stop words and punctuation
    for i in text:
        if i not in stopwords.words('english') and i not in string.punctuation:
            A.append(i)
    text = A[:]
    A.clear()

    for i in text:
        A.append(ps.stem(i))

  return " ".join(A)
```

In [45]:
```
# to check the above code is running
transform_text("I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today"
```

Out[45]: 'gon na home soon want talk stuff anymor tonight k cri enough today'

In [46]:
```
df['transformed_text'] = df['text'].apply(transform_text)
```

In [47]:
```
df.head(10)
```

In [48]:
```
## creating word cloud where the most used word will be foucesd and display
#!pip install wordcloud
```

In [49]:
```
from wordcloud import WordCloud
wc = WordCloud(width=500,height=500,min_font_size=10,background_color='white')
```

In [50]:
```
# converting all the transformed text into string using .astype(str)
# highlingh the spam words
#spam_wc = wc.generate(df[df['target'] == 1]['transformed_text'].astype(str).str.cat(sep=" ")) did the mistake in transformed
spam_wc = wc.generate(df[df['target'] == 1]['transformed_text'].str.cat(sep=" "))
```

In [51]:
```
plt.figure(figsize=(15,6))
plt.imshow(spam_wc)
```

In [52]:
```
# converting all the transformed text into string using .astype(str)
# highlingh the ham words
#ham_wc = wc.generate(df[df['target'] == 1]['transformed_text'].astype(str).str.cat(sep=" "))did the mistake in transformed_t
ham_wc = wc.generate(df[df['target'] == 0]['transformed_text'].str.cat(sep=" "))
```

In [53]:
```
plt.figure(figsize=(15,6))
plt.imshow(ham_wc)
```

In [54]:
```
df.head()
```

In [55]:
```
#spam_corpus = []

#for msg in df[df['target'] == 1]['transformed_text'].tolist():
 #   if isinstance(msg, str):  # Check if msg is a string
  #      for word in msg.split():
   #         spam_corpus.append(word)
    #elif isinstance(msg, list):  # Check if msg is a list
     #   spam_corpus.extend(msg)
```

In [56]:
```
spam_corpus = []
for msg in df[df['target'] == 1]['transformed_text'].tolist():
    for word in msg.split():
        spam_corpus.append(word)
```

In [57]:
```
len(spam_corpus)
```

Out[57]: 9930

In [58]:
```
from collections import Counter
spam_corpus_dataframe = pd.DataFrame(Counter(spam_corpus).most_common(30), columns=['Word', 'Repetition of words'])
spam_corpus_dataframe
```

In [59]:
```
sns.barplot(x ='Word', y='Repetition of words', data=spam_corpus_dataframe)
plt.xticks(rotation='vertical')
plt.show()
```

In [60]:
```
#ham_corpus = []

#for msg in df[df['target'] == 0]['transformed_text'].tolist():
 #   if isinstance(msg, str):  # Check if msg is a string
  #      for word in msg.split():
   #         ham_corpus.append(word)
```

In [61]:
```
ham_corpus = []
for msg in df[df['target'] == 0]['transformed_text'].tolist():
    for word in msg.split():
        ham_corpus.append(word)
```

In [62]:
```
len(ham_corpus)
```

Out[62]: 35296

In [63]:
```
ham_corpus_dataframe = pd.DataFrame(Counter(ham_corpus).most_common(30), columns=['Word', 'Repetition of words'])
ham_corpus_dataframe
```

In [64]:
```
sns.barplot(x ='Word', y='Repetition of words', data=ham_corpus_dataframe)
plt.xticks(rotation='vertical')
plt.show()
```

In [65]:
```
# Text Vectorization
# using Bag of Words
df.head()
```

4. Model Building

In [66]:
```
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
cv = CountVectorizer()
tfidf = TfidfVectorizer(max_features=2000) # 2000 means it will vectorive the most used 2000 word.
```

In [67]:
```
# Using Count Vectorizer
#X = cv.fit_transform(df['transformed_text']).toarray() # to check with the counter vector
```

In [68]:
```
# Using TFIDF vectorizer
X = tfidf.fit_transform(df['transformed_text']).toarray()
```

In [69]:
```
# using scaler for the improvement of the algorithm and cant use standerd scaler because it will give -ve vale
# but naive bayes didn't accept the -ve value
#from sklearn.preprocessing import MinMaxScaler
#scaler = MinMaxScaler()
```

In [69]:
```
X.shape
```

In [70]:
```
y = df['target'].values
y
```

Out[70]:
```
array([0, 0, 1, ..., 0, 0, 0])
```

In [71]:
```
from sklearn.model_selection import train_test_split
```

In [72]:
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=2)
```

In [73]:
```
# importing all the naive bayes as we dont know the dataset
from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB
from sklearn.metrics import accuracy_score,confusion_matrix,precision_score
```

In [74]:
```
gnb = GaussianNB()
mnb = MultinomialNB()
bnb = BernoulliNB()
```

In [77]:
```
#gnb.fit(X_train,y_train)
#y_pred1 = gnb.predict(X_test)
#print(accuracy_score(y_test,y_pred1))
#print(confusion_matrix(y_test,y_pred1))
#print(precision_score(y_test,y_pred1))
```

In [78]:
```
#bnb.fit(X_train,y_train)
#y_pred3 = bnb.predict(X_test)
#print(accuracy_score(y_test,y_pred3))
#print(confusion_matrix(y_test,y_pred3))
#print(precision_score(y_test,y_pred3))
```

In [75]:
```
mnb.fit(X_train,y_train)
y_pred2 = mnb.predict(X_test)
print(accuracy_score(y_test,y_pred2))
print(confusion_matrix(y_test,y_pred2))
print(precision_score(y_test,y_pred2))
```

In [80]:
```
# Now with model check we have an option to go with BNB or MNB. And in this case precision score matter due to unbalance of
# data so i am going with TfidfVectorizer MNB tfidf --> MNB
```

In [295]:
```
# now bringing all the machine learning code into effect to compare them
```

In [296]:
```
# to install the xgboost module in the device or algorithm
# pip install xgboost
```

In [297]:
```
#from sklearn.linear_model import LogisticRegression
#from sklearn.svm import SVC
#from sklearn.naive_bayes import MultinomialNB
#from sklearn.tree import DecisionTreeClassifier
#from sklearn.neighbors import KNeighborsClassifier
#from sklearn.ensemble import RandomForestClassifier
#from sklearn.ensemble import AdaBoostClassifier
#from sklearn.ensemble import BaggingClassifier
#from sklearn.ensemble import ExtraTreesClassifier
#from sklearn.ensemble import GradientBoostingClassifier
#from xgboost import XGBClassifier
```

In [298]:
```
# creating all the object for all the module above
#svc = SVC(kernel='sigmoid', gamma=1.0)
#knc = KNeighborsClassifier()
#mnb = MultinomialNB()
#dtc = DecisionTreeClassifier(max_depth=5)
#lrc = LogisticRegression(solver='liblinear', penalty='l1')
#rfc = RandomForestClassifier(n_estimators=50, random_state=2)
#abc = AdaBoostClassifier(n_estimators=50, random_state=2)
#bc = BaggingClassifier(n_estimators=50, random_state=2)
#etc = ExtraTreesClassifier(n_estimators=50, random_state=2)
#gbdt = GradientBoostingClassifier(n_estimators=50,random_state=2)
#xgb = XGBClassifier(n_estimators=50,random_state=2)
```

In [299]:
```
# creating a dictionary where in keys there are algorithm name
#clfs = {
#    'SVC' : svc,
#    'KN' : knc,
#    'NB': mnb,
#    'DT': dtc,
#    'LR': lrc,
#    'RF': rfc,
#    'AdaBoost': abc,
```

```
#     'BgC': bc,
#     'ETC': etc,
#     'GBDT':gbdt,
#     'xgb':xgb
#}
```

In [300]:
```
# creating a function as train classifier where i am givimg the train and test dataset where it can train the data set

#def train_classifier(clf,X_train,y_train,X_test,y_test):
#    clf.fit(X_train,y_train)                                  # this line train this classifier in those data set
#    y_pred = clf.predict(X_test)                              # finally calculate the accuracy, precission according to
#    accuracy = accuracy_score(y_test,y_pred)                  # the classifier and the for those module
#    precision = precision_score(y_test,y_pred)

#    return accuracy,precision
```

In [301]:
```
# to check the above function work or not I run the function using support vector (SVC)
#train_classifier(svc,X_train,y_train,X_test,y_test)
```

In [302]:
```
#       creting the loop to run in the above dictionary where this will store all the accuracy
#        and the precision value for each module according to their name
#
#accuracy_scores = []
#precision_scores = []

#for name,clf in clfs.items():

#    current_accuracy,current_precision = train_classifier(clf, X_train,y_train,X_test,y_test)

#    print("For ",name)
#    print("Accuracy - ",current_accuracy)
#    print("Precision - ",current_precision)

#    accuracy_scores.append(current_accuracy)
#    precision_scores.append(current_precision)
```

In [303]:
```
# creating the above output of the accuracy and precision in dataframe and sorting by precision
#performance_df = pd.DataFrame({'Algorithm':clfs.keys(),'Original_Accuracy':accuracy_scores,'Original_Precision':precision_sc

#performance_df
```

In [304]:
```
# from the above as well the top five are showing the best precision but the firswt one accuracy is too low
# and in this case precision will matter due to imbalance dataset so i am moving withNaive bayes because of its precision
# high and the accuracy is also not bad
```

In [305]:
```
# to plot the above data into the plot
#performance_df1 = pd.melt(performance_df, id_vars = "Algorithm")

#performance_df1

#sns.catplot(x = 'Algorithm', y='value',
#            hue = 'variable',data=performance_df1, kind='bar',height=5)
#plt.ylim(0.5,1.0)
#plt.xticks(rotation='vertical')
#plt.show()
```

model improve

In [306]:
```
# 1. Change the max_features parameter of TfIdf
```

In [307]:
```
# this will append the new data that comes from the output to improvement the model(i.e using max_feature = 3000)
#temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_max_ft_2000':accuracy_scores,'Precision_max_ft_2000':precision_sco
#new_df = performance_df.merge(temp_df,on='Algorithm')
#new_df
```

In [308]:
```
#2. Scaling minmax method
# this will append the new data that comes from the output to check improvement the model(i.e using scaling minmax method)
#temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_scaling':accuracy_scores,'Precision_scaling':precision_scores}).so
```

In [309]:
```
#new_df_scaled = new_df.merge(temp_df,on='Algorithm')
```

In [310]:
```
#new_df_scaled
```

In [311]:
```
## By using scaling the accuracy wents more down in naive bayes and the precision also went down but mostly it didn't
## chnage anything in other, So I planned not to go with scaling.
```

```
In [310]: #new_df_scaled

In [311]: ## By using scaling the accuracy wents more down in naive bayes and the precision also went down but mostly it didn't
          ## chnage anything in other, So I planned not to go with scaling.

In [312]: ## trying to improve with the NUM-CHARACTERS column of dataset

In [313]: #temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_num_chars':accuracy_scores,'Precision_num_chars':precision_scores}
          #new_df_num_char = new_df_scaled.merge(temp_df,on='Algorithm')
          #new_df_num_char

In [314]: #new_df_num_char1 = pd.melt(new_df_num_char, id_vars = "Algorithm")

          #new_df_num_char1

          #sns.catplot(x = 'Algorithm', y='value',
          #            hue = 'variable',data=new_df_num_char1, kind='bar',height=7,)
          #plt.ylim(0.5,1.0)
          #plt.xticks(rotation='vertical')
          #plt.show()

In [315]: # finally with the comparison of these accuracy and precision I decided to go with tfid (Max_feature=3000) model
          # now the model is prepared

In [316]: # Now the step is to create the pipe line for connecting the model and the website to check the user entered email is spam
          # ham

In [317]: # for final setup of the model run the file 1 + to 2 and 3 and finally 4

In [76]:  import pickle
          pickle.dump(tfidf,open('vectorizer.pkl','wb'))
          pickle.dump(mnb,open('model.pkl','wb'))
```

Code for the user interface creation:

```
import streamlit as st
import pickle
import string
from nltk.corpus import stopwords
import nltk
from nltk.stem.porter import PorterStemmer

ps = PorterStemmer()

def transform_text(text):
    text = text.lower()  # Convert into the lowercase
    text = nltk.word_tokenize(text)  # break the word in the text

    a = []
    for i in text:
        if i.isalnum():
            a.append(i)

    text = a[:]
    a.clear()

    for i in text:
        if i not in stopwords.words('english') and i not in
     string.punctuation:
            a.append(i)

    text = a[:]
    a.clear()

    for i in text:
        a.append(ps.stem(i))
```

```
    return " ".join(a)

tfidf = pickle.load(open('vectorizer.pkl', 'rb'))
model = pickle.load(open('model.pkl', 'rb'))

st.title(":green[ :e-mail: _Email Classifier App: Spam or Ham_]")

input_email = st.text_area("Enter The Email You Want To Check",
     height=150, placeholder="Enter The Email You Want To Check",
     label_visibility="hidden")

if st.button("Predict", type="secondary"):

    # 1. preprocess
    transformed_email = transform_text(input_email)
    # 2. vectorize
    vector_input = tfidf.transform([transformed_email])
    # 3. predict
    result = model.predict(vector_input)[0]
    # 4. Display
    if result == 1:
        st.header("_Email you have entered is Spam._ :red[:warning: Be
     careful while using it:warning:]")
        #st.image("Spam.jpg")
    else:
        st.header("Email you have entered is not spam. You are Safe to
     access the email")
```