**Mini Project Report on**



**Email/SMS Spam**

**Detection**   


**Submitted in partial fulfillment of the requirement for the award of the degree of**

**MASTER**

**IN**

**COMPUTER APPLICATION**

**Submitted by:**

**Sarthak Jaiswal**  **2201344**

**Nitin Bhist**  **2201262**

***Under the Mentorship of***

**Mr. Amit Juyal**

**Assistant Professor**



**DEPARTMENT OF COMPUTER SCIENCE**

**AND ENGINEERING**

**GRAPHIC ERA HILL UNIVERSITY**

**May, 2023**

**CANDIDATE’S DECLARATION**

This is to certify that the thesis titled **“Email/Sms Spam Detection” submitted Sarthak Jaiswal**, to Graphic Era Hill University for the award of the degree of **Master in Computer Application**, is a bona fide record of the research work done by him/her under our supervision. The contents of this project in full or in parts have not been submitted to any other Institute or University for the award of any degree or diploma.

Place:Dehradun **Amit Juyal**

Date:11/7/23

GEHU,Dehradun

**ACKNOWLEDGEMENT**

I am highly indebted to my teachers and my parents for their Guidance and supervision as well as for providing necessary Information regarding this Project and also for helping in completing the report.

I would like to express my gratitude towards my parents and Graphic Era Hill University for their kind co-operation and encouragement which help me in completion of he report.

My thanks and appreciations also go to my colleague and people who have willingly helped me out with their abilities.

**Sarthak Jaiswal** 2201344

**Nitin Bisht** 2201262

**ABSTRACT**

The number of people using mobile devices increasing day by day. SMS(short message service) is a text message service available in smart phones as well as basic phones.So, the traffic of SMS increased drastically. The spam messages also increased. The hackers try to send spam messages for their financial or business benefits like market growth, lottery ticket information, credit card information, etc. So, spam classification has special attention. In this paper, we applied various machine learning techniques for SMS spam detection. We used a data set to train the machine learning models like naive bay’s and KNN. The SMS spam collection data set is used for testing the method. The data set is split into two categories for training and testing the research. Our experimental results have shown that out naive bay’s model outperforms previous models in spam detection with an accuracy of good.

**Table of Contents**

**Chapter No. Description Page no.**

Chapter 1 Introduction 1

Chapter 2 Methodology 4

Chapter 3 Snapshot of project 17

Chapter 4 Conclusion 18

Refrences 18

**INTRODUCTION**

* 1. **ABOUT PROJECT**

The increasing mobile phones become one of the attached companions for many individuals. With the explosive penetration of mobile devices and millions of people sending messages every day, Short Message Services (SMS) has become a multi-million-dollar commercial industry with a value between 11.3 to 24.7 percent of the developing countries’ Gross National Income (GNI) in the early year of 2013.

As the utilization of mobile phone devices has become commonplace, Short Message Services (SMS) has grown into a multi-billion dollars commercial industry. SMS is a text communication platform that allows mobile phone users to exchange short text messages (usually less than 160 seven-bit characters). It is the most widely used data subscribers at the end of 2010. As the popularity of the platform has increased, we have seen a surge in the number of unsolicited commercial advertisements sent to mobile phones using text messaging. SMS spam is still not as common as email spam, where 2010 around 90% of emails was spam, and in North America it is still not a major problem, contributing to less than 1% of text messages exchanged as of December 2012.

The spam increased in these days due more mobile devices deployed in environment for e-mail and message communication. Currently, 85% of mails and messages received by mobile users are spam. The cost of mails and messages are very low for senders but high for receipts of these messages. The cost paid some time by services providers and the cost of spam can be measured in the loss of human time and loss of important messages or mails. Due to these spam mails and messages, the values able emails and messages are affected because each user have limited internet services, short time, and memory.

The dataset is a large text file, in which watch line starts with the label of the message, followed by the text message string. After preprocessing of the data and extraction of features, machine learning techniques such as naive baye’s, SVM and other methods are applied to the samples, and their performances are compared. Finally, the performance of best classifier from the project is compared against the performance of classifiers applied in the original paper citing this dataset.

We proposed a spam detection method using machine learning algorithms such as NB (naive based) and LSTM for classification of ham and spam messages. The SMS spam collection dataset was considered for testing of the current research. The dataset was divided into two categories: 20% for testing and 80% for training purpose for the predictive models. The evaluation metrics for performance such as specificity, accuracy, and sensitivity were considered evaluating the proposed study. The results obtained from experiments confirmed that the proposed research achieved high accuracy.

* 1. **PROBLEM STATEMENT**

A number of major differences exist between spam-filtering in text messages and

emails. Unlike emails, which have a variety of large datasets available, real databases

for SMS spams are very limited. Additionally, due to the small length of text messages,

the number of features that can be used for their classification is far smaller than the

corresponding number in emails. Here, no header exists as well. Additionally, text

messages are full of abbreviations and have much less formal language that what one

would expect from emails. All of these factors may result in serious degradation in

performance of major email spam filtering algorithms applied to short text messages.

**1.3 MOTIVATIONS**

Being extremely interested in everything having a relation with the Machine Learning, the independent project was a great occasion to give me the time to learn and confirm my interest for this field. The fact that we can make estimations, predictions and give the ability for machines to learn by themselves is both powerful and limitless in term of application possibilities. We can use Machine Learning in Finance, Medicine, almost everywhere. That’s why I decided to conduct my project around the Machine Learning.

**1.4 IDEA**

This project was motivated by my desire to investigate the classification field of machine learning since it allows to approach natural language processing which is a very hot topic actually. I use the classification model to find out which of the SMS are spam or ham

**PROJECT**

**2.1 REQUIREMENT ANALYSIS**

1. SOFTWARE REQUIREMENT -
2. Python
3. Jupyter notebook
4. Visual-studio code
5. Latest web browser
6. HARDWARE REQUIREMENT - Windows and RAM(8-GB) minimum

**2.2 METHODOLOGY**

We use SMS/Email classification data set from the UCI machine learning repository. Then we load that dataset into our program. After that we remove the unwanted almost null column. Since we have only two columns i.e., target and text therefore we create three more column which determine the number of words, sentences and characters in the given message for better understanding of the data. Next, we perform pre-processing to clean, remove unwanted text, characters out of the message. After the preprocessing we split our data into training and testing and we train our classifier by fitting the train data to the classifier, there after prediction of results over unseen test data set is made which there after provides us with the accuracy with which the classifier had predicted the outcomes. There after we present our results in a pictorial manner which is the best way to showcase results because of its easiness to understand information out of it.

**2.2.1 Extraction of Data**

There are several datasets available for SMS classification such as the SMS Spam Collection and the UCI Machine Learning Repository. We took dataset from Kaggle. In that dataset we have 5,574 data entries. In this dataset we have 3 column which contain almost null values. To improve our data set we remove that columns and add extra features in our data frame which are number of characters, words, and sentences in the message.

In figure 2.2.2.1, this is our dataset before the modification and figure 2.2.2.2 showing the data set after the modification

With this our dataset is ready for Pre-processing.

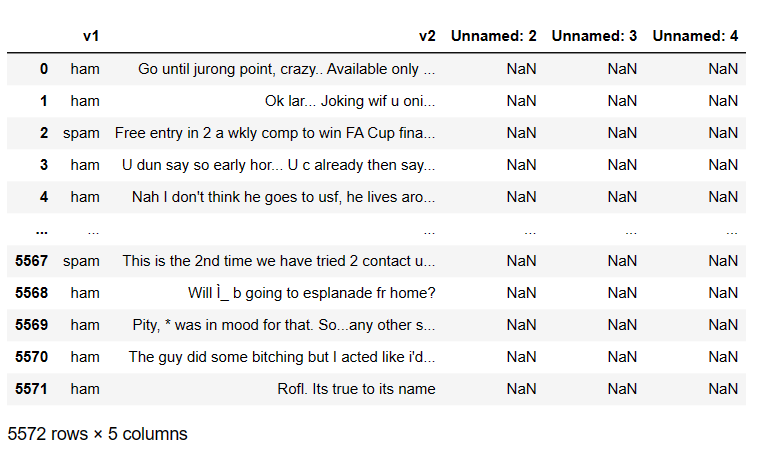


Fig **2.2.1.1**

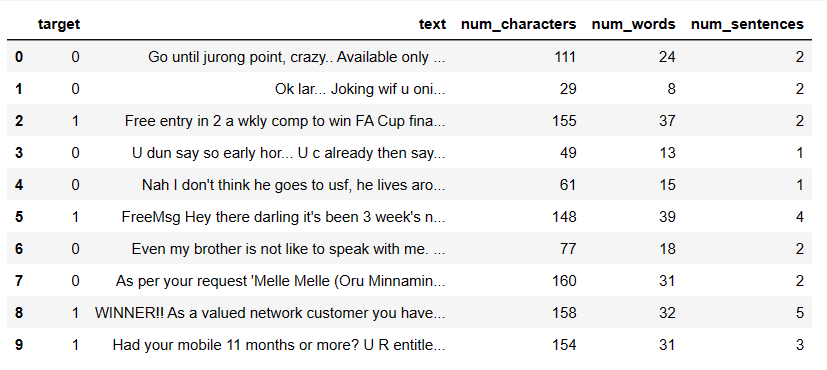


Fig 2.2.1.2

**2.2.2** **Pre-processing Of Data:**

Following are the Preprocessing steps that have been carried out:

At first, we know that machine learning does not work on textual data therefore first we have to convert the “spam” and “ham” into 0’s and 1’s. For this we use Labe Encounter form the sklearn to convert the target column into the binary form. In figure 2.2.2.1 you can clearly see the target column.

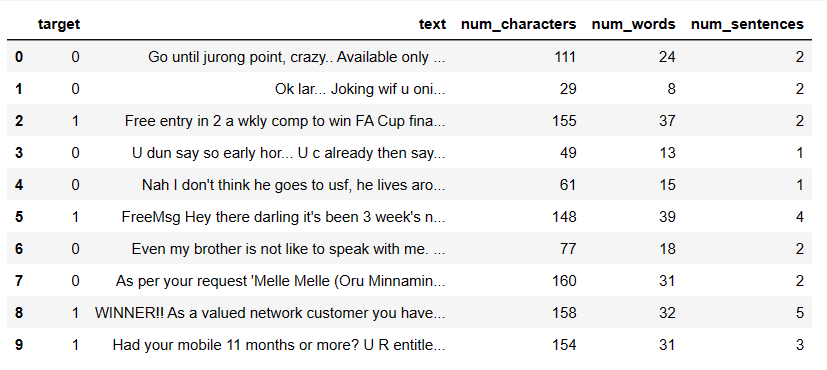


Fig 2.2.2.1

**Conversion to lowercase:**

To maintain uniformity all the messages, we convert them to lowercase .This will benefit to avert inconsistency in data. Python provides a function called lower() to convert sentences to lower case.

I have created a function called text\_transform which will apply all the preprocessing for a given string. That function is shown in figure 2.2.2.2



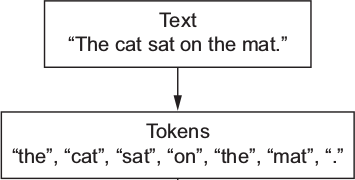
Fig 2.2.2.2

**Let is understand all the pre-processing techniques.**

**2.2.3 Tokenization:**

Tokenization is the process of converting text into tokens before transforming it into vectors. It is also easier to filter out unnecessary tokens. For example, a document into paragraphs or sentences into words. In this case we are tokenising the reviews into words

It is one of the most foundational NLP tasks and a difficult one, because every language has its own grammatical constructs, which are often difficult to write down as rules.

We have a bunch of text, and we want to computer to work on all the text, so why do we need to break the text into small tokens.

**2.2.4 Removing punctuations and special symbols:**

Apart from the considered set of emoticons punctuations and symbols like &,\,; are removed.

**2.2.5 Stop words removal:**

Stop words are the most commonly occuring words which are not relevant in the context of the data and do not contribute any deeper meaning to the phrase. In this case contain no sentiment.

"This is a sample sentence, showing off the stop words filtration."

['This', 'is', 'a', 'sample', 'sentence', ',', 'showing', 'off', 'the', 'stop', 'words', 'filtration', '.']

**After stop words removal:**

['This', 'sample', 'sentence', ',', 'showing', 'stop', 'words', 'filtration', '.']

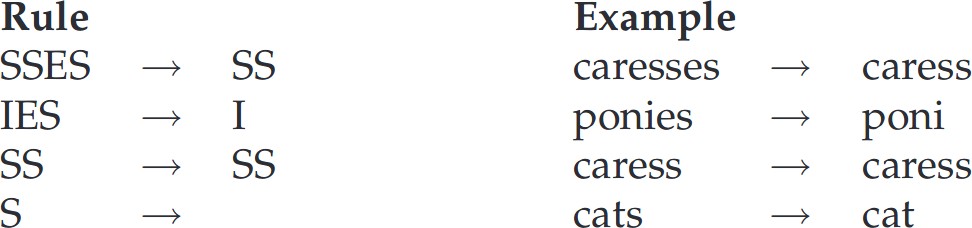
For performing this operation we have a library in python called stop words and inside it has a function called get\_stopwords() which return the stop words in list format.

**2.2.6 Stemming and Lemmatization:**

Sentences are always narrated in tenses,singular and plural forms making most words accompany with -ing,-ed,es and ies. Therefore,extracting the root word will suffice to identify sentiment behind the message.

Base forms are the skeleton for grammar stemming and lemmatization reduces inflectional forms and derivational forms to common base forms .

Example: Cats is reduced to cat ,ponies is reduced to poni.



Stemming is a crude way of reducing terms to their root, by just defining rules of chopping off some characters at the end of the word, and hopefully, gets good results most of the time. The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.With that being said, stemming/lemmatizing helps us reduce the number of overall terms to certain “root” terms.

**2.2.7 Classification of Messages:**

There are some words common on spam messages and non-spam messages as you can see in figure 2.2.7.1 and 2.2.7.2. On the basis of these Machine learning algorithm classify between spam and non-spam messages

The classification of messages in spam classifier is done by using machine learning algorithms such as Support Vector Machine (SVM) or Naive Bayes Classification Model to automatically categorize messages into spam or non-spam classes. These algorithms use Natural Language Processing (NLP) to analyze the text data of messages and assign them a class label.

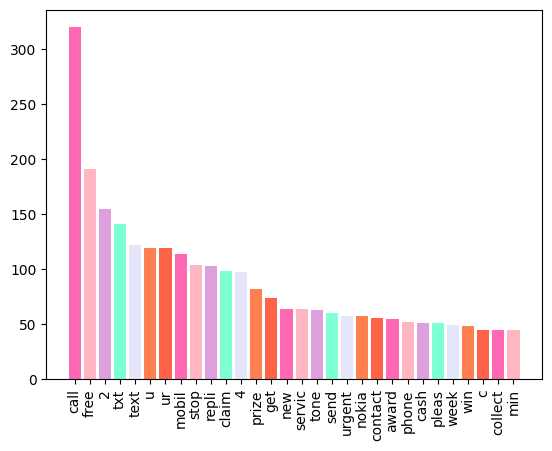
****

Fig 2.2.7.1

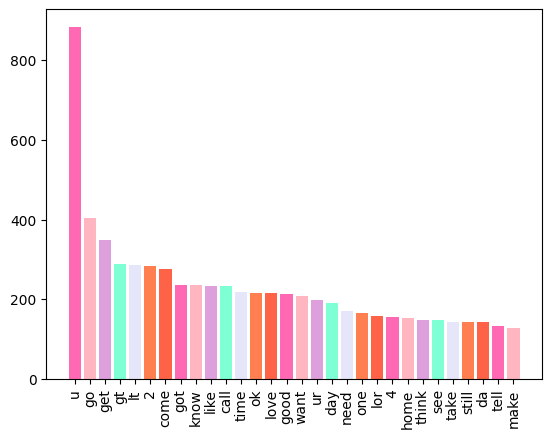


Fig 2.2.7.2

**2.3 Visual Representation**

**Pie chart:**

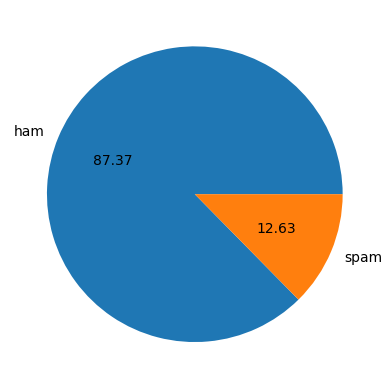


Fig 2.3.1

**Graph:**

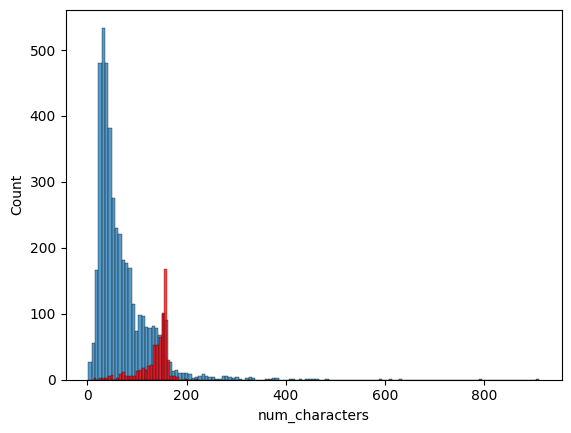


Fig 2.3.2

**Word-Cloud of Spam Messages:**

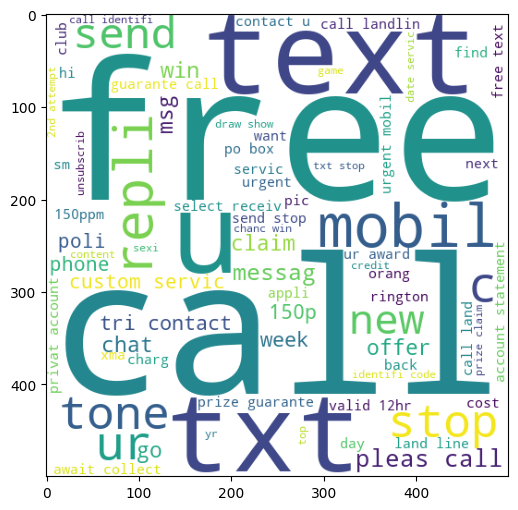
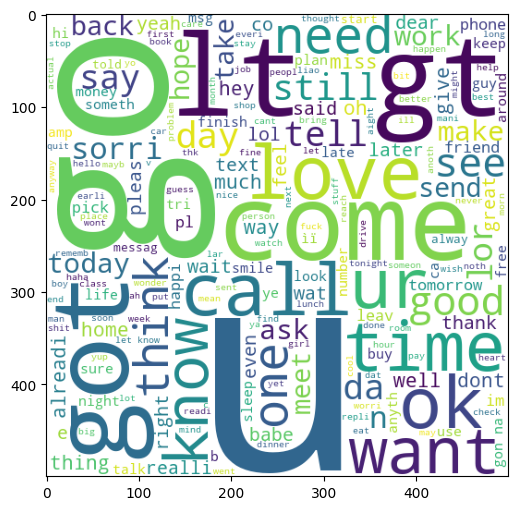


Fig 2.3.3

**Word-Cloud of ham messages :**



**Fig 2.3.4**

**SNAPSHOT OF PROJECT**

**3.1 WEB PAGE**

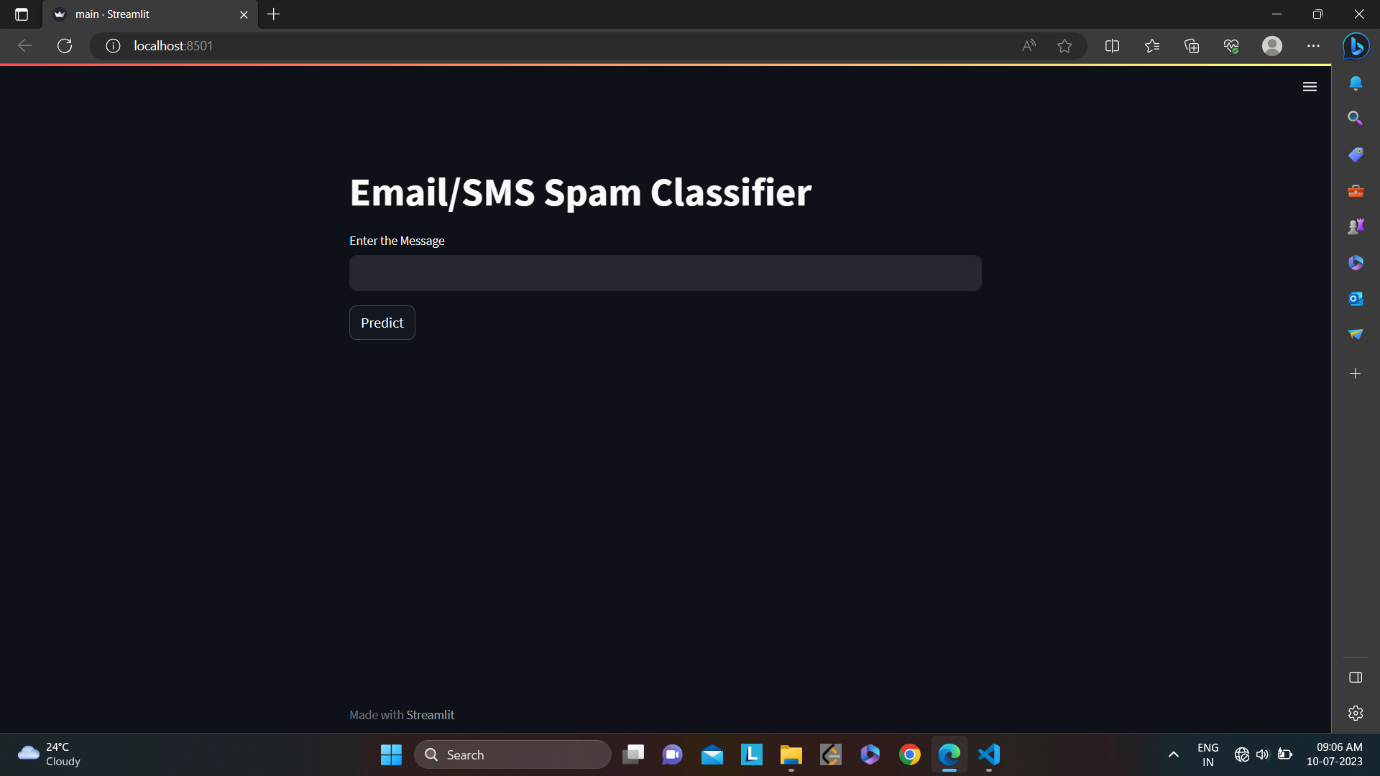
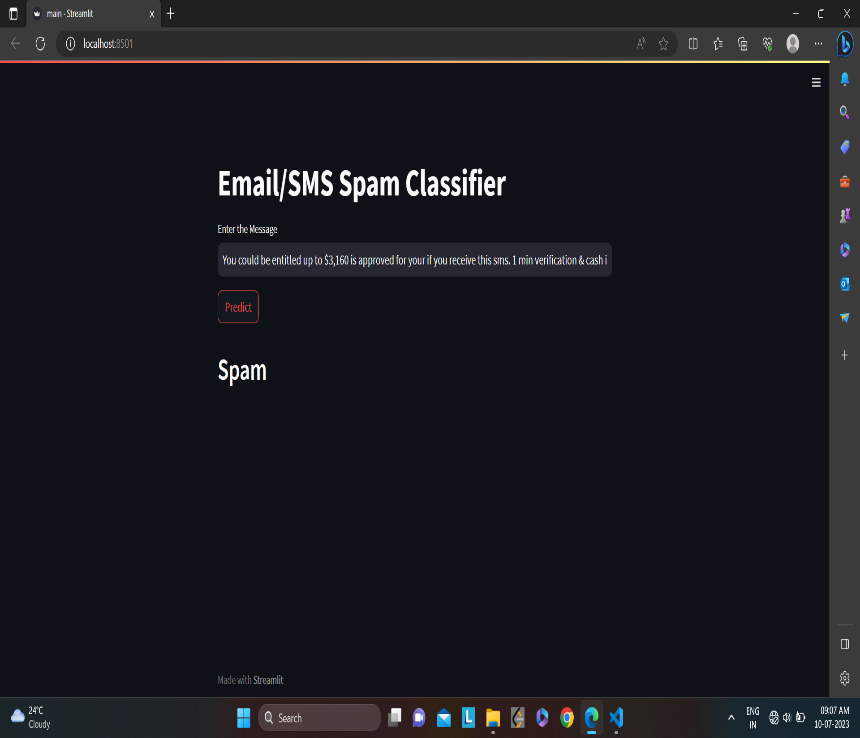
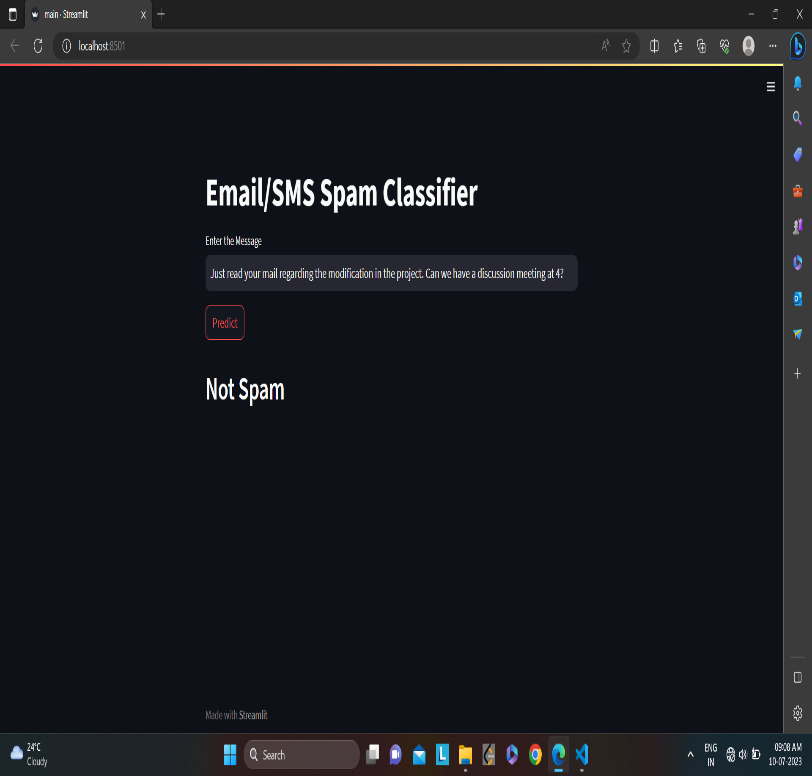


Fig 3.1.1

******3.2 Result**

**4. CONCLUSION**

Email classification is a technique used to classify emails into different categories such as spam or ham. Naive Bayes classifier technique has become a very popular method in mail filtering Email. Every word has certain probability of occurring in spam or ham email in its database. However, it is not enough to evaluate the performance based on the accuracy alone since the dataset is imbalanced. After some examinations, NB algorithm still manages to provide good precision and f measure with 0.98 of precision while 0.97 for f-measure. Different algorithms will provide different performances and results based on the features used. For future works, adding more features such as message lengths might help the classifiers to train data better and give better performance.

**5. REFERENCE**

1. [(76) Bayes Theorem Explained with Solved Example in Hindi ll Machine Learning Course - YouTube](https://www.youtube.com/watch?v=Fv_LGQKgWi0) [SMS Spam Collection Dataset | Kaggle](https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset)
2. <https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/>
3. https://www.bing.com/ck/a?!&&p=7f26e43864db015bJmltdHM9MTY4ODk0NzIwMCZpZ3VpZD0yYjE2YzVjMy1mOGMzLTYxOWItMGZhMy1kNzBmZjk3MTYwNzAmaW5zaWQ9NTI0Ng&ptn=3&hsh=3&fclid=2b16c5c3-f8c3-619b-0fa3-d70ff9716070&psq=machine+learning+classification+algorithms&u=a1aHR0cHM6Ly93d3cuamF2YXRwb2ludC5jb20vY2xhc3NpZmljYXRpb24tYWxnb3JpdGhtLWluLW1hY2hpbmUtbGVhcm5pbmc&ntb=1
4. Machine Learning Tom M. Mitchell