

PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and
Technology

(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of a Bachelor of Technology Degree in Information Technology

By

Muthu Yashwanth.S

REG. NO. 40120060



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

April 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI

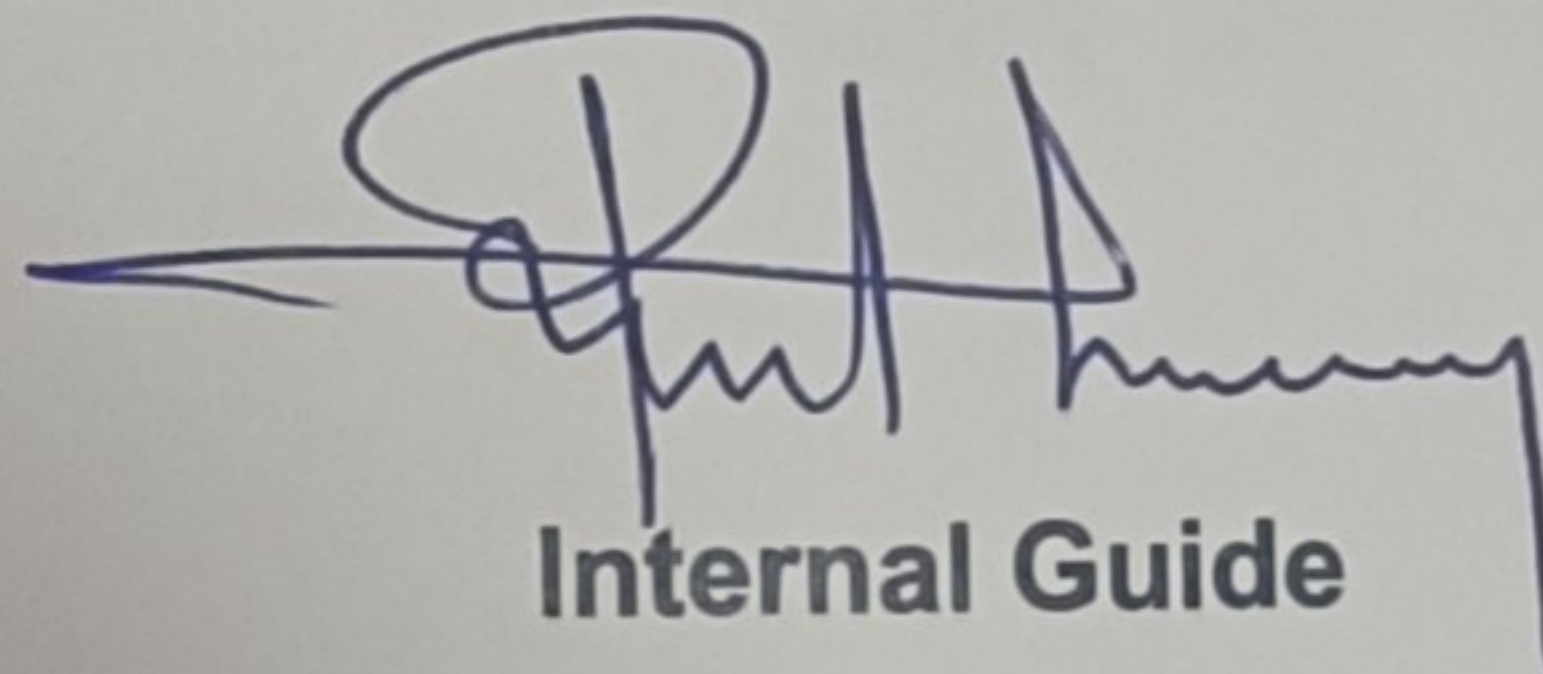
CHENNAI- 600119

www.sathyabama.ac.in

SCHOOL OF COMPUTING

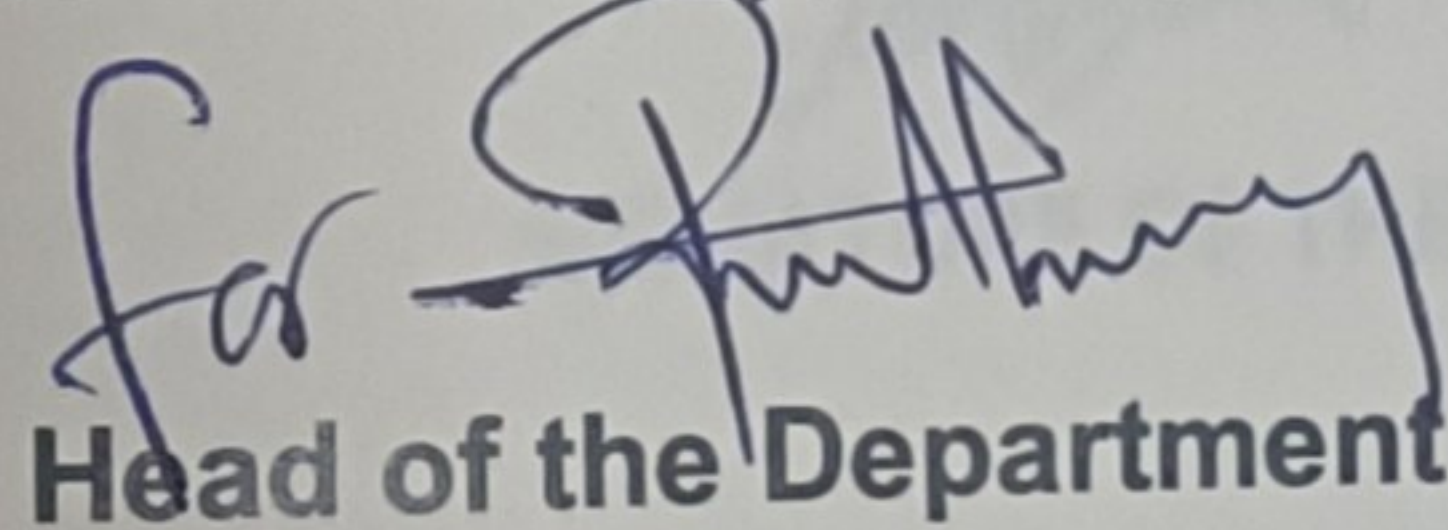
BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Muthu Yashwanth.S (Reg.No:40120060)** who carried out the project entitled "**Email Spam Classification**" under my supervision from January 2023 to April 2023.



Internal Guide

Dr. S.Revathy, M.E., Ph.D.,

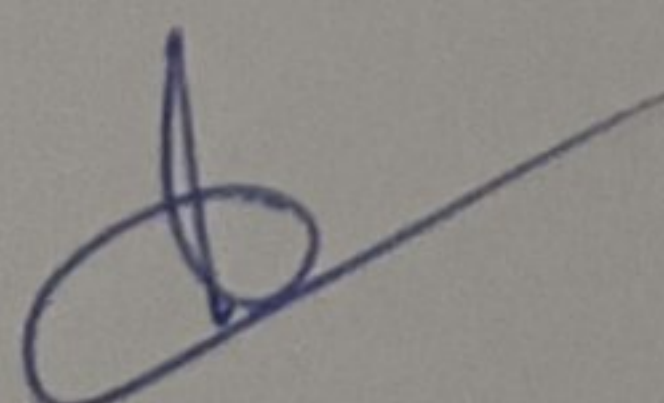


Head of the Department

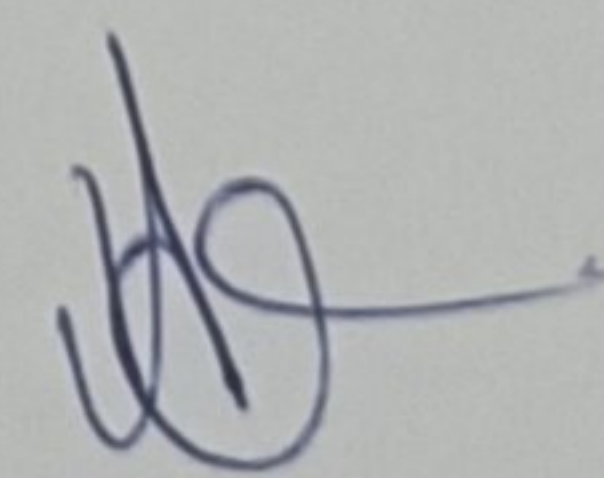
Dr. R. Subhashini, M.E., Ph.D.,

Submitted for Viva voce Examination held on

28-04-2023



Internal Examiner



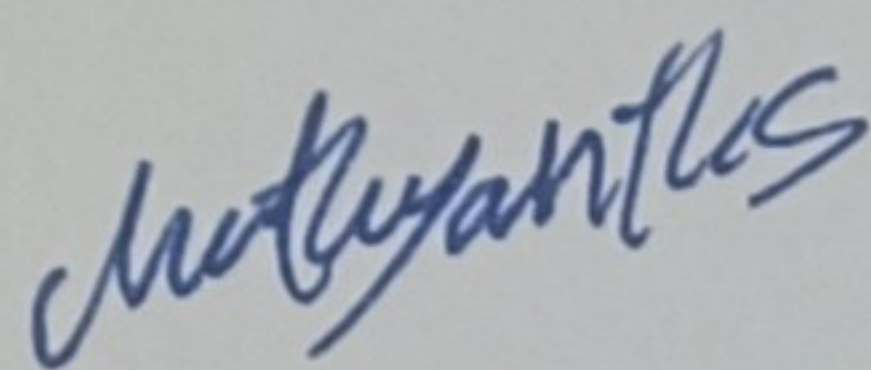
External Examiner

DECLARATION

I, **Muthu Yashwanth.S** hereby declare that the project report entitled **Email Spam Classification** done by me under the guidance of **Dr.S.Revathy** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Information Technology.

DATE: 28-04-2023

PLACE: CHENNAI



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to Dr. T. Sasikala M.E., Ph.D., Dean, School of Computing, and Dr. R. Subhashini M.E., Ph.D., Head of the Department of Information Technology for providing me with necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr. S.Revathy.ME.PhD., for his valuable guidance, suggestions, and constant encouragement that paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Information Technology who were helpful in many ways for the completion of the project.

ABSTRACT

Machine learning methods of recent are being used to successfully detect and filter spam emails. We present a systematic review of some of the popular machine learning-based email spam filtering approaches. Our review covers a survey of the important concepts, attempts, efficiency, and research trend in spam filtering. Naive Bayes is very popular in commercial and open-source anti-spam e-mail filters. There are, however, several forms of Naive Bayes, something the anti-spam literature does not always acknowledge. We discuss five different versions of Naive Bayes and compare them on six new, nonencoded datasets, that contain ham messages of particular Enron users and fresh spam messages. The new datasets, which we make publicly available, are more realistic than previous comparable benchmarks, because they maintain the temporal order of the messages in the two categories, and they emulate the varying proportion of spam and ham messages that users receive over time. In this paper, we have discovered various aspects of the Naïve Bayes Classifier and smoothing techniques for the extraction of useful data along with our research criteria.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No
	ACKNOWLEDGEMENT	i
	ABSTRACT	ii
	TABLE OF CONTENTS	iii
	LIST OF FIGURES	iv
1.	INTRODUCTION	1
	1.1 GENERAL	1
	1.2 MACHINE LEARNING IN E-MAIL CLASSIFICATION	2
	1.3 Different types of machine learning methods	3
	1.4 Naïve Bayes Algorithm	4
	1.5 Aim	4
	1.6 Objective	5
	1.7 Scope	
	1.8 Problem Statement	
2.	LITERATURE SURVEY	5
	2.1 GENERAL	5
	2.2 LITERATURE REVIEW	5
3.	METHODOLOGY AND ALGORITHMS	8
	3.1 GENERAL	8
	3.2 HARDWARE REQUIREMENTS	8
	3.3 SOFTWARE REQUIREMENTS	8
	3.4 TOOLS AND LIBRARIES	8
	3.4.1 Pandas	9
	3.4.2 Python	9
	3.4.3 Numpy	9
	3.4.4 SK-learn	10
	3.5 Spam Dataset	10
	3.6 Algorithm Used	11

4		RESULT AND DISCUSSION	22
	4.1	Result	28
5		CONCLUSION AND FUTURE SCOPE	23
		Conclusion	23
		Future Scope	23
		APPENDIX	24
		REFERENCE	33

LIST OF FIGURES

FIGURE No	FIGURE NAME	PAGE No
Fig 1.1	Spam Architecture	2
Fig 1.1.1	Naïve Bayes Algorithm	4
Fig 3.6.1	Importing modules	11
Fig 3.6.2	Reading the Dataset	11
Fig 3.6.3	Calling the Dataset	12
Fig 3.6.4	Map Function	12
Fig 3.6.5	Count Vectorizer	13
Fig 3.6.6	Xshape,Yshape	14
Fig 3.6.7	Cv.fit_transform	14
Fig 3.6.8	Import and Train_Test	15
Fig 3.6.9	MultinomialNB	15
Fig 3.6.10	Importing pickle	16
Fig 3.6.11	Pickle.dump	16
Fig 3.6.12	Model.predict	17
Fig 4.1.1	Importing pickle	18
Fig 4.1.2	Model.predict	19
Fig 4.1.3	New model.predict	19
Fig 4.1.4	Data class	20
Fig 4.1.5	New Data class	20
Fig 4.1.6	Sublime text	22
Fig 4.1.7	Opening cmd	21
Fig 4.1.8	Final Result	22

TRAINING CERTIFICATE



Elite

NPTEL Online Certification

(Funded by the MoE, Govt. of India)



This certificate is awarded to

MUTHU YASHWANTH

for successfully completing the course

Foundation of Cloud IoT Edge ML

with a consolidated score of **62** %

Online Assignments	23.13/25	Proctored Exam	38.46/75
--------------------	----------	----------------	----------

Total number of candidates certified in this course: **1649**

Prof. B. V. Ratish Kumar
Chairman, Centre for Continuing Education
IIT Kanpur

Feb-Apr 2023
(8 week course)

Prof. Satyaki Roy
NPTEL Coordinator
IIT Kanpur



Indian Institute of Technology Kanpur



Roll No: NPTEL23CS65S33366176

To validate the certificate



No. of credits recommended: 2 or 3

CHAPTER 1

INTRODUCTION

1.1 GENERAL

A Classification system is a type of information filtering system that attempts to assume the priorities of a user to offer recommendations based on those priorities. This type of system is also known as a collaborative filtering system. A system like this could. In recent years, the internet has become an integral part of life. With the increased use of the internet, the number of email users is increasing day by day. This increasing use of email has created problems caused by unsolicited bulk email messages commonly referred to as Spam. An email has now become one of the best ways for advertisements due to which spam emails are generated. Spam emails are emails that the receiver does not wish to receive. a large number of identical messages are sent to several recipients of the email. Spam usually arises as a result of giving out our email addresses on an unauthorized or unscrupulous website. There are many of the effects of Spam .Fills our Inbox with number of ridiculous emails .Degrades our Internet speed to a great extent .Steals useful information like our details on you Contact list .Alters your search results on any computer program .Spam is a huge waste of everybody's time and can quickly become very frustrating if you receive large amounts of it .Identifying these spammers and the spam content is a laborious task .even though extensive number of studies have been done, yet so far the methods set forth still scarcely distinguish spam surveys, and none of them demonstrate the benefits of each removed element compose .In spite of increasing network communication and wasting lot of memory space ,spam messages are also used for some attack . Spam emails, also known as non-self,are unsolicited commercial or malicious emails, sent to affect either a single individual or a corporation or a bunch of people. Besides advertising,these may contain links to phishing or malware hosting websites found out to steal confidential information. to solve this problem the different spam filtering techniques are used.

MACHINE LEARNING IN E-MAIL CLASSIFICATION:

Machine learning field is a subfield from the broad field of artificial intelligence, this aims to make machines able to learn like human. Learning here means understood, observe and represent information about some statistical phenomenon. In unsupervised learning one tries to uncover hidden regularities (clusters) or to detect anomalies in the data like spam messages or network intrusion. In e-mail filtering task some features could be the bag of words or the subject line analysis. Thus, the input to e-mail classification task can be viewed as a two dimensional matrix, whose axes are the messages and the features. E-mail classification tasks are often divided into several sub-tasks. First, Data collection and representation are mostly problem specific (i.e. e-mail messages), second, e-mail feature selection and feature reduction attempt to reduce the dimensionality (i.e. the number of features) for the remaining steps of the task. Finally, the e-mail classification phase of the process finds the actual mapping between training

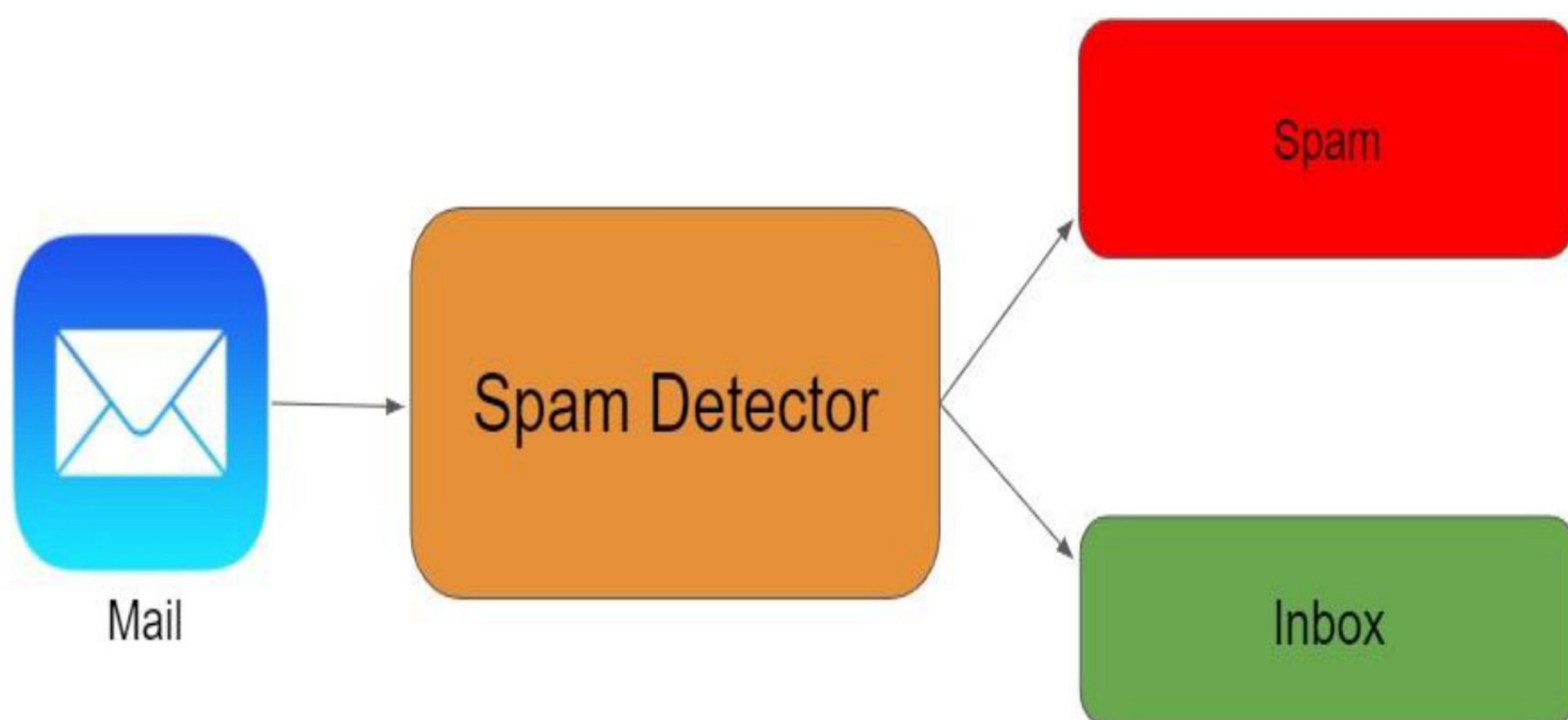


Fig 1.1 Spam Architecture

Different types of machine learning methods

- Naïve Bayes
- K-nearest neighbour classifier method
- Artificial Neural Networks classifier method
- Support Vector Machines classifier method
- Artificial Immune System classifier method
- Rough sets classifier method

1.1.1 Naïve Bayes Algorithm:

A naive Bayes classifier is a simple probabilistic classifier with strong assumptions of independence. Simply put, a naive bayes classifier assumes that the presence / absence of a particular property of a class is not related to the presence / absence of any other feature, considering the class variable as a function of the Class Probability Model, Trained in a supervised learning environment. An advantage of the naive bayes classification is that it requires only a small amount of training data to estimate the parameters required for classification. The Bayesian classification assumes that the data belongs to a particular class. We then calculate the probability that the assumption is true. Bayesian classmates are statistical classifiers, that is, they can predict probabilities of class membership, such as the probability that a given test belongs to a particular class.

$$P(y|x) = \frac{p(x|y) \cdot p(y)}{p(x)}$$

Where, x is any feature vector set ($x_1, x_2, x_3, \dots, x_n$) and y are the class variables with possible outcomes ($y_1, y_2, y_3, \dots, y_n$). $P(y|x)$ stands for posterior probability, $P(x|y)$ is any particular class on which $P(y|x)$ is dependent. $P(x)$ is evidence depending on the known feature variables, $P(y)$ is the prior probability. So, Naïve Bayes classification model consists of set of probabilities of prior probability, class conditional probability and posterior probability.

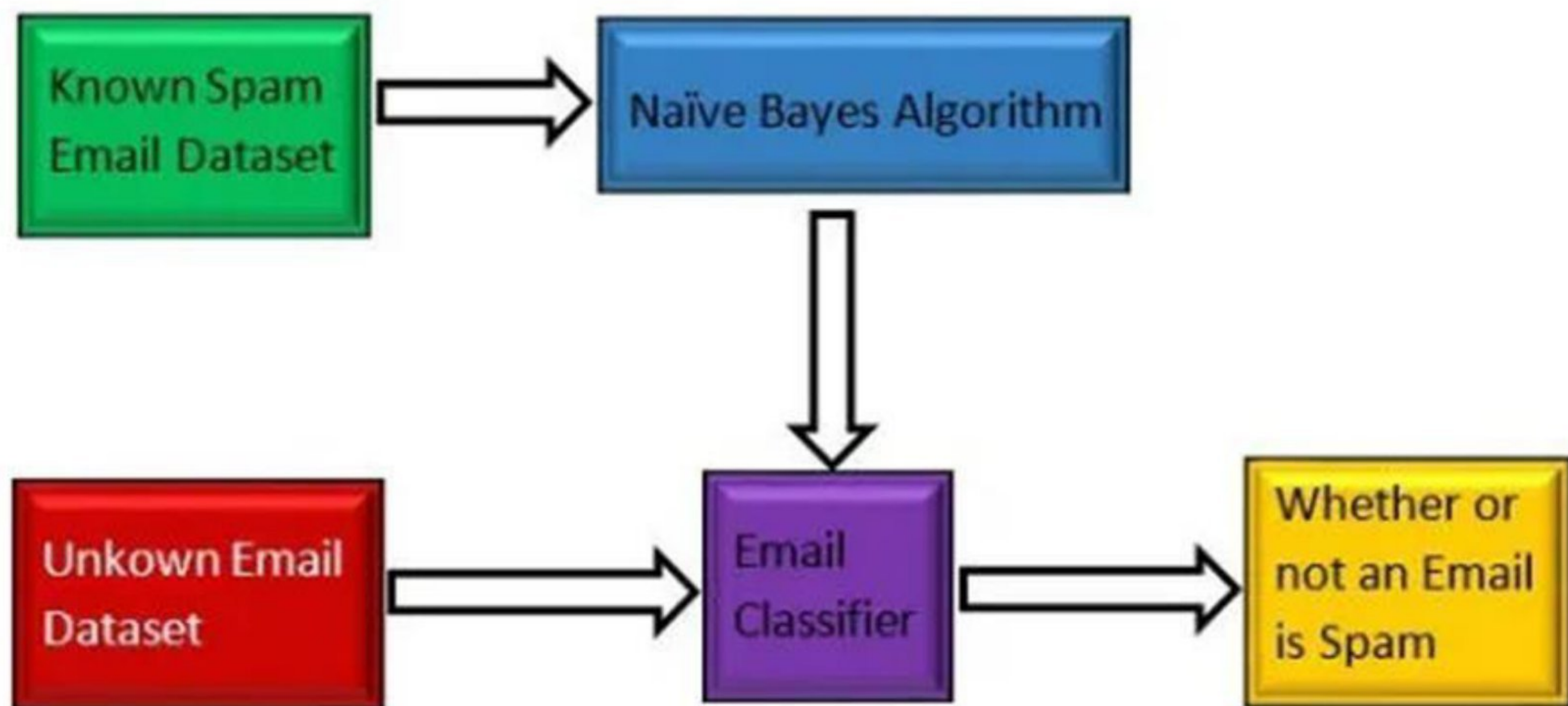


Fig 1.1.1 Naïve Bayes Algorithm

AIM:

This project aims to use predict email spam or ham by using the Naïve Bayes classifier. using Naïve Bayes we can get very accurate and it is a supervised Machine learning.

SCOPE:

Naïve Bayes is very accurate; however, it is unable to correctly classify emails when they contain leetspeak or diacritics. Thus in this proposal, we implemented a novel algorithm for enhancing the accuracy of the Naive Bayes Spam Filter so that it can detect text modifications and correctly classify the email as spam or ham.

PROBLEM STATEMENT: The project proposes “Spam Message Filtering using Intelligent Text Modification method”. This project predicts spam messages more accurately than the existing projects. This project uses Naïve Bayes algorithm to modify the text.

CHAPTER 2

LITERATURE SURVEY

GENERAL

In recent years, there has been an increase in the number of academic papers and reviews devoted to the topic of computerized Email classification systems. The following is a selection of the literature reviews that were taken into consideration for this project.

LITERATURE REVIEW

[N.F. Rusland] [N.Wahid] [S.Kasim] [H.Hafit](2017) Performed analysis on email classification on two different dataset by using Naïve Bayes algorithm based on the Accuracy, Precision, F-Measure and Recall. The process was divided into 3 steps. First step is data pre- processing in which all articles, conjunctions and undesired words is removed from the text. Next is the feature extraction followed by training of the Naive Bayes model. Based on the training of the model, it predicts whether the given text is ham or spam. By using Spam data Dataset, the author achieved an accuracy of 91.13% and for the other Spam Base dataset, accuracy achieved was 88%.

[S. Ajaz] [M. T. Nafis] [V.Sharma] (2017) They collected email dataset from the online available websites and used Naïve Bayes for filtering of emails. He proposed a hybrid approach using secure hash method and Naive Bayes to filter email data but could not provide information regarding the misuse of storage resources and network bandwidth. By using Secure Hash Algorithm, the email is considered as a message M due to a generated function. The message M is further classified into S and L where L stands for ham email or genuine email and on the other hand S stands for spam email.

Almeida,tiago. Almeida, Jurandy.Yamakami, Akebo (2011) E-mail spam has become an increasingly important problem with a big economic impact in society. Fortunately, there are different approaches allowing to automatically detect and remove most of those messages, and the best-known techniques are based on Bayesian decision theory. However, such probabilistic approaches often suffer from a well-known difficulty: the high dimensionality of the feature space. Many term-selection methods have been proposed for avoiding the curse of dimensionality. Nevertheless, it is still unclear how the performance of Naive Bayes spam filters depends on the scheme applied for reducing the dimensionality of the feature space. In this paper, we study the performance of many term-selection techniques with several different models of Naive Bayes spam filters. Our experiments were diligently designed to ensure statistically sound results. Moreover, we perform an analysis concerning the measurements usually employed to evaluate the quality of spam filters. Finally, we also investigate the benefits of using the Matthews correlation coefficient as a measure of performance.

M. N. Marsono, M. W. El-Kharashi, and F. Gebali (2017) The increasing volume of unsolicited bulk e-mail (also known as spam) has generated a need for reliable anti-spam filters. Machine learning techniques now days used to automatically filter the spam e-mail in a very successful rate. In this paper we review some of the most popular machine learning methods (Bayesian classification, k-NN, ANNs, SVMs, Artificial immune system and Rough sets) and of their applicability to the problem of spam Email classification. Descriptions of the algorithms are presented, and the comparison of their performance on the SpamAssassin spam corpus is presented.

Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, Opeyemi Emmanuel Ajibuwa (2019) The upsurge in the volume of unwanted emails called spam has created an intense need for the development of more dependable and robust antispam filters. Machine learning methods of recent are being used to successfully detect and filter spam emails. We present a systematic review of some of the popular machine learning based email spam filtering approaches. Our review covers survey of the important concepts, attempts, efficiency, and the research trend in spam filtering. The preliminary discussion in the study background examines the applications of machine learning techniques to the email spam filtering process of the leading internet service providers (ISPs) like Gmail, Yahoo and Outlook emails spam filters. Discussion on general email spam filtering process, and the various efforts by different researchers in combating spam through the use machine learning techniques was done. Our review compares the strengths and drawbacks of existing machine learning approaches and the open research problems in spam filtering. We recommended deep leaning and deep adversarial learning as the future techniques that can effectively handle the menace of spam emails.

John Wiley & Sons (2009) Email spam has become a major problem for Internet users and providers. One major obstacle to its eradication is that the potential solutions need to ensure a very low false-positive rate, which tends to be difficult in practice. We address the problem of low-FPR classification in the context of naive Bayes, which represents one of the most popular machine learning models applied in the spam filtering domain. Drawing from the recent extensions, we propose a new term weight aggregation function, which leads to markedly better results than the standard alternatives. We identify short instances as ones with disproportionally poor performance and counter this behavior with a collaborative filtering-based feature augmentation. Finally, we propose a tree-based classifier

cascade for which decision thresholds of the leaf nodes are jointly optimized for the best overall performance.

These improvements, both individually and in aggregate, lead to substantially better detection rate of precision when compared with some of the best variants of naive Bayes.

CHAPTER 3

METHODOLOGY AND ALGORITHMS

GENERAL

Research methodology refers to the exact techniques or strategies used in order to obtain, select, analyze, and evaluate material relevant to a given subject issue. If the methodology portion of a research paper is written correctly, it will enable the reader to assess the general dependability and durability of the study that is being addressed in a way that is objective. The following answers may be found in the section on approaches for each of the two primary questions that were given in the introduction: Which methods and procedures were used when it came to the gathering and creation of the information? What kind of analysis was carried out on it? Based on the objective of the study the methodology for the research is designed in form of a flow chart.

HARDWARE REQUIREMENTS

- Microsoft Windows 7/8/10 64-bit
- 11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz
- Minimum of 6gb,8gb RAM

SOFTWARE REQUIREMENTS

- Jupyter Notebook
- Sublime Text

TOOLS AND LIBRARIES

- Python – 3. x
- Pandas – 1.2.4
- Scikit-learn – 0.24.1.
- Spam DataSet.

Pandas

Pandas is an open-source library that was developed primarily for working with relational or labeled data easily and naturally. Its primary purpose is to facilitate this type of work. It offers a wide variety of data structures and procedures for the purpose of processing numerical data and time series. The NumPy library serves as the foundation upon which this other library was constructed. Pandas is incredibly quick, and it provides users with great performance and productivity levels. Pandas is a Python module that is utilized for data set manipulation and management. It possesses capabilities for analyzing, cleaning, investigating, and modifying data, among other things.

Python

Over the past few years, Python has rapidly risen in popularity to become one of the most widely used programming languages in the world. It finds application in everything from machine learning to the construction of websites and the testing of software. It is accessible to users who are not developers as well as those who are.

Python is a high-level programming language that can be interpreted, used in interactive scripting, and object-oriented. The programming language Python was developed with readability in mind. It makes frequent use of English terms, whereas other languages make extensive use of punctuation, and it has a much-reduced number of syntactical formations in comparison to other languages.

Numpy

Numpy stands for Numerical Python. It refers to a library that consists of multidimensional array objects as well as a collection of innumerable methods for processing those arrays. Operations on arrays, both mathematical and logical, can be carried out using this method. Numpy works with numerical data, whereas pandas work with tabular data.

SKlearn

When it comes to machine learning in Python, the Scikit-learn (Sklearn) library is by far the most helpful and reliable option. It offers a standardized Python interface for accessing a variety of powerful tools for machine learning and statistical modelling, such as classification, regression, clustering, and dimensionality reduction. These tools can be used to analyze data. NumPy, SciPy, and Matplotlib were utilized in the construction of this package, which was primarily created in Python.

It was initially developed by David Cournapeau in 2007 as a project for the Google Summer of Code under the name scikits. learn. The project's original name was also scikits.learn. Later, in 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, and Vincent Michel, from FIRCA (French Institute for Research in Computer Science and Automation), took this project to another level and released the first public release (v0.1 beta) on February 1st, 2010. This was in 2010.

SPAM DATASET

A dataset in Python has a lot of significance and is mostly used for dealing with a huge amount of data. Python datasets consist of dataset object, which in turn comprises metadata as part of the dataset. Querying to these datasets may include dataset objects to return the required index based on rows and columns. The dataset object appears when the data gets loaded initially that also comprise the metadata consisting of other important information. A Spam dataset is very popular dataset for Spam filter for emails.

ALGORITHMS USED

I am importing the following modules.

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

In [2]: data=pd.read_csv("spam.csv", encoding="latin-1")
```

Fig 3.6.1 Importing modules

- import pandas as pd

pandas is a software library used in python programming for data manipulation and analysis.

- Reading the Dataset

In this project I used Spam dataset and named it as “Spam.CSV” ,this files contain one message per line. Each line is composed by two columns: v1 contains the label (ham or spam) and v2 contains the raw text (message).

```
In [3]: data.head(5)
```

	class	message	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

Fig 3.6.2 Reading the Dataset

- Calling the Dataset

Next we are calling the dataset of first 5 group which contains the following (Class ,message, unnamed:2,unnamed:3,unnamed:4)

```
In [5]: data.drop(['Unnamed: 2','Unnamed: 3','Unnamed: 4'],axis=1, inplace=True)

In [6]: data.head()
```

	class	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

Fig 3.6.3 Calling the Dataset

- Dropping the column

Here im dropping the column ('unnamed:2','unnamed:3','unnamed:4')
And after that the whole column will be shown as Class name,message.

- Map Function

Next I'm converting the Spam and ham class as 0 and 1.so map function is used to read all Spam and ham class from the rows.

```
In [7]: data['class']=data['class'].map({'ham':0,'spam':1})

In [8]: data.head()
```

	class	message
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

Fig3.6.4 Map Function

- CountVectorizer

CountVectorizer is used to convert a collection of text documents to a vector of term/token counts. here CountVectorizer is used to make these message as sparse matrix.

```
In [9]: from sklearn.feature_extraction.text import CountVectorizer

In [10]: cv=CountVectorizer()

In [11]: x=data['message']
          y=data['class']
```

Fig 3.6.5 Count Vectorizer

- X. Shape, Y. Shape

The shape is a tuple of integers. These numbers denote the lengths of the corresponding array dimension. In other words: The "shape" of an array is a tuple with the number of elements per axis (dimension). In our model, the shape of X is (5572), i.e. we have 5572 lines and shape of Y is (5572) the shape of X and Y axis are equal.

```
In [12]: x.shape
(5572,)
```

```
In [13]: y.shape
(5572,)
```

Fig 3.6.6 Xshape,Yshape

- `cv.fit_transform`

The `fit_transform()` method is basically the combination of the `fit` method and the `transform` method. we need vectorizer format it will us object.

```
In [14]: x=cv.fit_transform(x)
```

```
In [15]: x
<5572x8672 sparse matrix of type '<class 'numpy.int64'>'
with 73916 stored elements in Compressed Sparse Row format>
```

Fig 3.6.7 Cv.fit_transform

- Import And Train_Test

Here we are importing Sklearn library and train refers to measure the accuracy of our model. x_{train} :The training part of the first sequence (x) x_{test} :The test part of the first sequence (x) y_{train} :The training part of the second sequence (y) y_{test} :The test part of the second sequence (y).

```
In [16]: from sklearn.model_selection import train_test_split

In [17]: x_train, x_test, y_train, y_test=train_test_split(x, y, test_size=0.2)

In [18]: x_train.shape

(4457, 8672)
```

Fig3.6.8 Import and Train_Test

- MultinomialNB

From sklearn we are importing the MultinomialNB. The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts.

```
In [19]: from sklearn.naive_bayes import MultinomialNB

In [20]: model=MultinomialNB()

In [21]: model.fit(x_train, y_train)

In [22]: result=model.score(x_test, y_test)

In [23]: result=result*100
```

Fig3.6.9 MultinomialNB

- Import Pickle

Now we are importing the pickle. Python's Pickle module is a popular format used to serialize and deserialize data types. This format is native to Python, meaning Pickle objects cannot be loaded using any other programming language

```
In [24]: result

98.20627802690582

In [25]: import pickle
```

Fig3.6.10 Importing pickle

- Pickle.dump

To save the ML model using Pickle all we need to do is pass the model object into the dump() function of Pickle. This will serialize the object and convert it into a “byte stream” that we can save as a file called model. pkl .

```
In [26]: pickle.dump(model, open("spam.pkl", "wb"))

In [27]: pickle.dump(cv, open("vectorizer.pkl", "wb"))

In [28]: clf=pickle.load(open("spam.pkl", "rb"))

In [29]: clf
```



The output of the Jupyter Notebook cell shows a `MultinomialNB` object. A tooltip is visible over the object, displaying the text `MultinomialNB` and `fit()`.

Fig3.6.11 pickle.dump

- Cv.transform

In this step we are using Cv.transform (Counter Vectorizer Transform).The transform() method allows you to execute a function for each value of the DataFrame.

- Model.Predict

Finally we are predict our model by using Model.Predict predict()function is used to predict outcomes using the model.in model.predict we are using the vectorization inside, then we typing print result .

```
In [30]: msg="Hello There"
data=msg
vect=cv.transform(data).toarray()
result=model.predict(vect)
print(result)
```

```
[0]
```

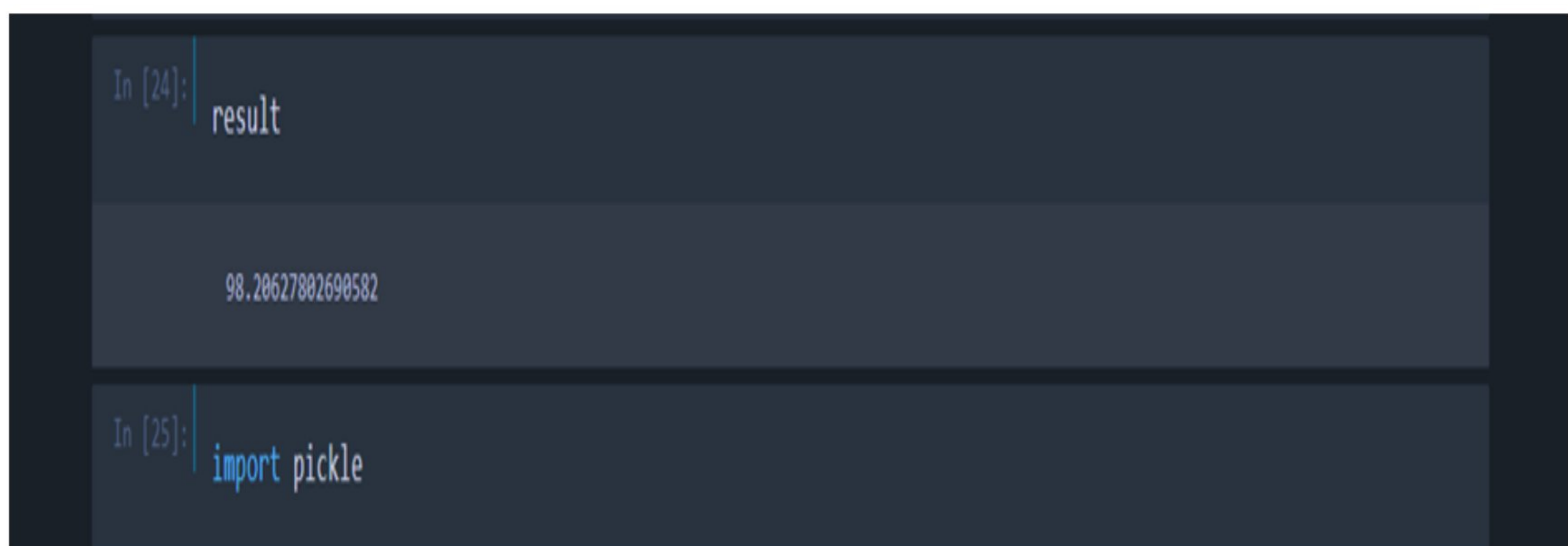
Fig3.6.12 Model.Predict

CHAPTER 4

RESULT AND DISCUSSION

4.1 RESULT:

After importing the Libraries MultinomialNB im importing the pickle.To save the ML model using Pickle all we need to do is pass the model object into the dump() function of Pickle. This will serialize the object and convert it into a “byte stream” that we can save as a file called model. Pkl .



```
In [24]: result
98.20627802690582

In [25]: import pickle
```

Fig4.1.1 Importing pickle

we are using Cv.transform (Counter Vectorizer Transform).The transform() method allows you to execute a function for each value of the DataFrame. Finally we are predict our model by using Model.Predict. predict()function is used to predict outcomes using the model.in model.predict we are using the vectorization inside, then we typing print result .

```
In [30]: msg="Hello There"
        data=[msg]
        vect=cv.transform(data).toarray()
        result=model.predict(vect)
        print(result)

[0]
```

Fig4.1.2 model.predict

```
In [61]: msg="You have won $250 in your lottery"
        data=[msg]
        vect=cv.transform(data).toarray()
        result=model.predict(vect)
        print(result)

[1]
```

Fig4.1.3 New model.predict

At the end we created a model to predict the email by classifying the email by ham or spam. we give in the msg="hello There" as input and model uses the naïve bayes

classification and we can get the result as {1} shows ham mail, and same we trained the model by naïve bayes and we give the input in msg=” You have won \$250 in your lottery” and we see that the model we trained give us output as {1} shows it is spam mail.

```
In [6]: data.head()
```

	class	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

```
In [7]: data['class']=data['class'].map({'ham':0,'spam':1})
```

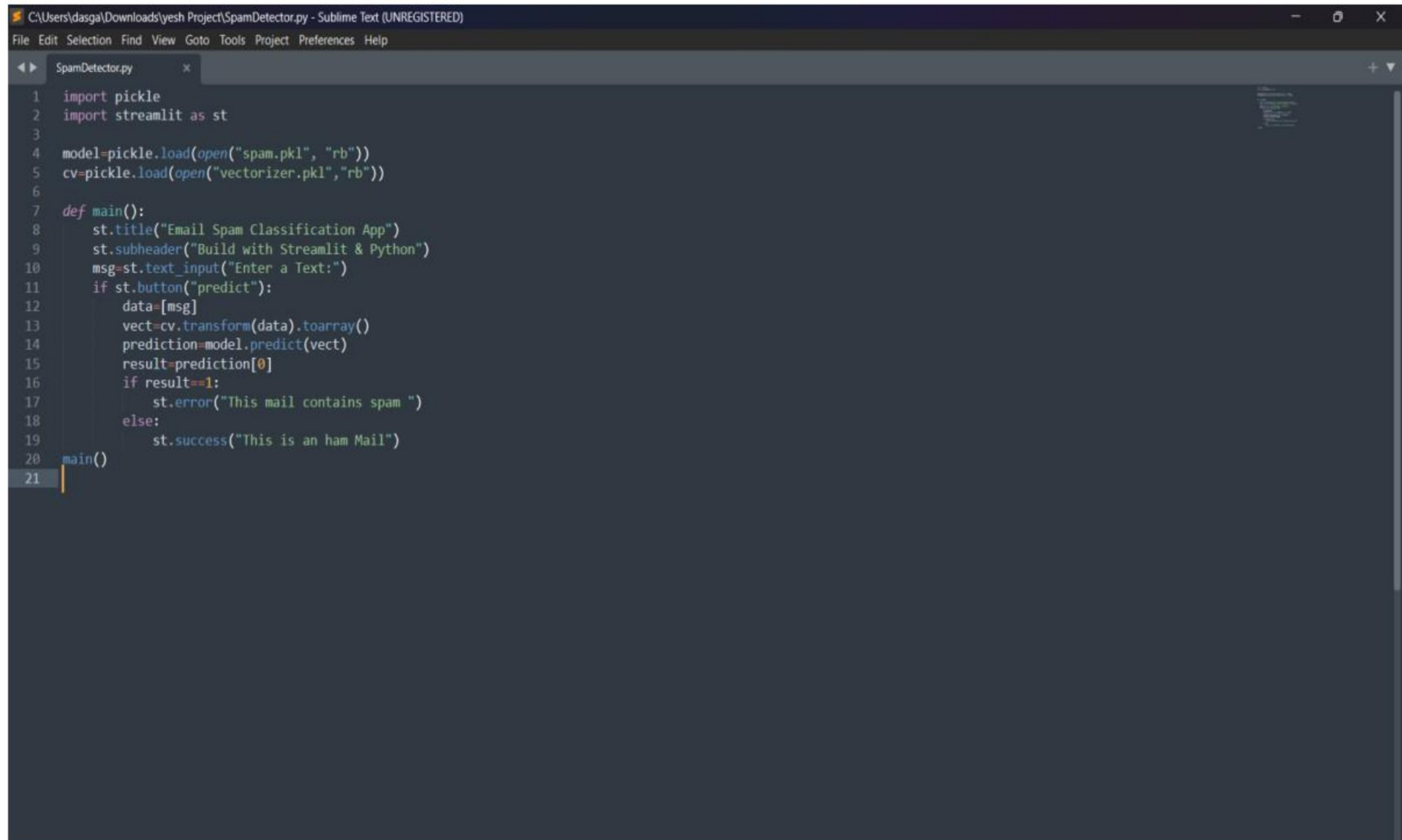
Fig4.1.4 Data class

```
In [8]: data.head()
```

	class	message
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

Fig4.1.5 New Data class

In the above table shows the class section contains which messages are ham as {0} and spam as {1} as we called the five section. So in our result we gave 2 input and we got 2 output as ham {0} and spam {0}.



```
C:\Users\dasga\Downloads\yesh Project\SpamDetector.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

SpamDetector.py
1 import pickle
2 import streamlit as st
3
4 model=pickle.load(open("spam.pkl", "rb"))
5 cv=pickle.load(open("vectorizer.pkl", "rb"))
6
7 def main():
8     st.title("Email Spam Classification App")
9     st.subheader("Build with Streamlit & Python")
10    msg=st.text_input("Enter a Text:")
11    if st.button("predict"):
12        data=[msg]
13        vect=cv.transform(data).toarray()
14        prediction=model.predict(vect)
15        result=prediction[0]
16        if result==1:
17            st.error("This mail contains spam ")
18        else:
19            st.success("This is an ham Mail")
20    main()
21
```

Fig4.1.6 Sublimetext

In front end I used sublime text.in there I imported my libraries and Implemented code.Sublime Text is one of the most popular text editors in the world. It's jam-packed with powerful features like multi-line editing, build systems for dozens of programming languages, regex find and replace, a Python API for developing plugins.

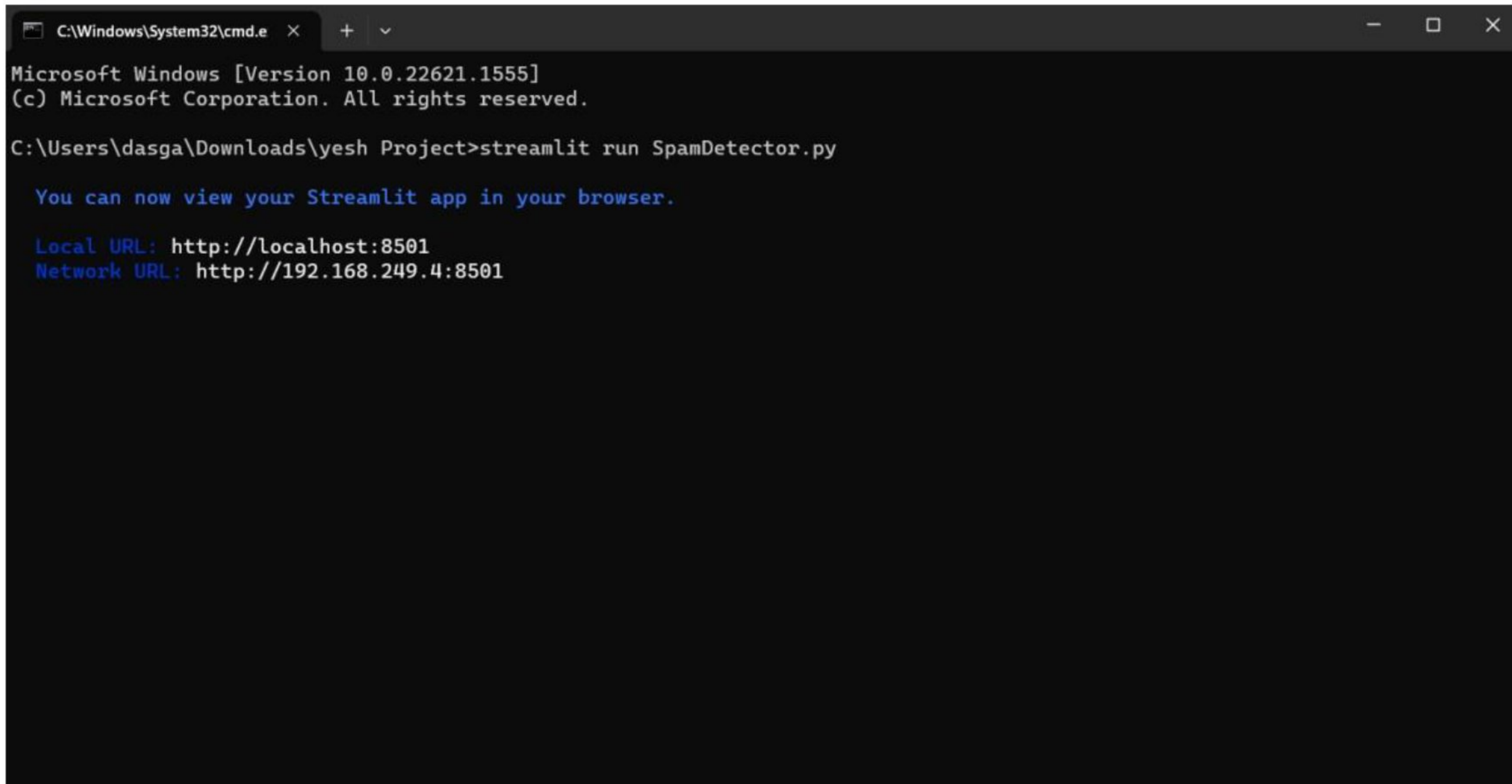
A screenshot of a Windows Command Prompt window. The title bar shows 'C:\Windows\System32\cmd.e' with standard window controls. The text inside the window reads: 'Microsoft Windows [Version 10.0.22621.1555] (c) Microsoft Corporation. All rights reserved. C:\Users\dasga\Downloads\yesh Project>streamlit run SpamDetector.py'. Below the command, it says 'You can now view your Streamlit app in your browser.' followed by 'Local URL: http://localhost:8501' and 'Network URL: http://192.168.249.4:8501'.

Fig4.1.7 opening cmd

After we completing the codes in sublime text we need to save the file and we need to go to command prompt and we need choose the drive and folder where we saved our works.then we need to type our created model in python format by typing >run streamlit SpamDetector.py .an interface will open in localhost by chrome .

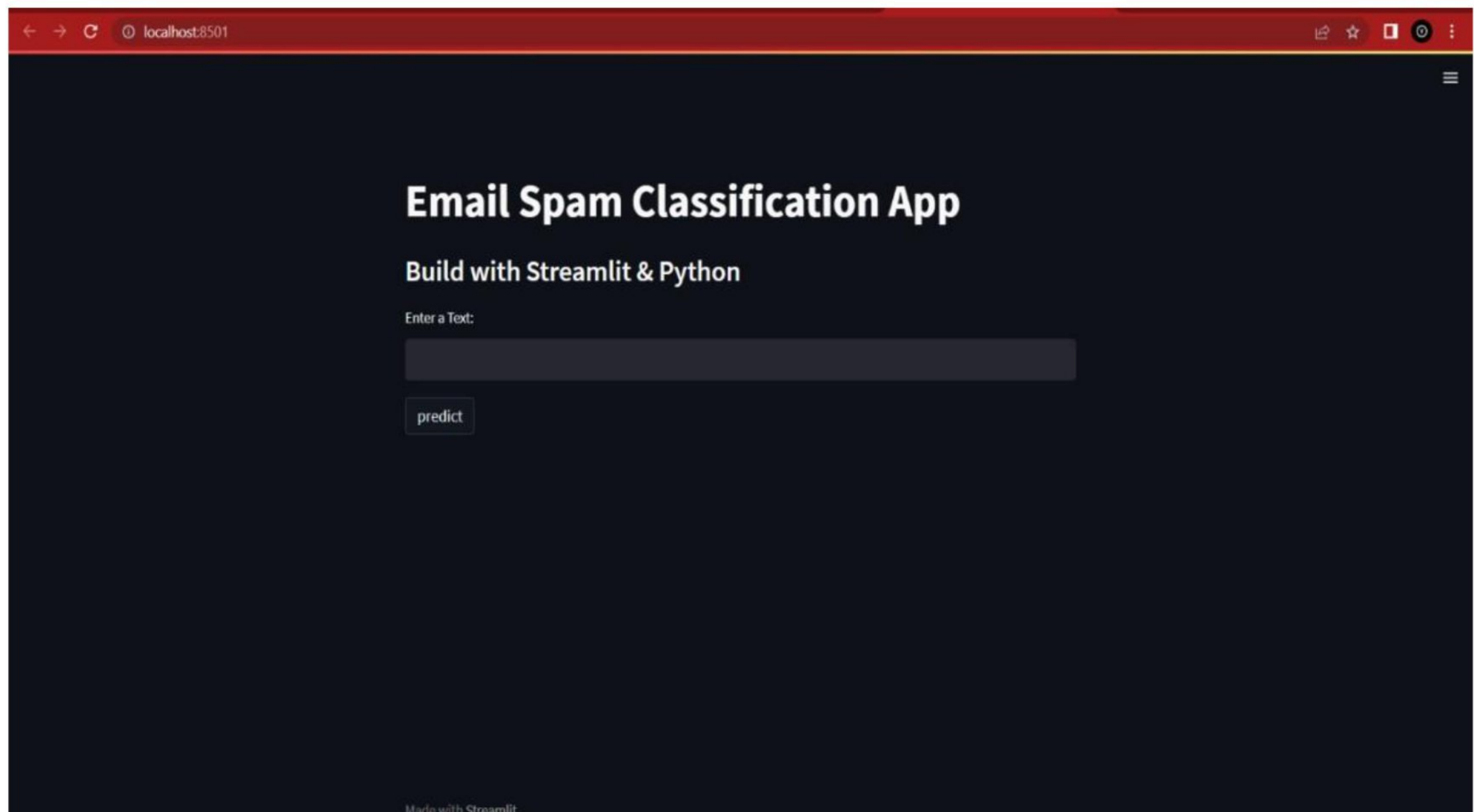


Fig4.1.8 final result

At the end we have our output using sublimetext. An interface where we can insert input and we can get the output by clicking predict. We build our sublimetext with python and our model using machine learning in Naïve bayes classification.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

CONCLUSION:

We proposed a novel algorithm for enhancing the accuracy of the Naive Bayes Spam Filter. The algorithm was implemented as an enhancement for Naive Bayes Classifier and is tested in Spam mail classification is a major area of concern these days as it helps in the detection of unwanted email send threats. spam filtering is an important issue in network security and machine learning techniques; the Naive Bayes classifier that is used has a very important role in this process of filtering e-mail spam. The quality of performance Naive Bayes classifier is also based on datasets that are used. As can see, a dataset that have fewer instances of e-mails and attributes can give good performance for Naive Bayes classifier. Naive Bayes classifier also can get highest precision that give highest percentage spam message manage to block if the dataset collect from single e-mail accounts. so we can see, why performance of Naïve Bayes classifier is good when used SPAMBASE dataset.

FUTURE SCOPE:

In the future we will try to optimize this project for much larger amount of dataset. Since our addition successfully enhances the Naive Bayes spam filter, we will try to implement the addition onto other machine learning spam filters such as Vector Space Models, clustering, and artificial neural networks. Combining these other methods will allow the improvement of spam detection across many different systems to ultimately create a well developed spam detector for text modifications.

APPENDIX

SOURCE CODE:

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

In [2]: data=pd.read_csv("spam.csv", encoding="latin-1")

In [3]: data.head(5)
```

	class	message	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

```
In [4]: data.columns

Index(['class', 'message', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], dtype='object')

In [5]: data.drop(['Unnamed: 2','Unnamed: 3','Unnamed: 4'],axis=1, inplace=True)
```

```
In [6]: data.head()
```

	class	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

```
In [7]: data['class']=data['class'].map({'ham':0,'spam':1})
```

```
In [8]: data.head()
```

	class	message
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

```
In [9]: from sklearn.feature_extraction.text import CountVectorizer
```

```
In [10]: cv=CountVectorizer()
```

```
In [11]: x=data['message']  
y=data['class']
```

```
In [12]: x.shape
```

```
(5572,)
```

```
In [13]: y.shape
```

```
(5572,)
```

```
In [14]: x=cv.fit_transform(x)
```

```
In [15]: x
```

```
<5572x8672 sparse matrix of type '<class 'numpy.int64''  
with 73916 stored elements in Compressed Sparse Row format>
```

```
In [16]: from sklearn.model_selection import train_test_split
```

```
In [17]: x_train, x_test, y_train, y_test=train_test_split(x, y, test_size=0.2)
```

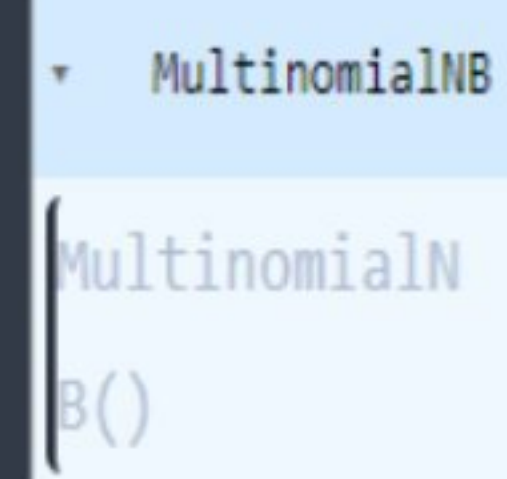
```
In [18]: x_train.shape
```

```
(4457, 8672)
```

```
In [19]: from sklearn.naive_bayes import MultinomialNB
```

```
In [20]: model=MultinomialNB()
```

```
In [21]: model.fit(x_train, y_train)
```



```
▼ MultinomialNB  
MultinomialNB()  
MultinomialNB()
```

```
In [22]: result=model.score(x_test, y_test)
```

```
In [23]: result=result*100
```

```
In [24]: result
```

```
98.20627802690582
```

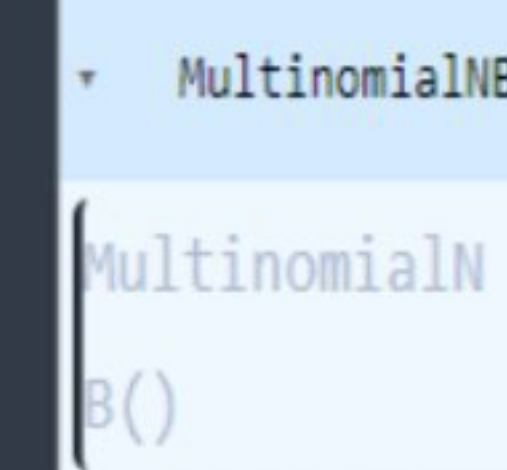
```
In [25]: import pickle
```

```
In [26]: pickle.dump(model, open("spam.pkl", "wb"))
```

```
In [27]: pickle.dump(cv, open("vectorizer.pkl", "wb"))
```

```
In [28]: clf=pickle.load(open("spam.pkl", "rb"))
```

```
In [29]: clf
```



```
▼ MultinomialNB  
MultinomialNB()  
MultinomialNB()
```

```
In [30]: msg="Hello There"

         data=[msg]

         vect=cv.transform(data).toarray()

         result=model.predict(vect)

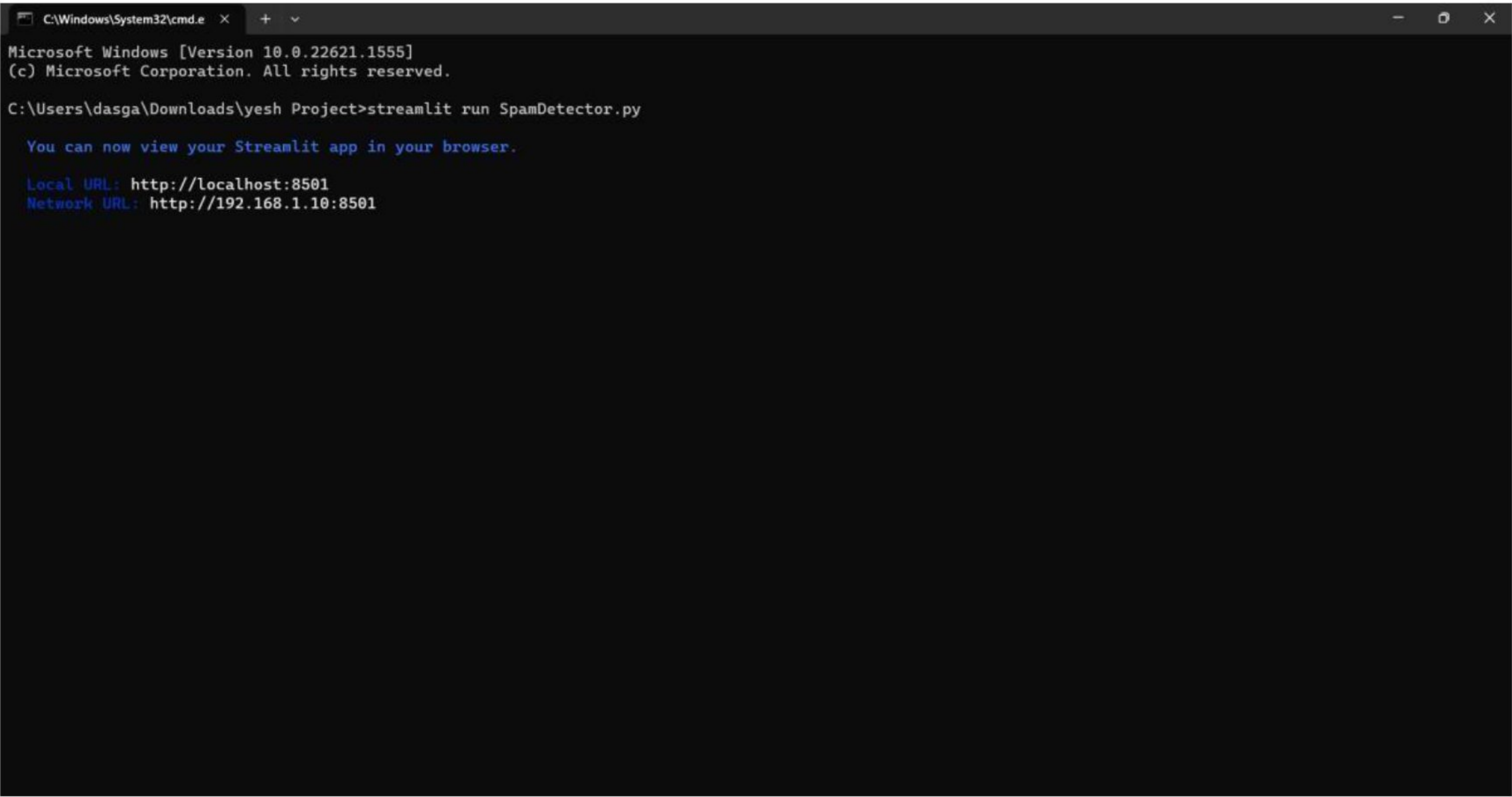
         print(result)

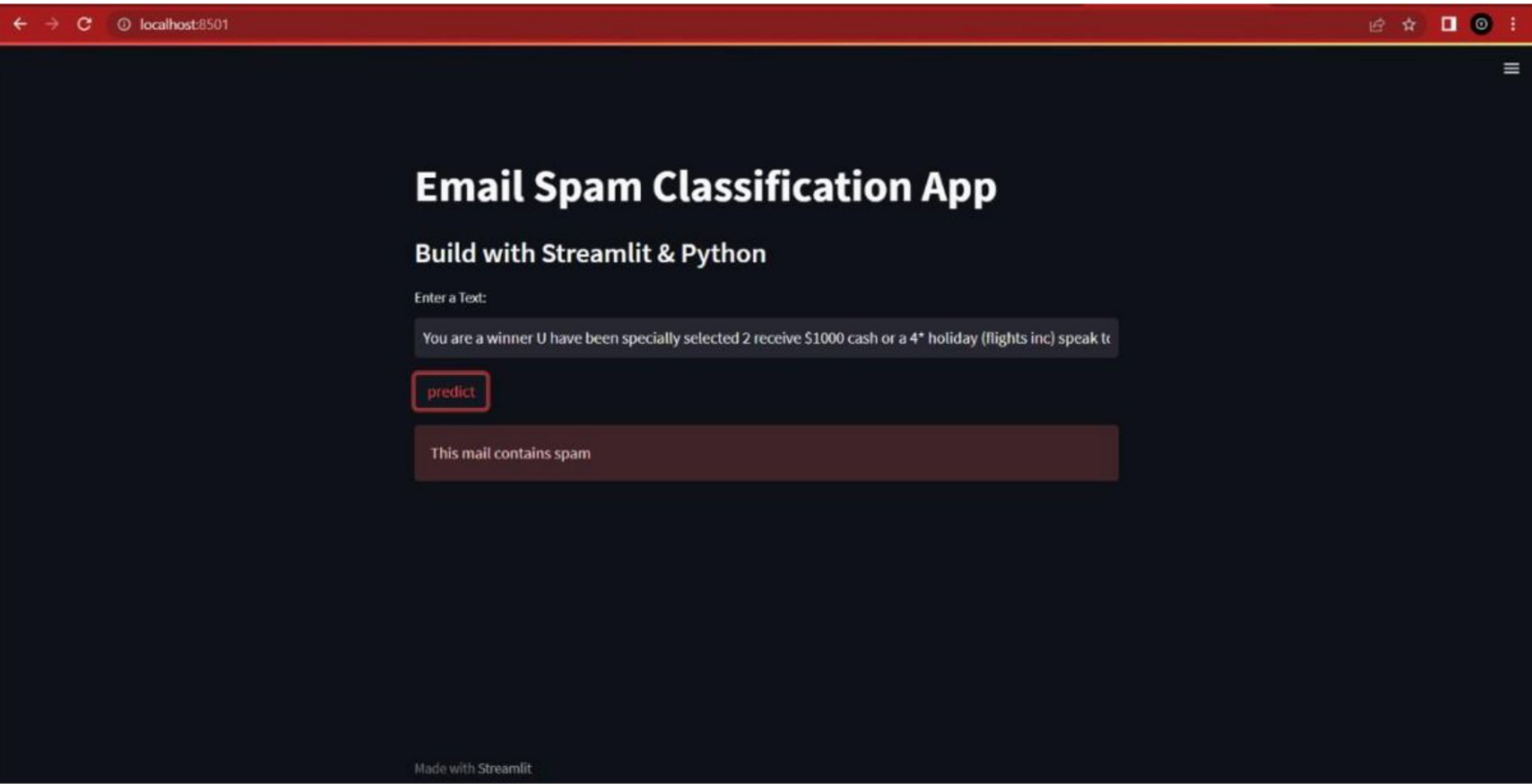
[0]
```

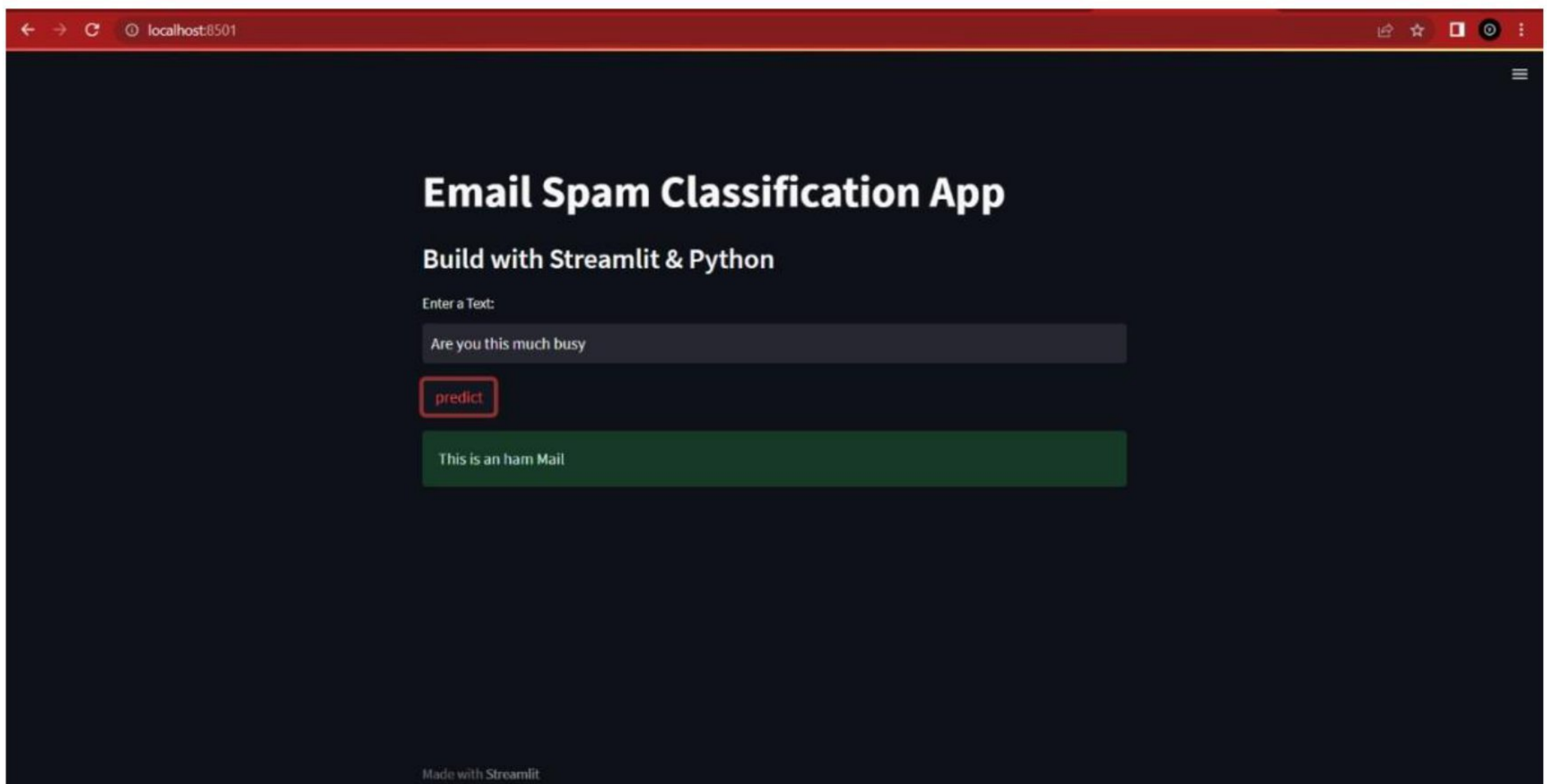
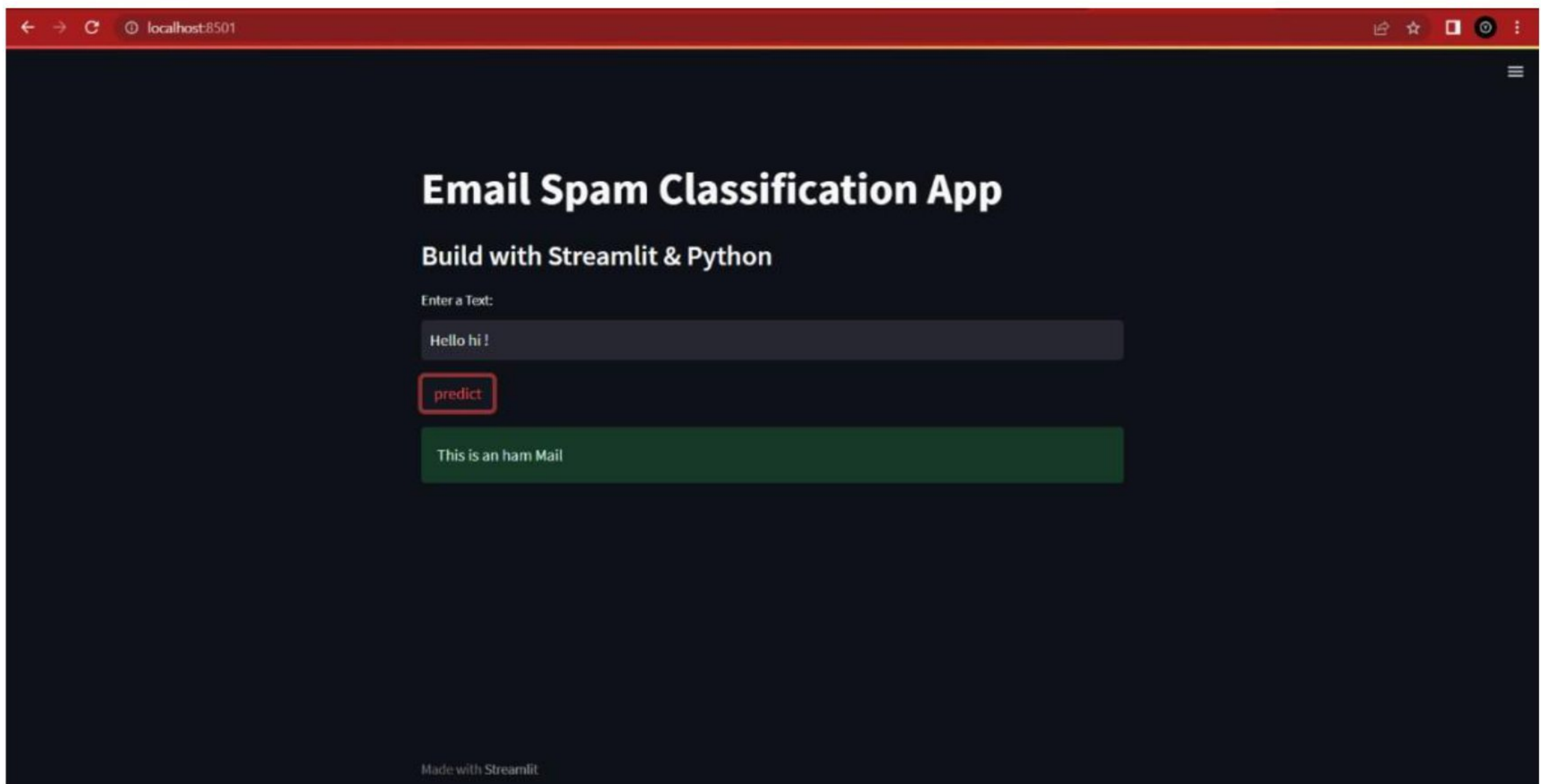
```
C:\Users\dasga\Downloads\yesh Project\SpamDetector.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

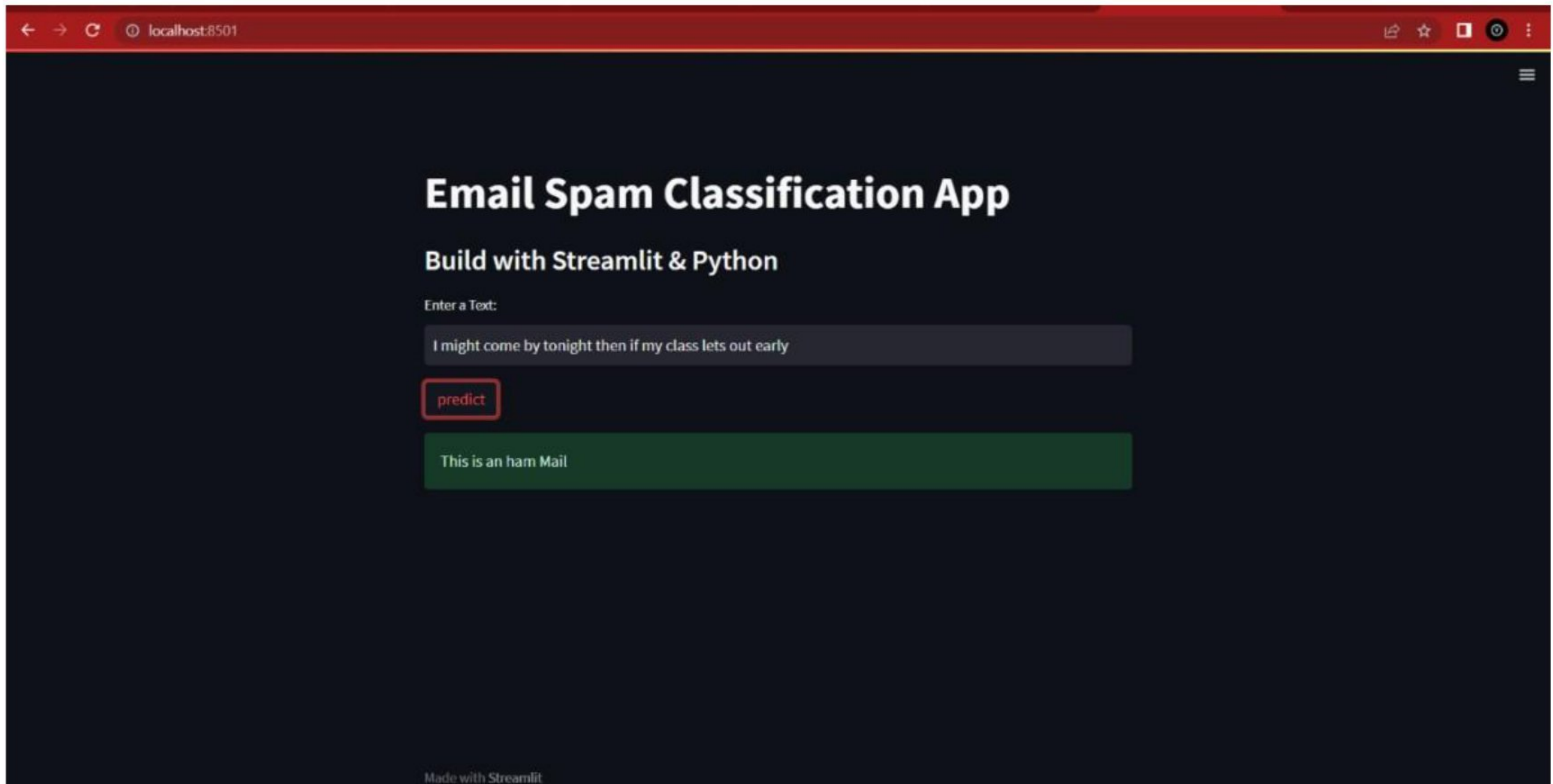
SpamDetector.py x
1 import pickle
2 import streamlit as st
3
4 model=pickle.load(open("spam.pkl", "rb"))
5 cv=pickle.load(open("vectorizer.pkl", "rb"))
6
7 def main():
8     st.title("Email Spam Classification App")
9     st.subheader("Build with Streamlit & Python")
10    msg=st.text_input("Enter a Text:")
11    if st.button("predict"):
12        data=[msg]
13        vect=cv.transform(data).toarray()
14        prediction=model.predict(vect)
15        result=prediction[0]
16        if result==1:
17            st.error("This mail contains spam ")
18        else:
19            st.success("This is an ham Mail")
20    main()
21
```

SCREEN SHOT :









REFERENCE:

1. <https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset>
2. Androutsopoulos I., J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos, "An evaluation of naive bayesian antispam filtering", In: 11th European Conference on Machine Learning, pp.9-17, Barcelona, Spain, 2000
3. S. K. Tuteja, "Classification Algorithms for Email Spam Filtering", 2016.
4. Rusland, N. F., Wahid, N., Kasim, S., & Hafit, H.. "Analysis of NaïveBayes Algorithmfor Email Spam Filtering across Multiple Datasets", 2017.