SPAM DETECTION USING MACHINE LEARNING

PROJECT REPORT

Submitted by

DEEPIKA A.	**(18BCS026)**
RAHUL R.	**(18BCS076)**
NARMADHA M.	**(18BCS014)**

in partial fulfillment for the award of the degree

of

Bachelor of Engineering

in

Computer Science and Engineering

Dr. Mahalingam College of Engineering and Technology

Pollachi - 642003

An Autonomous Institution

Affiliated to Anna University, Chennai - 600025

MAY 2022

Dr.MAHALINGAM COLLEGE OF ENGINEERING

AND TECHNOLOGY, POLLACHI -642003

(An Autonomous Institution Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report, "SPAM DETECTION USING MACHINE LEARNING" is the bonafide work of

DEEPIKA A.	(18BCS026)
RAHUL R.	(18BCS076)
NARMADHA M	(18BCS014)

who carried out the project work under my supervision.

Mrs.G.Gowri
SUPERVISOR
Assistant Professor
Dept. of Computer Science and Engineering
Dr. Mahalingam College of Engineering and Technology, Pollachi – 642003

Dr. G.Anupriya
HEAD OF THE DEPARTMENT
Dept. of Computer Science and Engineering
Dr. Mahalingam College of Engineering and Technology, Pollachi – 642003

Submitted for the Autonomous End Semester Examination Project Viva-voce held on ____________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

SPAM DETETION USING MACHINE LEARNING

ABSTRACT

Nowadays, a big part of people rely on available email or messages sent by the stranger. The possibility that anybody can leave an email or a message provides a golden opportunity for spammers to write spam message about our different interests.Spam fills inbox with number of ridiculous emails. Degrades our internet speed to a great extent .Steals useful information like our details on our contact list. Identifying these spammers and also the spam content can be a hot topic of research and laborious tasks. Email spam is an operation to send messages in bulk by mail .Since the expense of the spam is borne mostly by the recipient ,it is effectively postage due advertising. Spam email is a kind of commercial advertising which is economically viable because email could be a very cost effective medium for sender. With this proposed model the specified message can be stated as spam or not using Bayes' theorem and Naive Bayes' Classifier.

In Naive Bayes' Rule, we want to find the probability an email is spam, given it contains certain words. We do this by finding the probability that each word in the email is spam, and then multiply these probabilities together to get the overall email spam metric to be used in classification.

ACKNOWLEDGEMENT

First and foremost, we wish to express our deep unfathomable feeling, gratitude to our institution and our department for providing us a chance to fulfill our long cherished dreams of becoming Computer Science Engineers.

We express our sincere thanks to our honorable Secretary **Dr. C. Ramaswamy** for providing us with required amenities.

We wish to express our hearty thanks to **Dr. A. Rathinavelu,** Principal of our college, for his constant motivation and continual encouragement regarding our project work.

We are grateful to **Dr. G. Anupriya,** Head of the Department, Computer Science and Engineering, for her direction delivered at all times required. We also thank her for her tireless and meticulous efforts in bringing out this project to its logical conclusion.

Our hearty thanks to our guide **Mrs. G. Gowri,** Assistant Professor for his constant support and guidance offered to us during the course of our project by being one among us and all the noble hearts that gave us immense encouragement towards the completion of our project.

We also thank our review panel members **Dr. A. Noble Mary Juliet,** Associate Professor and **Mrs. N. Sumathi,** Assistant Professor for their continuous support and guidance.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

EDA	Exploratory Data Analyze
KNN	k- Nearest neighbor algorithm
ML	Machine Learning
NLTK	Natural Language Tool Kit
SVM	Support Vector Machine
UCI	Unified Configuration Interface

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

In recent years, internet has become an integral part of life. With increased use of internet, numbers of email users are increasing day by day. This increasing use of email has created problems caused by unsolicited bulk email messages commonly referred to as Spam. Email has now become one of the best ways for advertisements due to which spam emails are generated. Spam emails are the emails that the receiver does not wish to receive. a large number of identical messages are sent to several recipients of email. Spam usually arises as a result of giving out our email address 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 veryfrustrating if you receive large amounts of it .Identifying these spammers and the spam content isa 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. The spam filtering techniques areaccustomed protect our mailbox for spam mails.

1.1 Objective

- To Implement precision spam messages/e-mail filter using machine learning that uses Collected datasets from internet with help of datasets it filters messages/e-mail.
- To classify the received messages/e-mail using NLTK vectorizer.
- To show the filtered messages/e-mail spam or not in **stremlit** application.

1.2 Overview

The complete organized synopsis is as follows. Chapter 1 explains about the introduction and objectives of project Chapter 2 reflects about the existing literature on this domain and brief summary about the survey. Chapter 3 briefs about the Methodology. Chapter 4 explains results of data sets. Chapter 5 presents about the conclusion of this project.

CHAPTER 2

LITERATURE SURVEY

This chapter provides a brief insight on the related works of email spam detection and mail data optimization. Summary of various methodologies have also been discussed.

2.1 Opinion Rank:

From this survey , XiaofeiNiu et al. (2020) proposed the Opinion Rank algorithm for computing the trustworthiness of every available website and to identify the trustworthy ones with high trust values. This algorithm is based on a breadth-first search algorithm which starts from an existing set of trustworthy websites. Because, this websites play an important role in Opinion Rank. They also used other algorithms such as High PageRank and Inverse PageRank to rank thewebsites based on their trustworthiness. By using the public dataset, they have validated the Opinion Rank and HarMean PageRank which analyse the impact of website selection. The Opinion Rank algorithm computes the trustworthiness of all websites, trust propagation and trust combination operations which is defined in three valued subjective logic trust model. And the HarMean PageRank combines the results of PageRank and Inverse PageRank. The convergence and the performance of this algorithm is better than Trust Rank and Good Rank algorithms. This Opinion Rank algorithm consists of two components. Namely, the website selection and trust assessment. The website selection identifies the subset of trustworthy websites from a dataset. This algorithm will update the trust values of all websites from the chosen websites. Based on this trust values, websites are ranked by this Opinion Rank algorithm. Opinion Rank detects more trustworthy websites and lesser spam websites with a shorter time. One of its disadvantages is, it is very challenging to identify the subset of websites which are needed to update its trust value.

2.2 Spam Detection for Secure Mobile Communication:

In this paper, Luo GuangJun et al. (2020) proposed the applications of machine learning based-spam detection for accurate detection of spams. For classificationof spam and ham messages in mobile device communications they have used the Logistic Regression, K-nearest neighbor and

Detection Tree. The collection of SMS dataset is used for testing the methods. And the dataset is splitted into two sets as one is for testing and another one is for training. And 70 percent of data is used for training purposes and 30 percent is used for testing purposes. The Logistic Regression is a classifier which computes the predictive y in the problem of binary classification as 0 or 1 such that it belongs to class negative or class positive. It predicts values for the variable in multi classification. The Decision Tree is a supervised machine learning algorithm which is like the shape of a tree at which each node is a decision node or leaf node. In this tree the nodes are interlinked with each other. The K-nearest neighbour classification is also a supervised learning algorithm but this performance is not goodenough.

2.3 Machine Learning Based Spam Email Detection

In this Survey, Nandhini et al. (2018) proposed a machine learning model based on a hybrid bagging approach.by implementing with the help of two machine algorithms for detecting the spam emails. Namely, Naive Bayes algorithm and J48 (Decision tree) algorithm. In this process of detecting the spam mails, the dataset is divided into different sets and given as input to each of the algorithm.Totally, they performed three experiments in this paper. The first experiment is performed with the Naive Bayes algorithm. It is a classifier based on the probability and it computes the probabilities of the class of the given instances. And the second experiment is performed with the J48 Decision tree algorithm. It is based on the concept of entropy and it forms the decision trees of the training data. The third experiment is the proposed Spam Mail Detection (SMD) system by using the hybrid bagged approach which is the combination of J48 algorithm and Naive Bayes Multinomial classifier. It classifies the email intospam mails and ham mails. It consists of four modules which are preparation of email dataset, pre- processing of data, feature selection and hybrid bagged approach. Only the J48 algorithm gives the experimental results better. Other two experiment gives low performance. To enhance the system's performance by using the concept of boosting approach. It will replace the features of weak classifier learning features with a strong classifier's approach.

2.4 Email Spam Detection Using Integrated Naive Bayes approach

From this work, Kaur et al. (2018) have proposed a machine learning model by integrating the Naive Bayes algorithm and intelligence-based Particle Swarm Optimization which is used for detecting spam mail. The Naive Bayes algorithm is based on the Bayes theorem which has a strong probability distribution property. And the Particle Swarm Optimization is inspired from the behaviour of the fishes and the birds. The Naive Bayes algorithm determines the mail as spam class and non-spam class based on the keywords present on the email data. And the Particle Swarm Optimization method is further used to optimize the parameters of Naive Bayes algorithm to improve the accuracy and classification process. To perform the feature extraction, pre- processing is done for the email. The Pre-processing have some methods such as tokenization, stemming and stop word removal. After that we will apply the particle optimization method.Based on this feature of optimization method, the tokens of the mail is classified as spam or non- spam. They evaluated the performance of the system in terms of precision, recall and accuracy of the classification[6]. Their parameters are calculated with help of true positive, true negative, false positive and false negative. It has been found that the integrated approach of Naive Bayes and Particle Swarm Optimization overcomes the failure of the Naive Bayes approach. We can also use swarm optimization concepts like ant colony optimization, artificial bee colony optimization and firefly algorithm. Further, to improve the performance instead of Naive Bayes, we can use any other machine learning algorithm.

2.5 Machine Learning Methods For Spam EmailClassification

In this paper, Luo GuangJun et al. (2011) checked and reviewed the very popular machine learning methods for their capability of classifying the spam mails. Here the methods used are Bayesian classification, K-nearest neighbour classifier method, artificial neural network classifier method, Support vector machine classifier method, Artificial immune system classifier method, and rough sets classifier method. The Naive Bayes classifier method is based on the probability of an event occurring in the future which can be detected by the previous occurring of the same event.And based on that probability it will classify the mail as spam or ham mail. Here the probability of the word plays the major role in classification. The k-nearest neighbour classifier method is based on the example. It will check the previous documents for classification. And finding the nearest neighbour is done by using the traditional indexing methods. The Artificial Neural Network is also known as Neural Network. It is based on a biological neural network and consists of a collection of artificial neurons. At the time of the learning phase, it changes its structure based on the information that flows through the artificial network. It has the stages of training and filtering stage. The Support Vector Machine classifier method is based on the concept of decision planes which define the boundaries of the decision. This algorithm finds the optimal hyperplane with maximum margin for separating the two classes which is mainly required for solving the optimization problems. In the Artificial Immune System classifier method the overall response involves three evolutionary methods namely gene library, negative selection and global selection. This will organize the fittest antibodies by interacting with current antigens. The rough set classifier method has an ability to reduce the information systems. summarize these six methods, the Naive Bayes is the most accurate and also in terms of spam precision this method gave the highest precision among the six methods. The neural network has the simplest and fastest algorithms, while the rough set method is most complicated and it has to be hybrid with genetic algorithms to get the deserved results. The Artificial Immune System method gave a satisfying result which is to be expected for better performance but it gave the poor performance. It will provide the good performance when it is hybridized with rough set method.

2.6 Existing System

2.6.1 Standard Spam filtering:

Email Spam filtering process works through a set of protocols to determine either the message is spam or not. At present, a large number of spam filtering process have existed. Among them, Standard spam filtering process follows some rules and acts as a classifier with sets of protocols. Figure.1 shows that, a standard spam filtering process performed the analysis by following some steps.

First one is content filters which determine the spam message by applying several Machines learning techniques . Second, header filters act by extracting information from email header. Then, backlist filters determine the spam message and stop all emails which come from backlist file. Afterward, “Rules-based filters” recognize sender through subject line by using user defined criteria. Next, “Permission filters” send the message by getting recipients pre-approvement. Finally, “Challengeresponse filter” performed by applying an algorithm for getting the permission from the sender to send the mail.

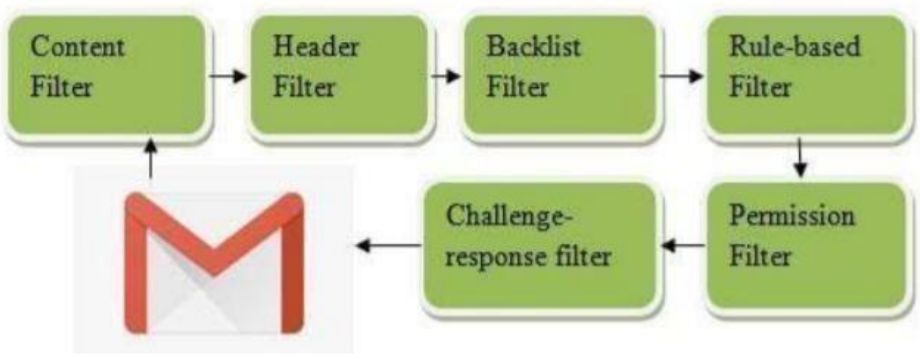


Figure 2.1:Standard spam filtering

2.6.2 Client Side and Enterprise Level Spam Filtering Methods

A client can send or receive an email by just one clicking through an ISP. Client level spam filtering provides some frameworks for the individual client to secure mail transmission. A client can easily filter spam through these several existing frameworks by installing on PC. This framework can interact with MUA (Mail user agent) and filtering the client inbox by composing, accepting and managing the messages . Enterprise level spam filtering is a process where provided frameworks are installing on mail server which interacts with the MTA for classifying the received messages or mail in order to categorize the spam message on the network. By this system, a user on that network can filter the spam by installing appropriate system more efficiently. By far most; current spam filtering frameworks use principle based scoring procedures. An arrangement of guidelines is connected to a message and calculate a score based principles thatare valid for the message. The message will consider as spam message when it exceeds the threshold value. As spammers are using various strategies, so all functions are redesigned routinely by applying a list-based technique to automatically block the message.

Figure 2.2 represents the method of client side and enterprise level spam filtering

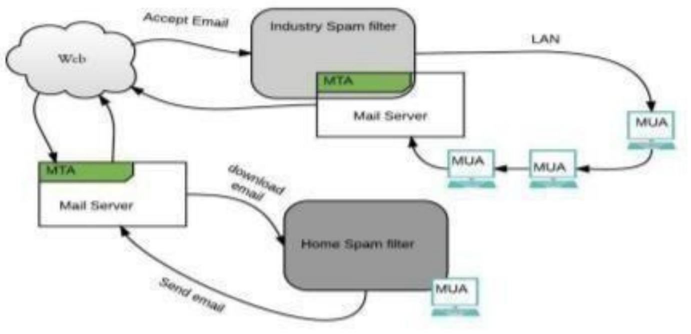


Figure 2.2: Client Side and Enterprise level Email spamFiltering System

2.7 Summary

Since last few decades, researchers are trying to make email as a secure medium. Spam filtering is one of the core features to secure email platform. Regarding this several types of research have been progressed reportedly but still there are some untapped potentials. Over time, still now e-mail spam classification is one of the major areas of research to bridge the gaps. Therefore, a large number of researches already have been performed on email spam classification using several techniques to make email more efficient to the users. That's why, this paper tried to arrange the summarized version of various existing Machine Learning approaches. In addition, in order to evaluates the most of the approaches like Random Forest, Naive Bayes , SVM , kNN , andRandom Forest used reliable and well known dataset for benchmarking performance such as SpamData , The Spam Assassin, The Spambase, Ecml-pkdd challenge dataset , corpora dataset , Enron dataset ,Trec dataset . Some of these dataset are in a prepared structure e.g. ECML and data accessible in Spambase UCI archive.

CHAPTER 3

METHODOLOGY

NLP is used to understand the structure and meaning of human language by analyzing different aspects like syntax, semantics, pragmatics, and morphology. Then, computer science transforms this linguistic knowledge into rule-based, machine learning algorithms that can solve specific problems and perform desired tasks.

Take Gmail, for example. Emails are automatically categorized as Promotions, Social, Primary, or Spam, thanks to an NLP task called keyword extraction. By "reading" words in subject lines and associating them with predetermined tags, machines automatically learn which category to assign emails.

3.1 Naive Bayes spam filtering

Naive Bayes classifiers are a popular statistical technique of e-mail filtering. They typically use bag-of-words features to identify spam e-mail, an approach commonly used in text classification.

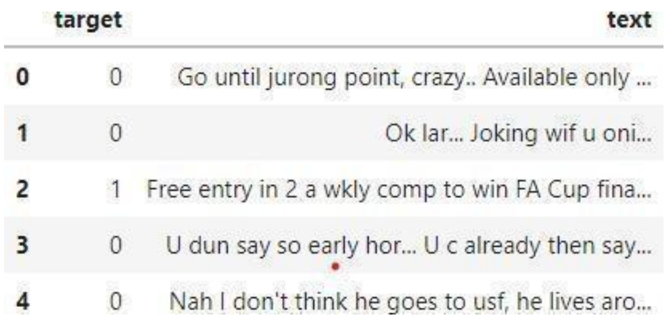
Naive Bayes classifiers work by correlating the use of tokens (typically words, or sometimes other things), with spam and non-spam e-mails and then using Bayes' theorem to calculate a probability that an email is or is not spam.

Naive Bayes spam filtering is a baseline technique for dealing with spam that can tailor itself to the email needs of individual users and give low false positive spam detection rates that are generally acceptable to users.

3.2 Email Dataset

Let's start with our spam detection data. We'll be using the open-source Spambase dataset from the UCI machine learning repository, a dataset that contains 5569 emails, of which 745 are spam. The target variable for this dataset is 'spam' in which a spam email is mapped to 1 and anything else is mapped to 0. The target variable can be thought of as what you are trying to predict.

In machine learning problems, the value of this variable will be modeled and predicted by other variables.



	target	text
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...

Figure 3.1: sample

Task: To classify an email into the spam or not spam.

To get to our solution we need to understand the four processing concepts below. Please note that the concepts discussed here can also be applied to other text classification problems.

3.3 Data Cleaning

Data cleaning or cleansing is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarsedata.

This phase involves the deletion of words or characters that do not add value tothe meaning of the text. Some of the standard cleaning steps are listed below:

- Lowering case
- Removal of special characters
- Removal of stopwords
- Removal of hyperlinks

3.3.1 Lowering Case:

Lowering the case of text is essential for the following reasons:

3.3.1.1 The words, ‘TEXT’, ‘Text’, ‘text’ all add the same value to a sentence.

3.3.1.2 Lowering the case of all the words is very helpful for reducing the dimensions by decreasing the size of the vocabulary.

3.3.2 Removal of special characters:

This is another text processing technique that will help to treat words like ‘hurray’ and ‘hurray!’ in the same way.

3.3.3 Removal of stop words:

Stopwords are commonly occurring words in a language like ‘the’, ‘a’, and so on. Most of the time they can be removed from the text because they don’t provide valuable information.

3.3.4 Removal of hyperlinks:

Next, we remove any URLs in the data. There is a good chance that email will have some URLs in it. We don’t need them for our further analysis as they do not add any value to the results.

3.4 Exploratory Data Analysis(EDA)

EDA is a phenomenon under data analysis used for gaining a better understanding of data aspects likevmain features of data,variables and relationships that hold between them identifying which variables are important for our problem. We shall look at various exploratory data analysis methods like Descriptive Statistics, which is a way of giving a brief overview of thedataset we are dealing with, including some measures and features of the sample grouping data [Basic grouping with group by] ANOVA, Analysis Of Variance, which is a computational method to divide variations in an observations set into different components,Correlation and correlation methods.

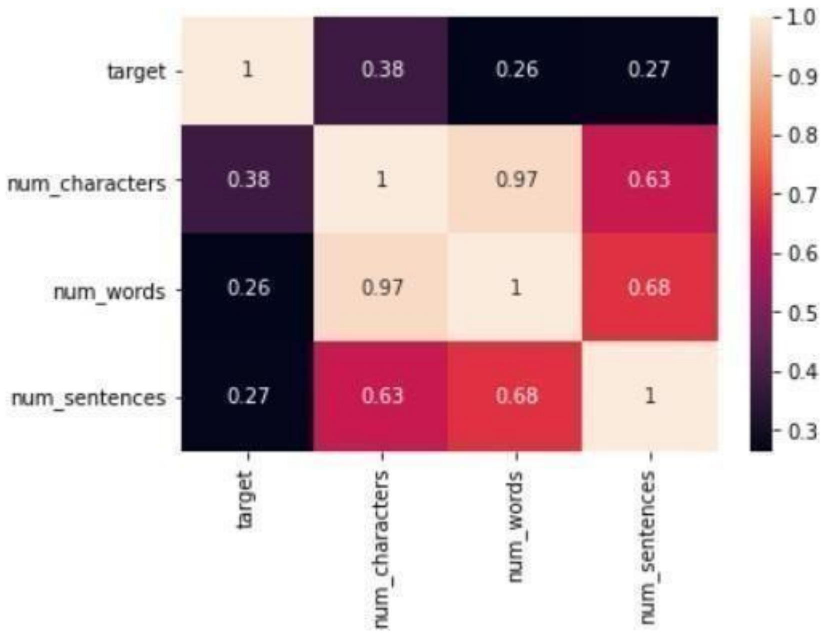


Figure 3.2: EDA

3.5 Data Preprocessing

Data preprocessing is an next step in building a Machine Learning model and depending on how well the data has been preprocessed; the results are seen. In NLP, text preprocessing is the first step in the process of building a model.

3.6 Model Building

The model building process involves setting up ways of collecting data, understanding and paying attention to what is important in the data to answer the questions you are asking, finding a statistical, mathematical or a simulation model to gain understanding and make predictions.

CHAPTER 4

RESULTS

The spam messages/e-mail power detection application was successfully deployed as Web Application. This Web Application uses trained machine learning model for predicton. Proposed Model(Naïve Bayes) produced good accuracy both on training and Testing.

Dataset

The given datasets produces about 99.5 percent accuracy in implementation

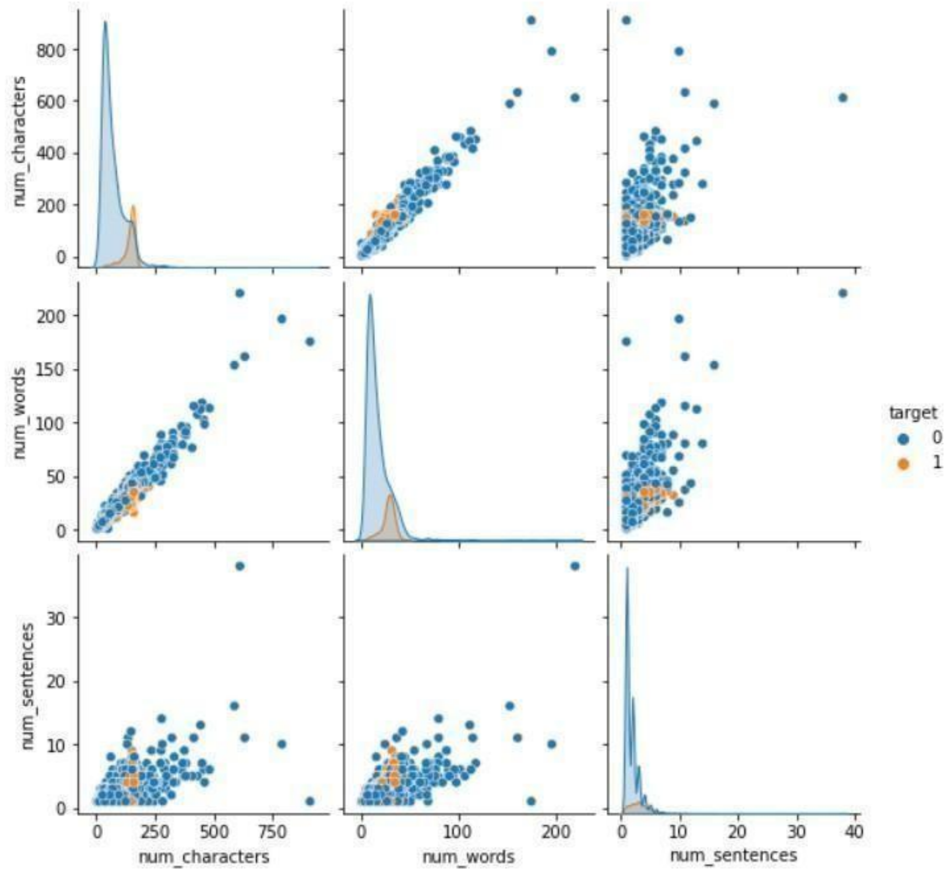


Figure 4.1: Model accuracy graph

CHAPTER 5

CONCLUSION

Spam email is one of the most demanding and troublesome internet issues in today’s world of communication and technology. Spammers by generating spam mails are misusing this communication facility and thus affecting organization’s and many email users. In this paper, a Spam Mail Detection system is introduced which makes use of a NLP approach for its implementation. The classification algorithms used in this approach are Naïve Bayes. The accuracy achieved by Naïve Bayes algorithm is 90% respectively. system shows that the experimental results are better when performed on only naive bayes algorithm. In order to enhance the system’s performance and results, the concept of boosting approach could be considered for future work. The boosting technique will replace the weak classifier’s learning features with the strong classifier’s features and thus enhancing the overall system’s performance.

REFERENCES

[1] A. J. Saleh, A. Karim, B. Shanmugam et al., “An intelligent spam detection model based on artificial immune system,” *Information*, vol. 10, no. 6, p. 209, 2019.View at: Publisher Site | Google Scholar

[2] A. Sharaff, “Comparative study of classification algorithms for spam email detection,” in Emerging Research in Computing, Information, Communicationand Applications, pp. 237–244, Springer, Berlin, Germany, 2016.View at: Publisher Site

[3] B. Yu and Z.-B. Xu, “A comparative study for content-based dynamic spam classification using four machine learning algorithms,” Knowledge-Based Systems, vol. 21, no. 4, pp. 355–362, 2008.View at: Publisher Site | Google Scholar

[4] D. Ruano-Ordás, F. Fdez-Riverola, and J. R. Méndez, “Using evolutionary computation for discovering spam patterns from e-mail samples,” Information Processing & Management, vol. 54, no. 2, pp. 303–317, 2018.View at: Publisher Site | Google Scholar

[5] D. Ruano-Ordás, F. Fdez-Riverola, and J. R. Méndez, “Using evolutionary computation for discovering spam patterns from e-mail samples,” *Information Processing & Management*, vol. 54, no. 2, pp. 303–317, 2018.View at: Publisher Site | Google Scholar

[6] I. Alsmadi and I. Alhami, “Clustering and classification of email contents,” Journal of King Saud University—Computer and Information Sciences, vol. 27, no. 1, pp. 46–57, 2015.View at: Publisher Site |Google Scholar

[7] M. Bassiouni, M. Ali, and E. A. El-Dahshan, “Ham and spam E-mails classification using machine learning techniques,” Journal of Applied Security Research, vol. 13, no. 3, pp. 315–331, 2018.View at: Publisher Site | Google Scholar

[8] R. K. Kumar, “Comparative study on email spam classifier using data mining techniques,” in Proceedings of the International MultiConference of Engineers and Computer Scientists, pp. 14–16, Hong Kong, March 2012.View at: Google Scholar

[9] S. K. Trivedi and S. Dey, “Interplay between probabilistic classifiers and boosting algorithms for detecting complex unsolicited emails,” *Journal of Advances in Computer Networks*, vol. 1, pp. 132–136, 2013.View at: Publisher Site | Google Scholar

[10] S. Smadi, N. Aslam, and L. Zhang, “Detection of online phishing email using dynamic evolving neural network based on reinforcement learning,” *Decision Support Systems*, vol. 107, pp. 88–102, 2018.View at: Publisher Site | Google Scholar

[11] S. Y. Bhat, “Spammer classification using ensemble methods over structural social network features,” in Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), pp. 454–458, Warsaw, Poland, August 2014.View at: Publisher Site | Google Scholar

[12] SMS, “Spam collection dataset,” 2019, https://www.kaggle.com/datasets.View at: Google Scholar

Website reference:

1. Streamlit package https://streamlit.io/
2. Natural Language Tool Kit https://www.nltk.org/
3. spam datasethttps://www.kaggle.com/karthickveerakumar/spam-filter/
4. simplespam filter : https://www.kaggle.com/mohitr/simple-spam-filter
5. https://in.springboard.com/blog/email-spam-filtering-using-naive-bayes- classifier/

APPENDIX A : SAMPLE CODE

```
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()

    text = nltk.word_tokenize(text)

    y = []

    for i in text:

        if i.isalnum():

            y.append(i)

    text = y[:]

    y.clear()

    for i in text:

        if i not in stopwords.words('english') and i not in string.punctuation:

            y.append(i)

    text = y[:]

    y.clear()

    for i in text:

        y.append(ps.stem(i))

    return " ".join(y)
```

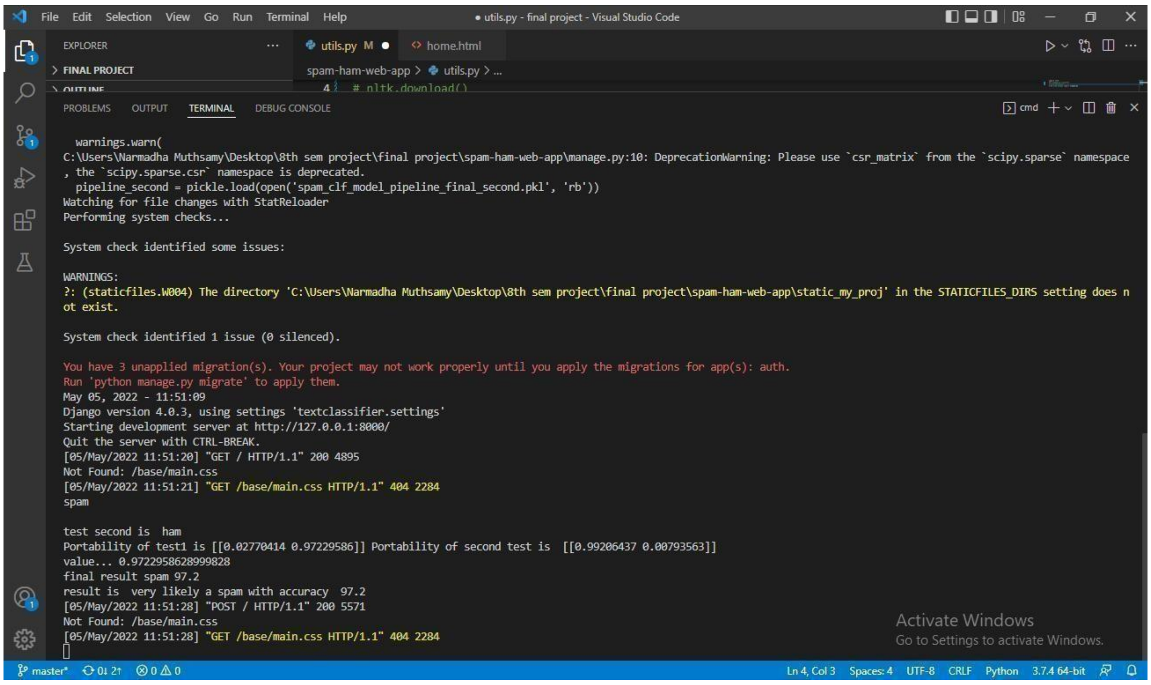
```
tfidf = pickle.load(open('vectorizer.pkl','rb'))

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

st.title("Email/SMS Spam Classifier")

input_sms = st.text_area("Enter the message")

if st.button('Predict'):

    # 1. preprocess

    transformed_sms = transform_text(input_sms)

    # 2. vectorize

    vector_input = tfidf.transform([transformed_sms])

    # 3. predict

    result = model.predict(vector_input)[0]

    # 4. Display

    if result == 1:

        st.header("Spam")

    else:

        st.header("Not Spam")

import numpy as np

import pandas aspd

df.info()

df.sample(5)

# renaming the cols

df.rename(columns={'v1':'target','v2':'text'},inplace=True) df.sample(5)

from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()

df['target'] = encoder.fit_transform(df['target'])
```

```
df.head()

# missing values

df.isnull().sum()

# check for duplicate values

df.duplicated().sum()

# remove duplicates

df = df.drop_duplicates(keep='first')

df.duplicated().sum()

df.shape

df.head()

df['target'].value_counts()

import matplotlib.pyplot as plt

plt.pie(df['target'].value_counts(), labels=['ham','spam'],autopct="%0.2f")

plt.show()

import nltk

nltk.download('punkt')

df['num_characters'] = df['text'].apply(len)

df.head()

# num of words

df['num_words'] = df['text'].apply(lambda x:len(nltk.word_tokenize(x)))

df['num_sentences']= df['text'].apply(lambdax:len(nltk.sent_tokenize(x)))

df[['num_characters','num_words','num_sentences']].describe()

df[df['target']==0][['num_characters','num_words','num_sentences']].describe()

df[df['target']==1][['num_characters','num_words','num_sentences']].describe()

plt.figure(figsize=(12,6))

sns.histplot(df[df['target'] == 0]['num_characters'])
sns.histplot(df[df['target'] == 1]['num_characters'],color='red')

plt.figure(figsize=(12,6))

sns.histplot(df[df['target'] == 0]['num_words'])

sns.histplot(df[df['target'] == 1]['num_words'],color='red')

sns.pairplot(df,hue='target')

sns.heatmap(df.corr(),annot=True)

def transform_text(text):

    text = text.lower()

    text = nltk.word_tokenize(text)

    y = []

    for i in text:

        if i.isalnum():

            y.append(i)

    text = y[:]

    y.clear()

    for i in text:

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

    text = y[:]

    y.clear()

    for i in text:

        y.append(ps.stem(i))

    return " ".join(y)

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.")

from nltk.stem.porter import PorterStemmer
```

```
ps = PorterStemmer()

ps.stem('loving')

df['transformed_text'] = df['text'].apply(transform_text)

from collections import Counter

sns.barplot(pd.DataFrame(Counter(spam_corpus).most_common(30))[0],pd.DataF
rame(Counter(spam_corpus).most_common(30))[1])

plt.xticks(rotation='vertical')

plt.show()

ham_corpus = []

for msg in df[df['target'] == 0]['transformed_text'].tolist():

    for word in msg.split():

        ham_corpus.append(word)

from collections import Counter

sns.barplot(pd.DataFrame(Counter(ham_corpus).most_common(30))[0],pd.DataFr
ame(Counter(ham_corpus).most_common(30))[1])

plt.xticks(rotation='vertical')

plt.show()

from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer

cv = CountVectorizer()

tfidf = TfidfVectorizer(max_features=3000)

from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB

from sklearn.metrics import accuracy_score,confusion_matrix,precision_score

gnb = GaussianNB()

mnb = MultinomialNB()

bnb = BernoulliNB()

gnb.fit(X_train,y_train)

y_pred1 = gnb.predict(X_test)
```

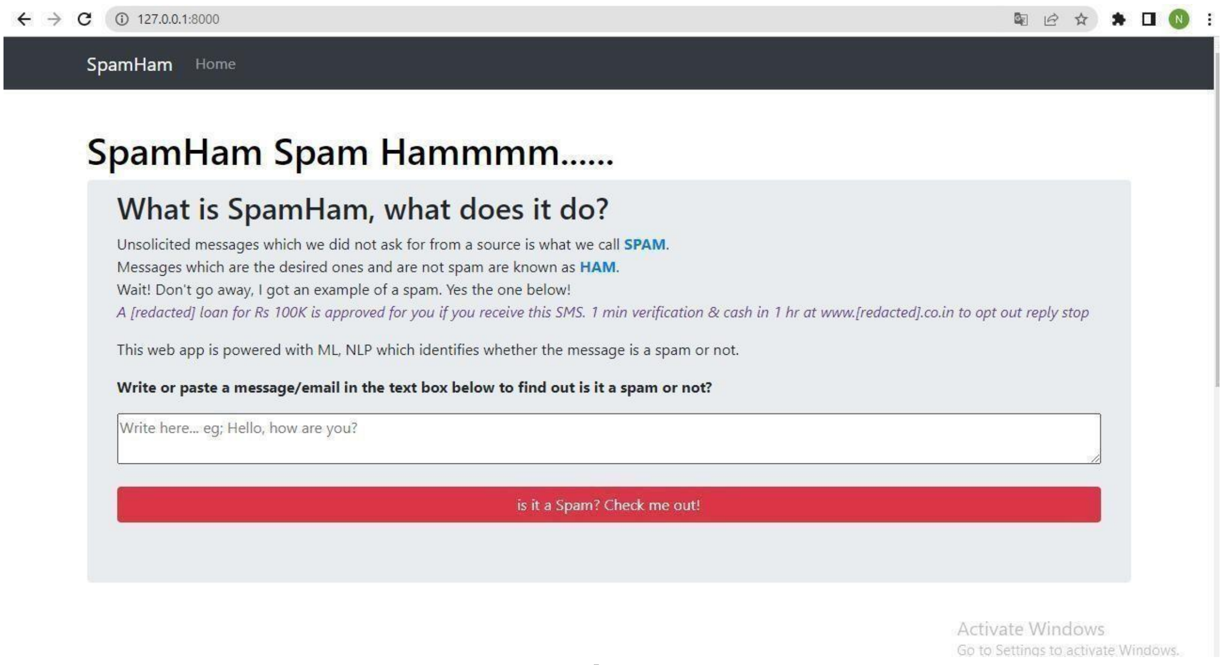
```
print(accuracy_score(y_test,y_pred1))
print(confusion_matrix(y_test,y_pred1))
print(precision_score(y_test,y_pred1))
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
fromsklearn.neighbors 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
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)
clfs = {
    'SVC' : svc,
    'KN' : knc,
    'NB': mnb,
    'DT': dtc,
    'LR': lrc,
    'RF': rfc,
    'AdaBoost': abc,
    'BgC': bc,
    'ETC': etc,
    'GBDT':gbdt,
    'xgb':xgb
}
def train_classifier(clf,X_train,y_train,X_test,y_test):
    clf.fit(X_train,y_train)
    y_pred = clf.predict(X_test)
    accuracy = accuracy_score(y_test,y_pred)
  precision = precision_score(y_test,y_pred)
  return accuracy,precision
  train_classifier(svc,X_train,y_train,X_test,y_test)
  for name,clf in clfs.items():
  current_accuracy,current_precision=train_classifier(clf,X_train,y_train,X_test,y_t
  est)
    print("For ",name)
    print("Accuracy - ",current_accuracy)
    print("Precision - ",current_precision)
    accuracy_scores.append(current_accuracy)
    precision_scores.append(current_precision)

temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_max_ft_3000':accuracy_scores,'Precision_max_ft_3000':precision_scores}).sort_values('Precision_max_ft_3000',ascending=False)

temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_scaling':accuracy_scores,'Precision_scaling':precision_scores}).sort_values('Precision_scaling',ascending=False)

new_df = performance_df.merge(temp_df,on='Algorithm')

new_df_scaled = new_df.merge(temp_df,on='Algorithm')

temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_num_chars':accuracy_scores,'Precision_num_chars':precision_scores}).sort_values('Precision_num_chars',ascending=False)

new_df_scaled.merge(temp_df,on='Algorithm')

# Voting Classifier

svc = SVC(kernel='sigmoid', gamma=1.0,probability=True)

mnb = MultinomialNB()

etc = ExtraTreesClassifier(n_estimators=50, random_state=2)

from sklearn.ensemble import VotingClassifier

voting = VotingClassifier(estimators=[('svm', svc), ('nb', mnb), ('et', etc)],voting='soft')

voting.fit(X_train,y_train)

VotingClassifier(estimators=[('svm', SVC(gamma=1.0, kernel='sigmoid',

probability=True)), ('nb', MultinomialNB()),('et',

ExtraTreesClassifier(n_estimators=50, random_state=2))],voting='soft')

y_pred = voting.predict(X_test)

print("Accuracy",accuracy_score(y_test,y_pred))

print("Precision",precision_score(y_test,y_pred))
```

APPENDIX B : SCREENSHOTS


File Edit Selection View Go Run Terminal Help
● utils.py - final project - Visual Studio Code
EXPLORER
utils.py M
home.html
FINAL PROJECT
spam-ham-web-app > utils.py > ...
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
cmd
master* ⟳ 0↓ 2↑ ⊗ 0 ⚠ 0
Ln 4, Col 3 Spaces: 4 UTF-8 CRLF Python 3.7.4 64-bit
Activate Windows
Go to Settings to activate Windows.

```
  warnings.warn(
C:\Users\Narmadha Muthsamy\Desktop\8th sem project\final project\spam-ham-web-app\manage.py:10: DeprecationWarning: Please use `csr_matrix` from the `scipy.sparse` namespace
, the `scipy.sparse.csr` namespace is deprecated.
  pipeline_second = pickle.load(open('spam_clf_model_pipeline_final_second.pkl', 'rb'))
Watching for file changes with StatReloader
Performing system checks...

System check identified some issues:

WARNINGS:
?: (staticfiles.W004) The directory 'C:\Users\Narmadha Muthsamy\Desktop\8th sem project\final project\spam-ham-web-app\static_my_proj' in the STATICFILES_DIRS setting does n
ot exist.

System check identified 1 issue (0 silenced).

You have 3 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): auth.
Run 'python manage.py migrate' to apply them.
May 05, 2022 - 11:51:09
Django version 4.0.3, using settings 'textclassifier.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[05/May/2022 11:51:20] "GET / HTTP/1.1" 200 4895
Not Found: /base/main.css
[05/May/2022 11:51:21] "GET /base/main.css HTTP/1.1" 404 2284
spam

test second is  ham
Portability of test1 is [[0.02770414 0.97229586]] Portability of second test is  [[0.99206437 0.00793563]]
value... 0.9722958628999828
final result spam 97.2
result is  very likely a spam with accuracy  97.2
[05/May/2022 11:51:28] "POST / HTTP/1.1" 200 5571
Not Found: /base/main.css
[05/May/2022 11:51:28] "GET /base/main.css HTTP/1.1" 404 2284
```

Figure B.1: Localhost deployment


127.0.0.1:8000
SpamHam
Home
is it a Spam? Check me out!
Activate Windows
Go to Settings to activate Windows.

SpamHam Spam Hammmm......

What is SpamHam, what does it do?

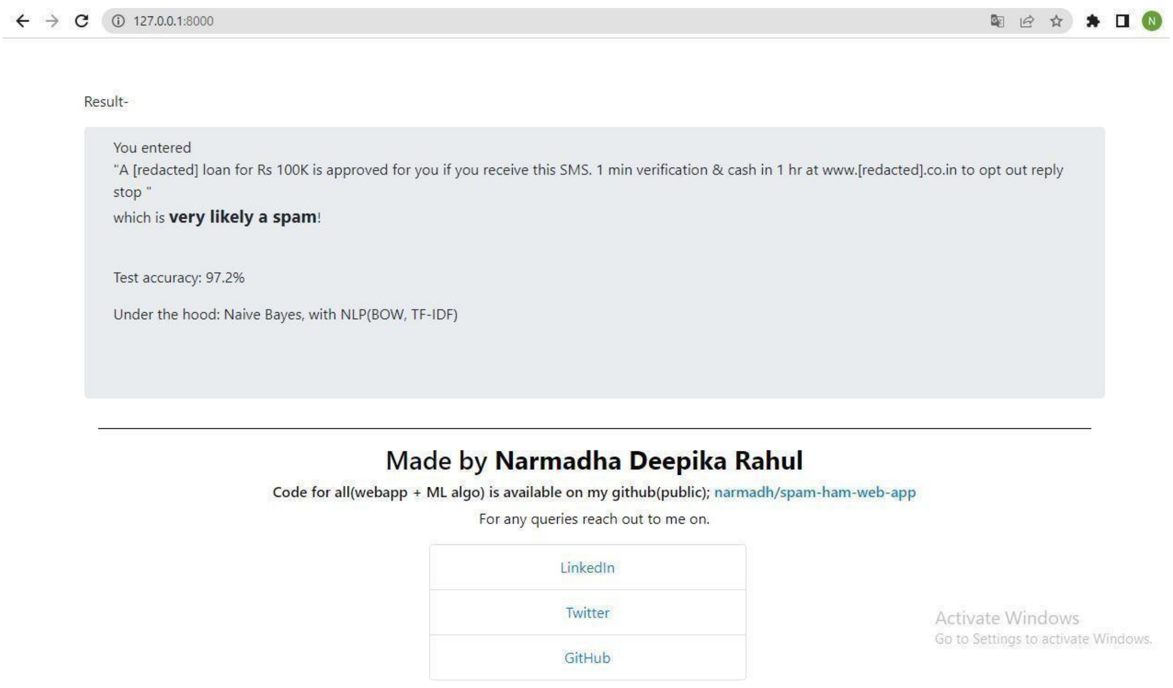
Unsolicited messages which we did not ask for from a source is what we call **SPAM**.
Messages which are the desired ones and are not spam are known as **HAM**.
Wait! Don't go away, I got an example of a spam. Yes the one below!
A [redacted] loan for Rs 100K is approved for you if you receive this SMS. 1 min verification & cash in 1 hr at www.[redacted].co.in to opt out reply stop

This web app is powered with ML, NLP which identifies whether the message is a spam or not.

Write or paste a message/email in the text box below to find out is it a spam or not?

Write here... eg; Hello, how are you?

Figure B.2:Sample output spam



Result-

You entered
"A [redacted] loan for Rs 100K is approved for you if you receive this SMS. 1 min verification & cash in 1 hr at www.[redacted].co.in to opt out reply stop "
which is **very likely a spam**!

Test accuracy: 97.2%

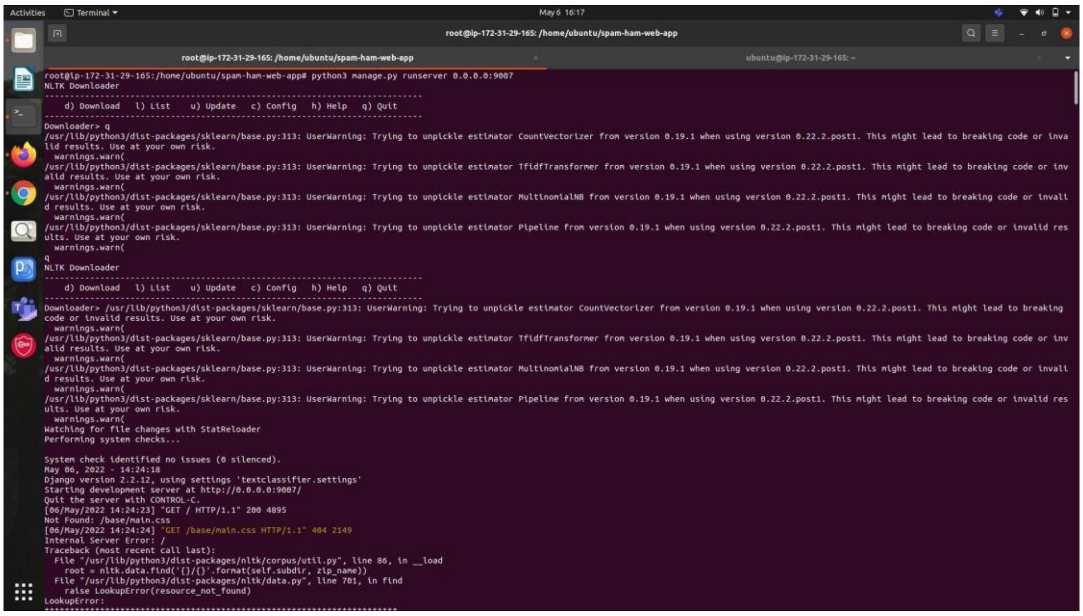
Under the hood: Naive Bayes, with NLP(BOW, TF-IDF)

Made by **Narmadha Deepika Rahul**

Code for all(webapp + ML algo) is available on my github(public); narmadh/spam-ham-web-app

For any queries reach out to me on.

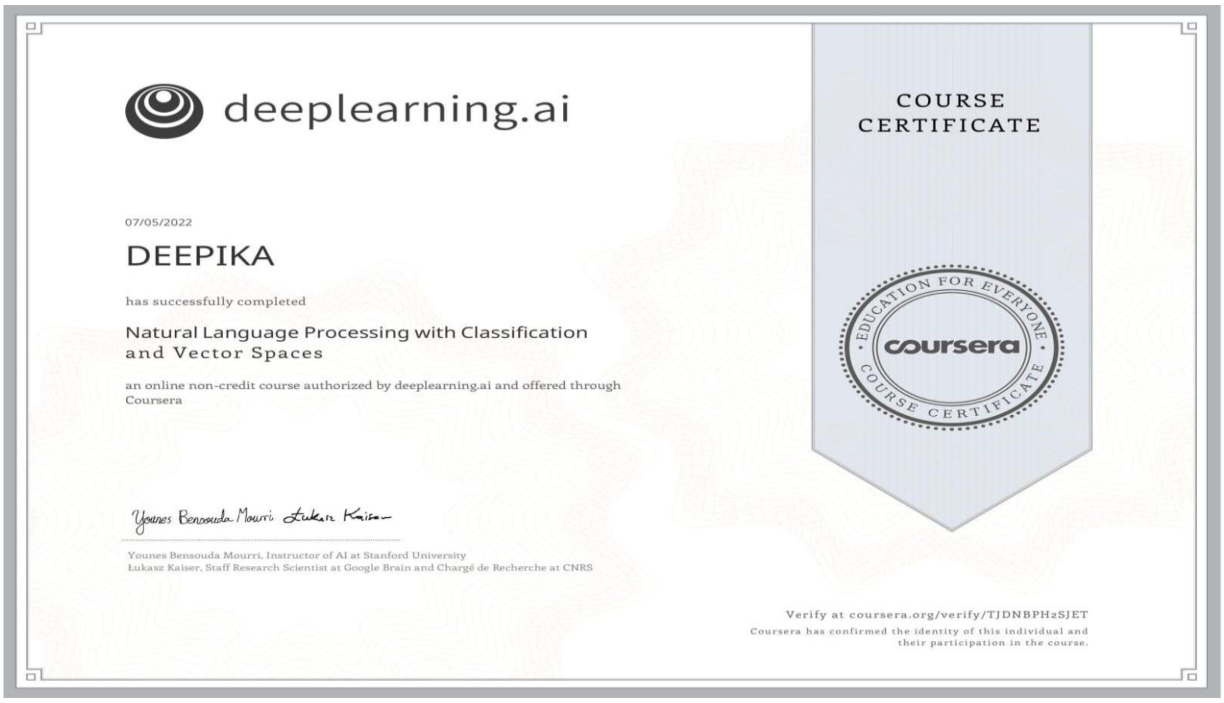
Figure B.3:Sample Output Not spam


Activities
Terminal
May 6 16:17
root@ip-172-31-29-165: /home/ubuntu/spam-ham-web-app
ubuntu@ip-172-31-29-165: ~

```
root@ip-172-31-29-165:/home/ubuntu/spam-ham-web-app# python3 manage.py runserver 0.0.0.0:9007
NLTK Downloader
---------------------------------------------------------------------------
    d) Download   l) List    u) Update   c) Config   h) Help   q) Quit
---------------------------------------------------------------------------
Downloader> q
/usr/lib/python3/dist-packages/sklearn/base.py:313: UserWarning: Trying to unpickle estimator CountVectorizer from version 0.19.1 when using version 0.22.2.post1. This might lead to breaking code or inva
lid results. Use at your own risk.
  warnings.warn(
/usr/lib/python3/dist-packages/sklearn/base.py:313: UserWarning: Trying to unpickle estimator TfidfTransformer from version 0.19.1 when using version 0.22.2.post1. This might lead to breaking code or inv
alid results. Use at your own risk.
  warnings.warn(
/usr/lib/python3/dist-packages/sklearn/base.py:313: UserWarning: Trying to unpickle estimator MultinomialNB from version 0.19.1 when using version 0.22.2.post1. This might lead to breaking code or invali
d results. Use at your own risk.
  warnings.warn(
/usr/lib/python3/dist-packages/sklearn/base.py:313: UserWarning: Trying to unpickle estimator Pipeline from version 0.19.1 when using version 0.22.2.post1. This might lead to breaking code or invalid res
ults. Use at your own risk.
  warnings.warn(
q
NLTK Downloader
---------------------------------------------------------------------------
    d) Download   l) List    u) Update   c) Config   h) Help   q) Quit
---------------------------------------------------------------------------
Downloader> /usr/lib/python3/dist-packages/sklearn/base.py:313: UserWarning: Trying to unpickle estimator CountVectorizer from version 0.19.1 when using version 0.22.2.post1. This might lead to breaking
code or invalid results. Use at your own risk.
  warnings.warn(
/usr/lib/python3/dist-packages/sklearn/base.py:313: UserWarning: Trying to unpickle estimator TfidfTransformer from version 0.19.1 when using version 0.22.2.post1. This might lead to breaking code or inv
alid results. Use at your own risk.
  warnings.warn(
/usr/lib/python3/dist-packages/sklearn/base.py:313: UserWarning: Trying to unpickle estimator MultinomialNB from version 0.19.1 when using version 0.22.2.post1. This might lead to breaking code or invali
d results. Use at your own risk.
  warnings.warn(
/usr/lib/python3/dist-packages/sklearn/base.py:313: UserWarning: Trying to unpickle estimator Pipeline from version 0.19.1 when using version 0.22.2.post1. This might lead to breaking code or invalid res
ults. Use at your own risk.
  warnings.warn(
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
May 06, 2022 - 14:24:18
Django version 2.2.12, using settings 'textclassifier.settings'
Starting development server at http://0.0.0.0:9007/
Quit the server with CONTROL-C.
[06/May/2022 14:24:23] "GET / HTTP/1.1" 200 4895
Not Found: /base/main.css
[06/May/2022 14:24:24] "GET /base/main.css HTTP/1.1" 404 2149
Internal Server Error: /
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/nltk/corpus/util.py", line 86, in __load
    root = nltk.data.find('{}/{}'.format(self.subdir, zip_name))
  File "/usr/lib/python3/dist-packages/nltk/data.py", line 701, in find
    raise LookupError(resource_not_found)
LookupError:
```

Figure B.4 : Cloud deployment

ONLINE CERTIFICATION



deeplearning.ai

COURSE CERTIFICATE

07/05/2022

DEEPIKA

has successfully completed

Natural Language Processing with Classification and Vector Spaces

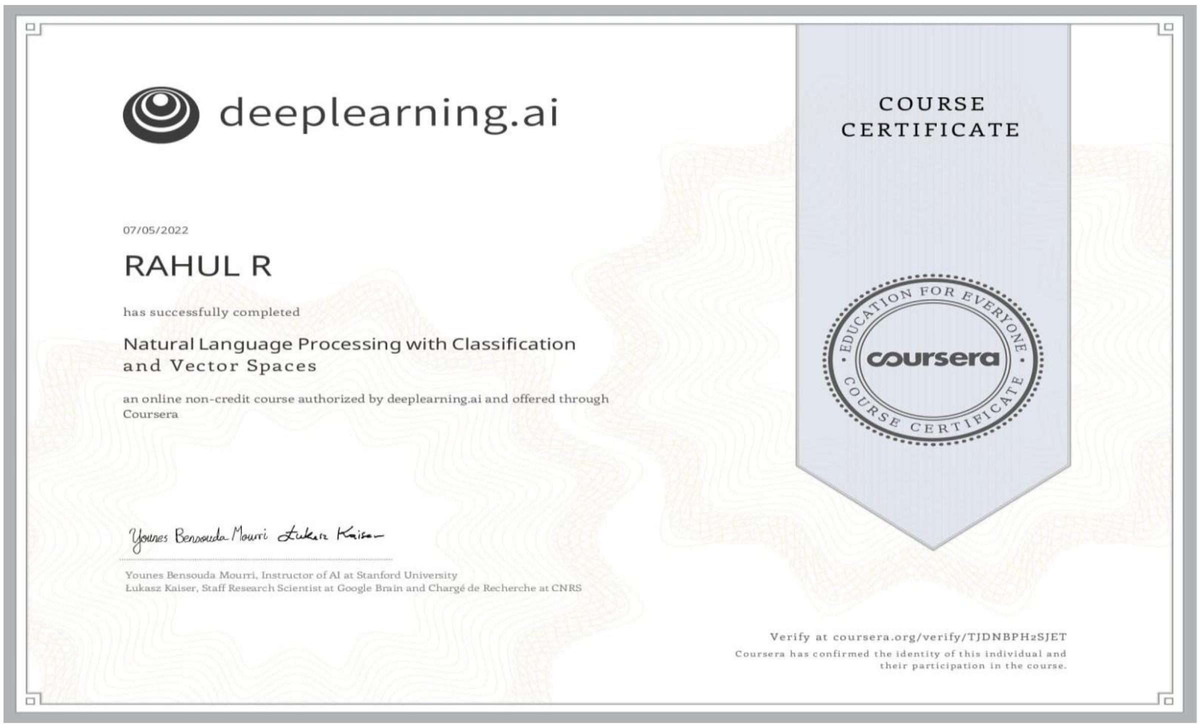
an online non-credit course authorized by deeplearning.ai and offered through Coursera

EDUCATION FOR EVERYONE coursera COURSE CERTIFICATE

Younes Bensouda Mourri Łukasz Kaiser

Younes Bensouda Mourri, Instructor of AI at Stanford University
Łukasz Kaiser, Staff Research Scientist at Google Brain and Chargé de Recherche at CNRS

Verify at coursera.org/verify/TJDNBPH2SJET
Coursera has confirmed the identity of this individual and their participation in the course.



deeplearning.ai

COURSE CERTIFICATE

07/05/2022

RAHUL R

has successfully completed

Natural Language Processing with Classification and Vector Spaces

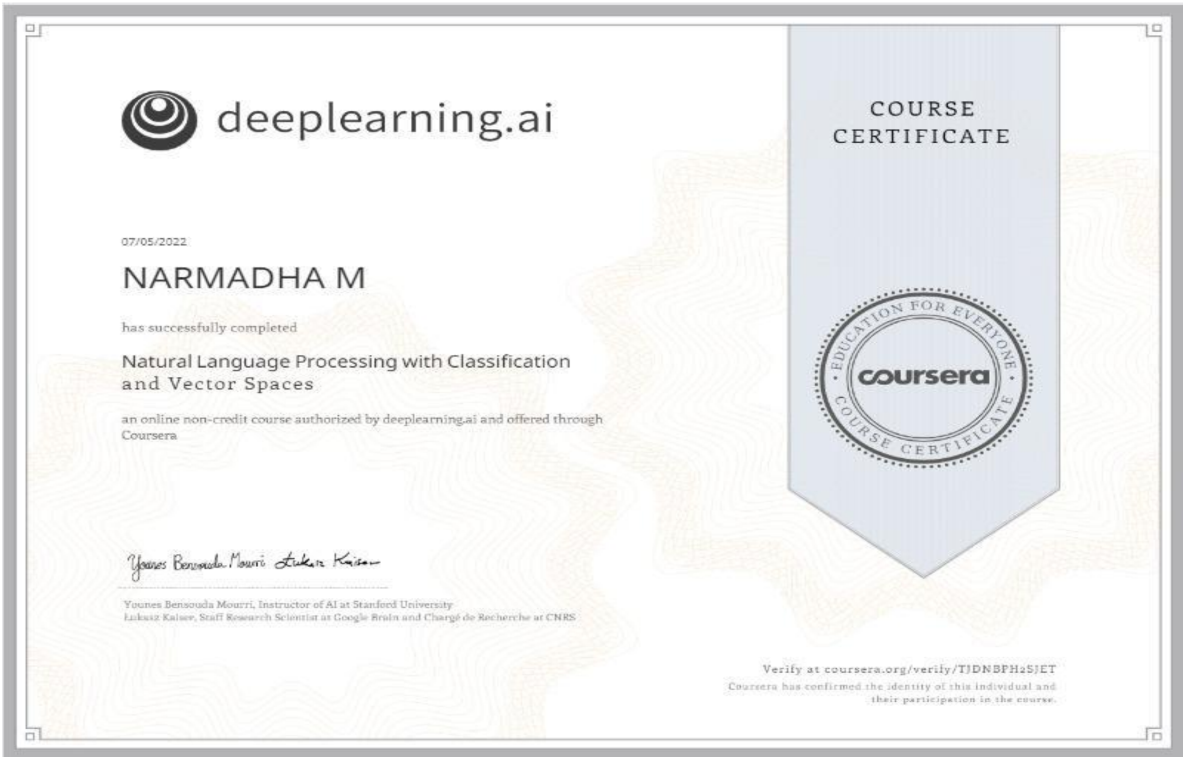
an online non-credit course authorized by deeplearning.ai and offered through Coursera

EDUCATION FOR EVERYONE coursera COURSE CERTIFICATE

Younes Bensouda Mourri Łukasz Kaiser

Younes Bensouda Mourri, Instructor of AI at Stanford University
Łukasz Kaiser, Staff Research Scientist at Google Brain and Chargé de Recherche at CNRS

Verify at coursera.org/verify/TJDNBPH2SJET
Coursera has confirmed the identity of this individual and their participation in the course.



deeplearning.ai

COURSE CERTIFICATE

07/05/2022

NARMADHA M

has successfully completed

Natural Language Processing with Classification and Vector Spaces

an online non-credit course authorized by deeplearning.ai and offered through Coursera

EDUCATION FOR EVERYONE · coursera · COURSE CERTIFICATE

[illegible]

Younes Bensouda Mourri, Instructor of AI at Stanford University
Łukasz Kaiser, Staff Research Scientist at Google Brain and Chargé de Recherche at CNRS

Verify at coursera.org/verify/TJDNBPH25JET
Coursera has confirmed the identity of this individual and their participation in the course.